MBA Thesis

Improving Cryptocurrency Price Direction Forecasting: Increasing Accuracy by Applying Confidence Score Thresholds to Random Forest Ensemble Predictions

Daniel Csanadi & Martin Lennartsson

Supervisor
Jonas Månsson

Karlskrona, Sweden

June 2024

This thesis is submitted to the Department of Industrial Economics at Blekinge Institute of Technology in partial fulfilment of the requirements for the Degree of Master of Science in Industrial Economics and Management. The thesis is awarded 15 ECTS credits.

The author(s) declare(s) that they have completed the thesis work independently. All external sources are cited and listed under the References section. The thesis work has not been submitted in the same or similar form to any other institution(s) as part of another examination or degree.

Author information:

Daniel Csanadi
dacs23@student.bth.se

Martin Lennartsson
martin.lennartsson@gmail.com

Department of Industrial Economics
Blekinge Institute of Technology
SE-371 79 Karlskrona, Sweden

Website: www.bth.se
Telephone: +46 455 38 50 00
Fax: +46 455 38 50 57

Abstract

The intersection of econometrics, statistics, and corporate finance has led to significant advancements in algorithmic trading in the cryptocurrency markets. While machine learning has been widely applied to predict price movements, most research focus solely on directional classification without considering the possibility to use the prediction confidence score of the models as a mean to increase the accuracy of the predictions further. This thesis investigates whether incorporating the confidence scores from machine learning classifiers can improve the prospect of positive trading outcomes in algorithmic cryptocurrency trading. As a result, it contributes to both academic understanding of market efficiency and serve as a basis for future algorithmic trading strategies. We employ an ensemble of two Random Forest models to predict the directional movement of cryptocurrency prices, increasing the accuracy by filtering out low-confidence predictions. The approach is benchmarked against Chen et al. (2020) using the same historical Bitcoin market dataset and further validated on ten cryptocurrencies from January 2022 to December 2023 with 5-minute trading data samples from the ByBit exchange. The results demonstrate that strategies incorporating confidence thresholds yield higher prediction accuracy than those that do not, though they produce fewer trade signals. Our method outperforms the top results achieved in the benchmark study with an accuracy of 72%, while maintaining 44% of the output samples, and with further increasing levels of accuracy for higher confidence score thresholds. Thus, indicating that confidence-based filtering can enhance both prediction accuracy and in extension also trading performance. In conclusion, combining two Random Forest models in an ensemble and incorporating confidence score thresholds significantly improve the accuracy of cryptocurrency price movement predictions compared to individual models. The method outperforms previous research for higher confidence thresholds and provides algorithmic traders with a tool to further increase prediction accuracy, however at the cost of decreasing number of trading opportunities for increasing accuracy. It thus demonstrates a shortcoming in the Efficient Market Hypothesis (Fama, 1970) and provides support for a more evolutionary approach to understanding markets – especially those that are less mature – such as the Adaptive Market Hypothesis (Lo, 2004). Future research could extend this work by investigating other prediction windows to further increase accuracy and output size detention, and by extending the method to also predict the level of magnitude of price changes, enabling more advanced trading strategies and enhanced risk management.

Keywords:

cryptocurrency, machine learning, random forest, confidence score, efficient market hypothesis, adaptive market hypothesis

Table of contents

1. Introduction 2

2. Literature Review 4

2.1. Benchmark Paper 7

2.2. Crypto Outlook 8

3. Theoretical Framework and Empirical Strategy 9

3.1. Efficient Market Hypothesis 9

3.2. Empirical Strategy for Technical Analysis 10

3.3. Technical Indicators Used in the Technical Analysis 10

3.4. Using Machine Learning for Technical Analysis 14

3.5. Random Forest 15

4. Data 17

5. Method 19

5.1. Procedure 19

5.2. Evaluation metrics 20

6. Results 22

6.1. Benchmark Period Results (July 2017 – January 2018) 22

6.2. Multi-currency Results Over Two Years (January 2022 – December 2023) 26

7. Discussion 28

7.1. Limitations 28

7.2. Recommendations for Future Research 29

7.3. Practical Applications 29

8. References 30

Appendix A: Detailed Accuracy Data for Each of the Ten Cryptocurrency Pairs in the January 2022 – December 2023 Dataset 33

A.1 Bitcoin (BTC) 33
A.2 Solana (SOL) 34
A.3 Ripple (XRP) 35
A.4 DogeCoin (DOGE) 36
A.5 Cardano (ADA) 37
A.6 Curve DAO Token (CRV) 38
A.7 Litecoin (LTC) 39
A.8 Ethereum (ETH) 40
A.9 Avalanche (AVAX) 41
A.10 Chainlink (LINK) 42

List of Tables

Table 1: The 5-minute price data of Bitcoin 18
Table 2: Average accuracy, precision and recall for the two models RF_1 and RF_2, which together constitute the ensemble model detailed in the method chapter 22
Table 3: Ratios of correct, incorrect, and inconclusive observations for different threshold values, and the corresponding accuracy on the remaining data 25
Table 4: Accuracy % for the ten crypto currencies in the 2022-2023 sample for each threshold 1.0-2.0, including overall average accuracy % and average remaining data % 27

List of Figures

Figure 1: Schematic representation of a RF decision tree 15
Figure 2: Bearish (red) & bullish (green) candlestick representation 18
Figure 3: Illustrative description of how the sequential training and evaluation of the models has been carried out 19
Figure 4: Distribution of correct, incorrect and inconclusive predictions for the benchmark period samples 23
Figure 5: Boxplot of correct, incorrect and inconclusive predictions for the benchmark period samples and their distribution across the total confidence score of models RF_1 + RF_2 23
Figure 6: Number of correct versus incorrect predictions over increasing bins of confidence score for the RF_1 model 24
Figure 7: Number of correct versus incorrect predictions over increasing bins of confidence score for the RF_2 model 24
Figure 8: Number of correct versus incorrect predictions over increasing bins of confidence score for the combined predictions of RF_1 and RF_2, where both models predict the same price direction 25
Figure 9: Changes in accuracy for increasing values of the threshold T, with the maximum accuracy of 67% achieved by Chen et al. included as a benchmark line for comparison 26
Figure 10: Indexed price development (start=100) for the sampled ten currencies, including an equal-weighted market index for comparison of overall market development 2022-2023 27

List of abbreviations

ADA	Cardano [cryptocurrency]
ADX	Average Directional Index
AMH	Adaptive Market Hypothesis
API	Application Programming Interface
ARIMA	AutoRegressive Integrated Moving Average
ATR	Average True Range
AVAX	Avalanche [cryptocurrency]
BNN	Bayesian Neural Network
BTC	Bitcoin [cryptocurrency]
CRV	Curve DAO Token [cryptocurrency]
DOGE	Dogecoin [cryptocurrency]
EMA	Exponential Moving Average
EMH	Efficient Market Hypothesis
ETH	Ethereum [cryptocurrency]
GBC	Gradient Boosting Classifier
GRNN	Generalized Regression Neural Network
GRU	Gated Recurrent Unit
LINK	Chainlink [cryptocurrency]
LR	Logistic Regression
LSTM	Long Short-Term Memory
LTC	Litecoin [cryptocurrency]
MACD	Moving Average Convergence Divergence
ML	Machine Learning
MLP	Multilayer Perceptron
QDA	Quadratic Discriminant Analysis
RF	Random Forest
RSI	Relative Strength Index
SMA	Simple Moving Average
SOL	Solana [cryptocurrency]
SVM	Support Vector Machine
SVR	Support Vector Regression
TA	Technical Analysis
TR	True Range
USDT	Tether [cryptocurrency]
UTC	Coordinated Universal Time
XGBoost	Extreme Gradient Boosting
XRP	Ripple [cryptocurrency]

1. Introduction

In recent years, econometrics, statistics, and corporate finance have become increasingly important within the field of industrial economics. One of the most dynamic areas within this intersection is the study of algorithmic trading, particularly its application to cryptocurrencies. This thesis brings together three contemporary and highly relevant topics: cryptocurrencies, algorithmic trading, and machine learning. By investigating the application of machine learning (ML) methods to the trading of digital assets, this research master thesis aims to contribute to both academic understanding and practical advancements in the field.

Most existing research focuses primarily on the classification task of predicting whether the price of a cryptocurrency will increase or decrease (see e.g. Anghel, 2020). However, there is no research on the use of the confidence score provided by machine learning classifiers in trading decisions. The motivation for this study is thus that incorporating the confidence score could potentially enhance prediction accuracy and, in extension, the overall profit-and-loss. Addressing this gap is significant both academically, by advancing the methodological toolkit available to researchers as well as examining existing theoretical market assumptions, and practically, by improving the decision-making process for traders.

The primary objective of this thesis is to evaluate whether incorporation of the confidence score of predictions from machine learning models can improve trading outcomes in algorithmic cryptocurrency trading. Specifically, we utilize an ensemble of two different Random Forest ML models to predict if the price of the asset will move up or down in the near future, aiming to increase the accuracy of the method through conformal prediction (Johansson et al. 2013). That is, by discarding predictions in the output where the model's confidence score does not meet a certain threshold. We expect that this approach will yield higher prediction accuracy than strategies that do not consider confidence scores, but at the cost of getting fewer predictive samples as output. To benchmark our results, we use the same Bitcoin historical sample dataset as Chen et al. (2020) and directly compare our prediction accuracy with their findings. The research question is:

Does incorporating predictions that meet a specific prediction score threshold improve overall accuracy in algorithmic cryptocurrency trading?

By addressing a gap in the literature regarding the use of confidence scores in ML-based trading strategies, this thesis contributes to the development of more robust and effective algorithmic trading models. Previous studies, such as Chen et al. (2020), have demonstrated the potential of utilisation of ML in predicting cryptocurrency price movements, but have not explored the added value of confidence-based decision making. Our research aims to fill this gap. This study focuses on two different time periods: July 17, 2017, to January 17, 2018, for the price of Bitcoin as a comparison to Chen et al., and the two full years of 2022 and 2023 for ten different cryptocurrencies as an additional broadening and verification of the method's validity. For both time periods we are utilising historical 5-minute sampled trading data downloaded from the ByBit cryptocurrency exchange API.

The thesis is organized as follows. First, we present the literature review, the benchmark paper, and a general overview of cryptocurrencies. Next, we describe the theoretical framework, including the efficient market hypothesis, technical analysis, and the Random Forest machine learning model. The subsequent chapter presents the data sources and data preparation procedures. Then, we outline the methodology. In the following section, we present the results in comparison to the benchmark paper as well as the multi-currency results over two years. Lastly, we conclude with a summary of key insights, limitations and suggestions for future research.

2. Literature Review

The cryptocurrency market, with its high volatility and speculative nature, has become an area of interest for algorithmic trading powered by machine learning (ML). Because large institutions are less involved, small firms have a better chance of beating the game. Researchers have applied various ML models to predict price movements, volatility, and returns, with the goal of developing profitable trading strategies. However, while significant progress has been made in forecasting price direction and volatility, previous research has not included confidence levels associated with the magnitude of price changes. Confidence measures or determination are critical for risk management and decision-making in algorithmic trading, yet this dimension remains unresearched. This review analyses existing research on ML applications in cryptocurrency trading to highlight improvements and limitations. The review concludes by identifying the research gap and outlining how this thesis will address it.

Our focus is on reviewing papers that address volatility forecasting, price prediction and other aspects of algorithmic trading. Particular attention is on how these papers deal with prediction confidence. We notice recurring methodological themes, making it easy to sort the research projects into distinct groups. These groups include comparisons between common statistical methods, machine learning, deep learning, ensemble methods, volatility forecasting and feature engineering.

Statistical models versus machine learning approaches

Two studies have compared traditional statistical models with ML techniques. Jang and Lee (2017) used the Bayesian Neural Networks[1] (BNNs) to forecast Bitcoin prices by using macroeconomic indicators and blockchain data. They have found that BNNs outperform traditional models such as linear regression and Support Vector Regression[2] (SVR) but did not explore confidence intervals for price forecasts. McNally et al. (2018) compared AutoRegressive Integrated Moving Average[3] (ARIMA) with Long Short-Term Memory[4] (LSTM) networks for Bitcoin price forecasts. Although LSTM has better accuracy than ARIMA, none of the approaches addressed uncertainty quantification for price changes.

[1] BNN is a stochastic artificial neural network that treats its weights as random variables and uses Bayesian inference to learn a posterior distribution over these parameters, enabling the model to estimate uncertainty in its predictions by comparing outputs from multiple sampled parameter sets (Jang & Lee, 2018)

[2] SVR is a machine learning technique that maps input data into a high-dimensional feature space using kernel functions and estimates a linear model in this space by minimizing an ε-insensitive loss function and model complexity, resulting in a regression function defined by a subset of training points called support vectors (Dudek et al., 2024)

[3] ARIMA model is a stochastic time series forecasting method defined by the number of autoregressive lags, the degree of differencing to achieve stationarity and the number of moving average lags that combines autoregressive and moving average polynomials to model and predict future data points based on historical observations (Al-Chalabi et al., 2018)

[4] LSTM is a neural network architecture that can choose which information to remember or forget based on the weight and importance of features, enabling it to effectively learn long-term dependencies in sequential data for tasks such as time series prediction (McNally et al., 2018)

Deep learning models

Deep learning models have been extensively used for capturing complex patterns. Lahmiri and Bekiros (2019) used LSTM networks to model the turbulent dynamics of cryptocurrency prices. The study confirmed the superiority of LSTM networks over Generalized Regression Neural Networks[5] (GRNN). However, it did not investigate the confidence levels of the forecasts. Chen et al. (2019) explored LSTM with simpler models like Logistic Regression[6] (LR) for Bitcoin price prediction on different timeframes. The LSTM model worked best on high-frequency settings or smaller timeframes. The study focused entirely on price direction without prediction confidence. Overall, deep learning models, such as XGBoost[7], Random Forest[8] (RF) and LSTM networks successfully capture complex patterns and improve predictive accuracy in cryptocurrency price forecasting. However, by focusing exclusively on accuracy, the reviewed literature ignores addressing uncertainty quantification and confidence levels.

Ensemble methods

Ensemble methods are well-known for their robustness. Jaquart et al. (2022) used ensemble strategies combined with RF, Gradient Boosting Classifiers[9] (GBC) and deep learning models such as LSTM and Gated Recurrent Units (GRU). RF will be described in more detail in the Random Forst chapter. The strategies investigated resulted in high Sharpe ratios, i.e. 3.23 for LSTM and 3.12 for GRU, but did not include probabilistic measures of prediction confidence. Sebastião and Godinho (2021) used ensemble strategies combining RF and Support Vector Machines (SVM) to generate trading signals in changing market conditions. Even though the ensemble strategies were profitable, the study did not quantify prediction certainty. Therefore, ensemble methods are robust and profitable in algorithmic trading, but they lack probabilistic measures or uncertainty quantification.

---

[5] GRNN is a parallel and memory-based neural network system that estimates the regression surface of a continuous variable, providing fast learning and convergence to the optimal regression surface as the sample size increases, and is based on Gaussian kernels for local approximation of non-stationary signals (Lahmiri & Bekiros, 2019)

[6] LR is a simple and efficiently trainable benchmark model that models the probability of a binary event occurring based on a linear combination of predictors, equivalent to linear regression for binary response variables, and can be represented as a single-layer neural network with a sigmoid activation function (Jaquart et al., 2022)

[7] XGBoost is a framework and library that parallelizes the growth of gradient boosted trees in a forest, aiming to minimize the time required to grow trees and optimize the process, which makes gradient boosting decision trees practical to use by combining the results of many weak classifiers to make strong predictions and summing the outputs of each decision tree (Chen et al., 2019)

[8] RFs are based on randomized decision trees and operate by creating and combining a diverse set of weak learners (decision trees), whose individual predictions are aggregated while tuning the maximum tree depth and the number of features used for splitting to optimize performance (Jaquart et al., 2022)

[9] GBCs build an ensemble of shallow decision trees sequentially, where each new tree is trained to correct the errors of the previous ensemble, thereby reducing bias in the combined model and generalizing the AdaBoost algorithm, as implemented in this study using scikit-learn with 100 estimators and tunable tree depth (Sebastiao & Godinho, 2021)

Volatility forecasting

Dudek et al. (2023) compared ML techniques such as Support Vector Regression (SVR) and Multilayer Perceptron[10] (MLP) with statistical models like Ridge Regression for predicting the weekly and daily volatility of cryptocurrencies. Although the SVR models performed very well, Dudek et al. (2023) did not address either the confidence intervals or the probabilistic forecasts of the volatility magnitudes.

Feature engineering

Feature selection plays a very important role in improving the model's performance due to the improved predictive accuracy. Jang and Lee (2017) merged blockchain-specific features with macroeconomic indicators to improve prediction accuracy, but did not use the same features for confidence estimation. Orte et al. (2022) found that candlestick patterns[11] outperformed technical indicators in predicting Bitcoin price direction, but did not investigate the probability of the forecast.

To summarise: While existing research has made significant progress in predicting cryptocurrency prices and volatility, none of the reviewed studies explicitly addresses the confidence levels or probabilistic estimates for the magnitude of price changes, and most studies focus on point predictions or binary classifications without quantifying uncertainty or reliability of forecasts. Finally, most studies focus on daily or hourly timeframes. Confidence measures could enhance algorithmic trading strategies by enabling risk-adjusted decision-making.

The reviewed literature also highlights significant advancements in applying machine learning techniques to cryptocurrency trading, especially in predicting price direction, volatility, and returns. BNNs, LSTMs, RFs, and ensemble methods are the leading approaches due to their ability to handle non-linear relationships and chaotic market dynamics effectively. This gap is significant because incorporating confidence measures can, on the one hand, improve decision-making in algorithmic trading by quantifying uncertainty and, on the other hand, potentially enhance returns by allowing for more informed risk management. Although the literature addresses multiple methodological themes within algorithmic trading, none of the studies have systematically examined whether incorporating only predictions that meet a certain confidence or prediction threshold can improve overall accuracy. Therefore, this thesis will specifically investigate: *Does incorporating predictions that meet a specific prediction threshold improve overall accuracy?* By focusing on this targeted research question, we aim to address a critical gap and contribute new insights to the field.

To address this gap, this thesis will focus on the usage and development of machine learning models that provide confidence scores for cryptocurrency price change predictions and incorporate the levels

[10] MLP is a neural network widely used for regression problems due to its universal approximation property, learning capability, nonlinear modeling, robustness, and fault tolerance (Dudek et al. 2023)

[11] Visual representations of price action that use the open, high, low, and close prices of an asset to form specific configurations signaling potential market reversals or continuations (Orte et al., 2022)

of confidence provided by the models into the trading strategies, thus aiming to improve profits and reducing risk as both prediction and confidence are considered.

2.1. Benchmark Paper

Our quantitative research is benchmarked against the prior study of Chen et al., 2020, "Bitcoin price prediction using machine learning: An approach to sample dimension engineering", since the paper combines feature engineering, statistical methods and machine learning algorithms to predict Bitcoin prices at different time granularities. Chen et al. (2020) was selected as the benchmark for this study because it addresses Bitcoin price prediction through feature engineering and multi-model comparisons across different time granularities. Their work aligns closely with our focus on ML applications in cryptocurrency algorithmic trading. Our focus particularly meets their use of 5-minute sample data. By achieving 67.2% accuracy with machine learning models on high-frequency data, they set a high baseline for price movement classification, while they did not investigate confidence score thresholds for the prediction, which is the central theme of our approach.

Chen et al. used two types of data granularities. The daily, low-frequency data includes Bitcoin property and network data such as block size, hash rate, and mining difficulty. Furthermore, trading and market data, such as market capitalization and transaction value. Moreover, media and investor attention metrics such as Google Trends, Baidu media search volume, and gold spot price. This data was downloaded from CoinMarketCap. The high-frequency, 5-minute Bitcoin data includes features such as volume, open, high, low, and close values. The data was downloaded from ByBit.

Chen et al. applied different models for daily and 5-minute price prediction. For the 5-minute data, they applied RF, XGBoost, Quadratic Discriminant Analysis[12] (QDA), Support Vector Machine[13] (SVM) and LSTM neural networks. Their experiment was designed to split the dataset, with 75% used for training and 25% for testing. Default parameters in Python were used for most models, with Keras for LSTM and scikit-learn for the rest.

[12] QDA is a supervised classification method that assumes each class is normally distributed and, unlike linear discriminant analysis, allows for quadratic decision boundaries between classes, with class-specific parameters estimated from training data using maximum likelihood estimation (Chen et al., 2020)

[13] SVMs are machine learning methods for binary classification that find an optimal separating hyperplane with maximum margin in a feature space, where the solution is a linear combination of support vectors identified through quadratic programming (Chen et al., 2020)

The performance of their models was evaluated using a confusion matrix[14] and standard classification metrics such as accuracy[15], precision[16], recall[17], and F1-score[18]. Standard classification metrics will be described in more detail in the Method chapter.

The study compared the performance of statistical and machine learning models on both daily and 5-minute datasets. The results were benchmarked against prior studies such as McNally et al. (2018). They found that both the low-frequency statistical methods and the high-frequency machine learning methods achieved similar accuracy, with 66% and 67% respectively.

2.2. Crypto Outlook

Bitcoin was introduced in January 2009, following the global financial crisis of 2008, to express its creator's distrust of the centralized global financial system (Nakamoto, 2008). Bitcoin is a decentralized digital currency that allows payments to be sent directly from one party to another without the need for a trusted third party, such as a bank (Nakamoto, 2008). It uses a peer-to-peer network and a proof-of-work mechanism to address the double-spending problem, ensuring that a digital currency cannot be spent more than once (Nakamoto, 2008).

In the beginning, when the Bitcoin network launched, there was no established monetary value for the coin. It was exchanged informally among early adopters with effectively zero value (Bitget, 2024). In October 2009, the NewLibertyStandard exchange valued 1,309.03 BTC at $1, based on the electricity cost to mine it (Bitget, 2024). The first real-world transaction occurred in May 2010, when Laszlo Hanyecz paid 10,000 BTC for two pizzas, valuing Bitcoin at about $0.0041 per coin at that time (Bitget, 2024). Bitcoin began trading on public exchanges in mid-2010 at around $0.08 (Bitget, 2024). As of May 10, 2025, the price of Bitcoin is $104,788.8.

Unlike fiat currencies, which can be printed in unlimited quantities by central banks, the number of Bitcoins that can be created is limited to 21 million coins (Nakamoto, 2008). Therefore, it is fundamentally deflationary. As more people own Bitcoin while the supply remains fixed, scarcity will drive up the price.

As of April 2025, there are approximately 17,134 cryptocurrencies in existence (Howarth, 2023). However, not all of these are active or hold significant value. When discounting inactive or "dead" cryptocurrencies, the number of active cryptocurrencies is around 10,385 (Howarth, 2023).

---

[14] Confusion matrix is a tabular tool for evaluating classifier performance that displays the counts of true positives, false positives, true negatives, and false negatives, providing the foundation for calculating key metrics such as accuracy, precision, recall, sensitivity, specificity, false positive rate, and F1 score (Sathyanarayanan, 2024)

[15] Accuracy is the proportion of true results (both true positives and true negatives) among the total number of cases examined (Powers, 2020).

[16] Precision is the proportion of true positive results among all positive results predicted by the model (Powers, 2020).

[17] Recall is the proportion of true positive results among all actual positive cases (Powers, 2020).

[18] F1-Score is the harmonic mean of precision and recall, providing a balance between the two metrics (Powers, 2020).

3. Theoretical Framework and Empirical Strategy

3.1. Efficient Market Hypothesis

The Efficient Market Hypothesis (EMH) is a fundamental concept in modern financial theory, asserting that asset prices in financial markets instantly and fully reflect all available information that would influence their valuation. In essence, this means that assets are always correctly priced with no delay in time. Assuming that the EMH hold through, and under ideal market conditions, no investor can consistently achieve excess returns above the market average through trading based on past or present information about the asset, since any new information is immediately incorporated into the asset price (Fama, 1970)

Fama outlines three forms of market efficiency:

- **Weak-form efficiency:** Current asset prices reflect all historical market data, such as past prices and volumes.
- **Semi-strong efficiency:** prices also reflect all publicly available fundamental information (e.g., earnings, macroeconomic news, political events).
- **Strong-form efficiency:** prices also incorporate private or insider information.

In a strong-form efficient market, the implications for predictive modelling and active trading are heavy, essentially meaning that returns exceeding those of the underlying asset are only due to random variations and cannot be consistently achieved through the exploitation of historical or public information (Jensen, 1978). As a consequence, methods such as Technical Analysis (TA), which rely solely on identifying recurring patterns or trends in historical price data, would be theoretically invalid under the assumptions of even weak-form market efficiency. However, EMH has been subject to increasing empirical investigation and criticism as numerous studies have identified persistent return anomalies and predictable price patterns that challenge its core assumptions, particularly in emerging or less regulated markets (Lo, 2004; De Bondt & Thaler, 1985). In such settings, informational frictions, behavioural biases, and market microstructure inefficiencies may prevent prices from adjusting directly or fully to new information. This is especially relevant for cryptocurrency markets, which differ from traditional asset classes in several key aspects, such as higher volatility, lower liquidity, no intrinsic or underlying value and less institutional oversight (Baur et al., 2018). Lo (2004) argues that while the market and asset pricing can be efficient on average, there will always be inefficiencies and overreactions in the market, both due to the behaviour of traders – who do not always act rationally – and due to the constant evolution where market actors continue to experiment and compete. Thus, there will be temporary occurrences where the market is not even weak-form efficient, and methods such as TA can serve as a way to identify those discrepancies between current asset value and the true efficient value (Murphy, 1999). To conclude, the Efficient Market Hypothesis (EMH) points out that even for weak-form efficiency, future market prices should not be possible to predict using historical trading data alone. However more recent research, such as the Adaptive Market Hypothesis (AMH) introduced by Lo, suggests that – specifically in less mature markets – there will be opportunities where the price is not instantly efficient. Our goal is to investigate if market movements can be predicted using modern methods such as TA and ML, thus invalidating the strictness of EMH for the evolutionary approach of AMH, specifically when applied to a relatively new market such as cryptocurrency trading.

3.2. Empirical Strategy for Technical Analysis

Efficient Market Hypothesis provides a theoretical framework which suggests that consistently outperforming the market through analysis of historical or public information is unachievable (Fama, 1970). However, deviation from perfect efficiency can often be found in cryptocurrency markets (Urquhart, 2016). One of the most widely used approaches for identifying and capitalizing on such inefficiencies is Technical Analysis (TA) (Murphy, 1999). TA is built on the assumption that historical price and volume data contain information about future price movements in markets where anomalies and patterns persist.

In order to develop effective predictive models in financial markets, it is essential to select and engineer meaningful input variables. Features or Technical Indicators are mathematical calculations derived from historical data such as price, volume, and open interest. They are used in technical analysis to predict future price movements and identify trading opportunities in the financial markets. These indicators help traders understand market trends, momentum, and potential reversals by analysing past market movements. We use the technical indicators listed below since our preliminary investigation and sensitivity analysis found them to be the most meaningful out of the 43 technical indicators considered for machine learning purposes.

3.3. Technical Indicators Used in the Technical Analysis

Average True Range

The Average True Range (ATR) is a technical indicator used in financial markets to measure the volatility of an asset's price (Murphy, 1999). ATR does not indicate the direction of price movements but quantifies the magnitude of price fluctuations over a specified period (Murphy, 1999). The true range is the difference between the candle's high, low and previous close. It is calculated by summating the true range values for each period and then dividing the total by the number of periods. The true range is defined as the largest value of the following: the current high minus the current low, the absolute value of the current high minus the previous close, or the absolute value of the current low minus the previous close. ATR is used to assess market volatility for better risk management and to set stop-loss and take-profit levels that adapt to current volatility, rather than relying on fixed amounts (Murphy, 1999).

$$ATR = \frac{TR_1 + TR_2 + \cdots + TR_n}{n}$$

Average Directional Index

The Average Directional Index (ADX) is a technical indicator used in financial markets to measure the strength of a trend, regardless of its direction (Kaufman, 2013). It includes the Positive Directional Index (+DI) and the Negative Directional Index (-DI). The +DI and –DI indicate whether the momentum is positive or negative (i.e., the direction of the price movement).

The negative directional movement measures the strength of a downward trend. It is calculated by dividing the smoothed -DM by the ATR and multiplying it by 100 to express the result as a percentage.

$$-DI = \left(\frac{Smoothed\ -\ DM}{ATR}\right) \times 100$$

The positive directional movement measures the strength of an upward trend. It is calculated by dividing the smoothed +DM by the ATR and multiplying it by 100 to express the result as a percentage.

$$+DI = \left(\frac{Smoothed\ +\ DM}{ATR}\right) \times 100$$

The ADX for each period is calculated by taking the absolute value of (+DI minus -DI), dividing it by the sum of +DI and -DI, and multiplying by 100.

$$ADX = 100 \times \frac{Smoothed\ Moving\ Average\ of\ |+DI\ -\ -DI|}{+DI\ +\ -DI}$$

Balance of Power

The Balance of Power (BOP) indicator is a technical analysis tool used to measure the overall strength of buyers and sellers in the financial markets (TradingView, 2025). It is used to identify trends, potential reversals, overbought or oversold conditions and market sentiment by analyzing the relationship between opening and closing prices relative to the high and low prices of a trading period (TradingView, 2025). It is calculated by dividing the difference between the opening and the closing price of the period by the high and low of the period.

$$BOP = \frac{Close\ -\ Open}{High\ -\ Low}$$

Simple Moving Average

The Simple Moving Average (SMA) is a technical indicator used in financial markets to calculate the average price of a security or cryptocurrency over a specified period of time (Murphy, 1999). It is used to determine whether an asset is in an uptrend or downtrend by comparing current prices to the SMA value at the same time (Murphy, 1999). The SMA of X is the last X candles' equally weighted average closing price. It is one of the most basic types of moving averages and is widely used in technical analysis to identify trends. It is calculated by adding up the closing prices for a number of periods and dividing them by the number of periods.

$$SMA\ =\ \frac{A_1 + A_2 + \cdots + A_n}{n}$$

Exponential Moving Average

The Exponential Moving Average (EMA) is a technical indicator used in financial markets to track the price movement of a security or cryptocurrency over a specified period of time (Murphy, 1999). It is used to generate buy and sell signals based on price crossovers with the EMA, determining dynamic support and resistance levels (Murphy, 1999). Unlike the Simple Moving Average (SMA), which gives equal weight to all data points, the EMA places more weight on recent prices, making it more responsive to recent price changes (Murphy, 1999). It is calculated by first establishing an initial value using an SMA and then applying a multiplier to each new price, so that recent data is given greater weight.

$$EMA = close \times multiplier + previous\ EMA \times (1 - multiplier)$$

Percent Return

The percent return calculates the percent return of a series of prices. It is calculated by subtracting the closing price of a period from the closing price of a later period to get the price change, then dividing by the later period's price change to get the relative change, and the result is multiplied by 100 to express it as a percentage.

$$PCTRET = \left(1 + \frac{close_t - close_0}{close_0}\right) - 1$$

Log Return

The log return calculates the logarithmic return of a series of prices. It is calculated by subtracting the closing price at the start period from the closing price of the end period and dividing the result by the closing price of the end period to get the relative change. 1 is added to account for both positive and negative results, and then the natural logarithm is applied.

$$LOGRET = ln\left(1 + \frac{close_t - close_{t-length}}{close_{t-length}}\right)$$

Lyapunov Exponent

The Lyapunov Exponent is a concept from chaos theory that measures the rate of separation of infinitesimally close trajectories in a dynamical system (Young, 2013). In financial markets, it is used to assess the degree of chaos or unpredictability in price movements (Young, 2013). In trading, the Lyapunov exponent is used to analyse market stability, predict volatility shifts, and identify chaotic behaviour, which can signal impending market crashes or high-risk periods (Soloviev et al., 2020). The Lyapunov exponent is calculated by averaging the natural logarithm of the ratio of the separation between two initially close trajectories at successive time intervals, quantifying how quickly they diverge or converge over time (Young, 2013).

$$\lambda = \frac{1}{t}\sum_{i=0}^{n-1} \ln\left(\frac{\delta(t_i + \tau)}{\delta(t_i)}\right)$$

Moving Average Convergence Divergence (MACD)

The Moving Average Convergence Divergence (MACD) is a technical indicator used in financial markets to identify trends and measure momentum (Murphy, 1999). It is based on the relationship between two exponential moving averages (EMAs) of a security's or cryptocurrency's price (Murphy, 1999). It is calculated by taking the difference between the 12-period EMA and the 26-period EMA of a security's closing prices.

$$MACD = 12 - period\ EMA\ -\ 26 - period\ EMA$$

Relative Strength Index (RSI)

The Relative Strength Index (RSI) is a momentum oscillator used in technical analysis to measure the speed and magnitude of recent price changes (Murphy, 1999). It helps identify overbought and oversold conditions in financial markets, which can signal potential trend reversals or continuations. It is calculated by dividing the average gain by the average loss over a set period, converting this ratio into a relative strength value to produce a value between 0 and 100.

$$RSI = 100 - \left(\frac{100}{1 + \frac{AvgU}{AvgD}}\right)$$

Where:

AvgU is the average gain over the specified period and

AvgD is the average loss over the same period.

Volatility

In technical analysis, volatility refers to the degree of variation in the price of a financial instrument over a specific period (Kaufman, 2013). It measures how much and how quickly prices fluctuate, indicating the level of risk or uncertainty associated with an asset (Kaufman, 2013). High volatility means large price swings, while low volatility indicates more stable prices. It is calculated by multiplying the standard deviation of returns for a base period by the square root of the ratio of the target period length to the base period length.

$$\sigma_{annualized} = \sigma_{daily} \times \sqrt{T}$$

Where:

σ_T = volatility over the target period T,

σ_{base} = volatility over the base period τ,

T = length of the target period (in the same units as τ),

τ = length of the base period (e.g., 1 day, 1 hour, 1 month).

Volume

Volume refers to the total number of securities or cryptocurrency coins that are traded during a specific period (Murphy, 1999). The period is defined by the observing timeframe, for example, 5 minutes or a daily sample. It is a key indicator of market activity and liquidity, providing insights into market sentiment and the buying or selling pressure of price movements (Murphy, 1999).

$$Trading\ volume = \sum_{i=1}^{n} Number\ \ of\ coins\ traded\ in\ each\ transaction$$

3.4. Using Machine Learning for Technical Analysis

Following the Adaptive Market Hypothesis, which suggests that not all markets are immediately effective, and the toolkit that Technical Analysis provides to attempt to identify those actionable market patterns, we turn to modern data science algorithms specifically developed to solve large scale classification problems – Machine Learning (ML). While the field and theory of ML has a long history, tightly intertwined to the development of computers and computer science, ML usage and development have been hugely boosted by the fast advancements in computer performance that we have seen over the past 10 – 20 years (Géron, 2019). This makes ML methods and models highly suitable for processing and interpreting large amounts of information, such as minute-scale historical market data, at a rate that no human can compete with. A typical ML classification model takes a large number of input samples, containing some kind of information ("features") and a predefined set of corresponding classes ("target variable") as training data, and can, once training has completed, be used to predict the output classification of new samples where only the input features, and not the classification, is provided (Goodfellow et al., 2016). This approach is successfully used in such varying cases as email spam filtration, image recognition and fraud detection, and our aim is to apply this technique also to market data and price prediction, thus automating the methods developed within TA. One of the more commonly used ML classification models today, known for its robustness, speed, and high predictive power, is Random Forest, which we describe in detail below (Hastie et al., 2009).

3.5. Random Forest

Following Breiman (2001), this chapter provides an overview of Random Forest(s), detailing the theoretical foundation, algorithmic structure, and practical advantages for classification tasks in financial prediction. Random Forest are supervised machine learning algorithms that create and combine decision trees for regression and classification tasks. Regression is used when the goal is to predict a continuous quantity; for example, it can be used to determine a security's future price based on historical price data. Classification, on the other hand, is used when the goal is to assign input data to two or more predefined categories. An example of this is classifying images as cats or dogs.

In our case, classification is used to determine whether the future price will be higher or lower compared to the current price, meaning that there are exactly two predefined categories, “down” or “up”, making it a binary classification problem. In our model, only classification is utilised.

The algorithm creates many decision trees (a "forest"), which is where the name Random Forest comes from. A decision tree is a hierarchical model that utilizes nodes and leaves by splitting data based on feature values. It starts with a root node representing the entire dataset. There are decision nodes in between that split the data using rules, and branches that connect these nodes, representing outcomes such as "up" or "down." The final predictions are made at the leaf nodes. Figure 1illustrates a decision tree.

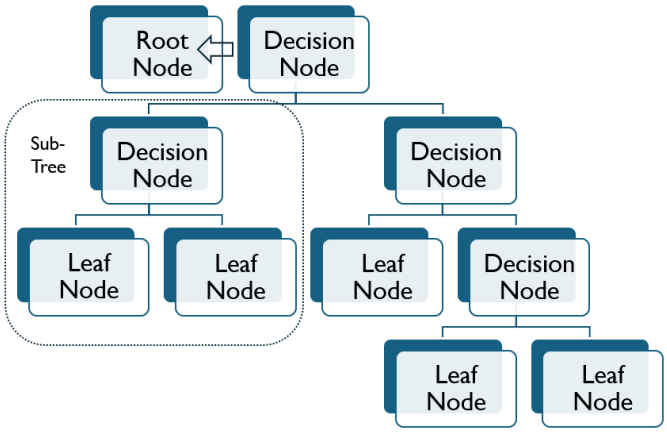


Figure 1: Schematic representation of a RF decision tree

These decision trees are each trained on a random subset of the training data. For each tree, a random subset of features is selected at each split to introduce further randomness and reduce correlation between the trees.

To overcome the two major problems with decision trees, high variance and overfitting, bagging and bootstrapping techniques (see e.g. Breiman, 2001) are applied to improve prediction accuracy. Bootstrapping is a statistical resampling technique that uses random samples from the dataset with replacement. Bagging applies the bootstrapping technique to machine learning algorithms by creating multiple bootstrap samples and training a separate decision tree on each sample. It then aggregates the predictions to make the final prediction. In classification, the majority vote is selected, and in regression, the average prediction is used.

The main reason we have chosen Random Forest as our model is due to their low risk of overfitting. Overfitting is a common problem where the model performs well on the training data but poorly on new data. Random Forest use ensemble learning – building many decision trees on different data subsets – to improve generalisation to unseen data. Furthermore, Random Forest can handle the complex and non-linear relationships that are present in financial markets. They are also capable of processing large and high-dimensional datasets with many features. Random Forest can automatically identify the most important variables for prediction, which supports feature selection and interpretability. In addition, financial data often contains both numerical and categorical variables as well as missing values, and Random Forest are robust to incomplete data. Studies have also shown that Random Forest outperform traditional models and other machine learning techniques in predicting securities' price movements (Biau, 2012). Lastly, a key feature of this model is its ability to quantify feature importance. Random Forest can measure the importance of each feature in making predictions, which aids both interpretability and feature selection.

4. Data

This chapter presents the data used in our study for the algorithmic cryptocurrency trading strategy. Our research focuses on the utilization of machine learning algorithms to make confidence-based trading decisions. To achieve this, we are using high-frequency data from the cryptocurrency market. The data was obtained from Bybit, the second largest cryptocurrency exchange by trading volume (CoinMarketCap, 2024). The covered period is from July 17, 2017, to January 17, 2018 for Bitcoin. Furthermore, we collected the data for 10 cryptocurrency USDT pairs on the 5-minute timeframe to provide an extensive dataset for our analysis. The time frame for the collected data was January 1, 2022 until December 31, 2023.

Our data was collected directly from ByBit API, which has a historical price and volume database for cryptocurrency trading pairs. We focused on USDT pairs, as they represent a significant portion of the cryptocurrency trading market and provide a stable pricing relation to the US dollar. The data collection process involved identifying 10 USDT trading pairs available on ByBit, which were suitable for our analysis. USDT is a stablecoin cryptocurrency designed to maintain 1:1 value with the US dollar (Bitnob Technologies, 2024). Unlike classic cryptocurrencies such as Bitcoin, the value of stablecoins doesn't fluctuate significantly and can act as a digital savings tool (Bitnob Technologies, 2024). The coins in our analysis are Bitcoin (BTC), Ethereum (ETH), Litecoin (LTC), Solana (SOL), Ripple (XRP), Dogecoin (DOGE), Cardano (ADA), Curve DAO Token (CRV), Avalanche (AVAX) and Chainlink (LINK). These coins were chosen for this study as they had sufficient historical data, and were among the most traded during the investigated time period. The ByBit API was queried for historical data on each pair to collect data at the 5-minute timeframe and finally, storing the data in a structured format for further analysis.

The raw data downloaded from the cryptocurrency exchange required several processing steps to guarantee its quality and usability for our analysis. During data cleaning, we removed any duplicate entries and checked for missing values. Subsequently, the time zones were standardised. All timestamps were converted to Coordinated Universal Time (UTC) to ensure consistency in the dataset. Also, we performed checks to confirm the integrity of the historical price and volume data, highlighting any values that required further investigation. Finally, we calculated additional metrics like Moving Average Convergence Divergence (MACD), Relative Strength Index (RSI) and Average Directional Index (ADX), which are used as features in the research. For data preparation and analysis, we used Python. Furthermore, Pandas libraries which is an open-source Python package developed for data manipulation and analysis as well as NumPy for numerical calculations.

The dataset includes trading data for BTC for 6 months and 10 cryptocurrency USDT pairs over a period of 24 months. The following variables are included in our research for each pair. Timestamp, open price, high price, low price, close price and volume. In trading, a 5-minute candle refers to a specific visual chart representation, where each candle depicts the price movement of a cryptocurrency USDT pair over a five-minute period. Figure 2 shows two candles. The 5-minute candle starts trading at the top of each 5-minute interval within an hour, starting at 00:00, 00:05, 00:10, 00:15, etc. The duration of the candle is fixed, meaning every 5 minutes, a new one is forming to reflect the latest price action. Firstly, key characteristics of a 5-minute candle include:

- the open price, the price at which the cryptocurrency USDT pair starts trading at the beginning of the 5-minute interval
- the closing price, the price at which the cryptocurrency finishes trading
- the high price, the highest price reached during the 5-minute period
- the low price, the lowest price reached during the 5-minute period

The volume of a candle represents the total number of contracts that were traded during the time period of that candle. To provide an overview of our dataset, a summary is presented in Table 1.

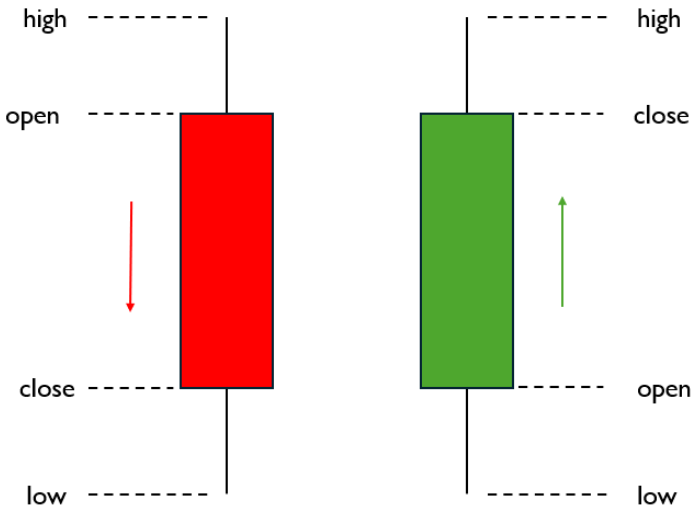


Figure 2: Bearish (red) & bullish (green) candlestick representation

Pair	Timestamp	Open	High	Low	Close	Volume
BTC/USDT	2017-07-18 01:00	2196.26	2196.26	2188.02	2190.77	73.1790
BTC/USDT	2017-07-18 01:05	2190.77	2190.77	2166.64	2174.09	72.6986
BTC/USDT	2017-07-18 01:10	2174.09	2174.09	2166.64	2166.65	32.7133
BTC/USDT	2017-07-18 01:15	2166.65	2169.99	2165.00	2165.78	36.4019

Table 1: The 5-minute price data of Bitcoin

We observed substantial variations in price and volume across different cryptocurrencies, reflecting the high volatility of the crypto market. Bitcoin and Ethereum showed the highest price levels and the highest 24-hour volumes in USDT among the investigated pairs.

5. Method

While the Efficient Market Hypothesis typically holds true for assets and markets in general, and over time, we have set out to utilize machine learning with the goal of identifying occurrences where the invisible hand (Smith, A., 1776) of the market has not yet efficiently priced an asset correctly, i.e. to predict if the price of an asset will move up or down in the future based only on historical trading data for that asset. With inspiration from the field of Technical Analysis (TA), we suggest a method utilizing a selection of technical indicators as features fed into a machine learning algorithm, in our case, two parallel binary-class Random Forest Classifiers. The goal of this method is to predict at time t if the price will have moved up or down at time t + Δ, where Δ is a positive number, thus predicting future price movements. For this, we suggest a procedure as described below.

5.1. Procedure

The method applied to the data samples is to use a combination of short training and test windows, frequent rolling retraining and a mini-ensemble of two binary-class Random Forest models: RF_1 and RF_2. Both these models are executed in parallel using the same features but with different configurations to try and capture different signals in each model, then using their combined predictions and confidence scores to enhance prediction accuracy. Figure 3 provides an illustrative overview of how the sequential training and evaluation of the models have been carried out.

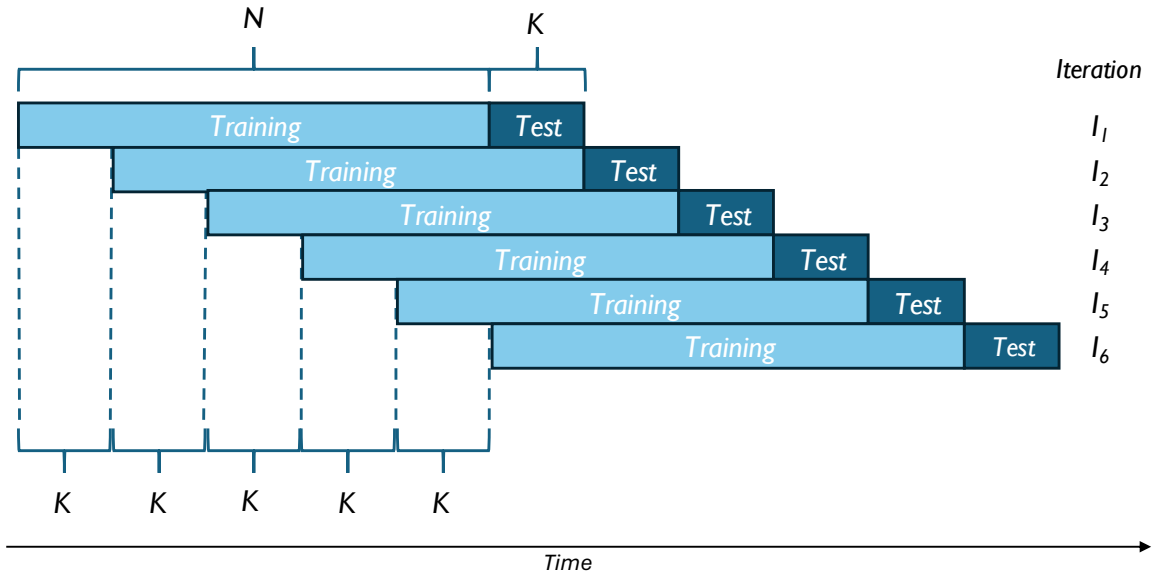


Figure 3: Illustrative description of how the sequential training and evaluation of the models has been carried out. In this illustration N = 5 • K, meaning that for iteration I_6, the training window N will consist of data from iterations I_1 through I_5.

N describes the length of the training data window, and *K* the length of the test/evaluation data window. This means that the first *N* samples are only used for training, while all the consecutive samples are first used as test data exactly one time each and then included in the training window for the upcoming N/K iterations. Both models are trained to predict if the closing price for sample *n+60* will be higher or lower than the closing price for the current sample *n*, meaning that when the models are running

on 1-minute samples, they predict the price direction 60 samples = 60 minutes into the future. For 5-minute samples, they instead predict the price direction for 5 • 60 = 300 minutes later. To further increase the accuracy, we also apply a threshold *T* to the combined confidence score of the two models, discarding any predictions where the models either predict different price directions, or where the combined confidence score is below the threshold T. This usage of confidence scores to achieve better predictions is typically referred to as conformal prediction and aims to increase the prediction accuracy at the cost of reduced size of the output prediction set (Johansson et al., 2013). As both RF_1 and RF_2 are binary classifiers, the confidence score of each prediction will be in the range 0.5 – 1.0, since the sum of confidence scores for all possible classes is exactly 1 for all predictions, and the predicted class is based on the class with the highest confidence score. This means that the threshold for ConfidenceScore(RF_1) + ConfidenceScore(RF_2) should be greater than 1, but always at or below 2. All predictions where the models predict different outcomes or where the threshold is not met will fall into the "inconclusive" outcome class. Following this, we get a three-class output space with the classes {"down", "up", "inconclusive"}, where the "inconclusive" class contains all predictions that are discarded as not reliable enough, as the models RF_1 and RF_2 give different predictions. When also applying a threshold *T* to the combined confidence score, the rate of "inconclusive" predictions will become higher, and the size of the remaining output prediction set will decrease correspondingly.

Example 6.1

> Models RF_1 and RF_2 are used at sample n to predict if the price will have moved up or down at sample n+60. The confidence scores for the two classes {"down", "up"} for RF_1 are {0.35, 0.65}, and for RF_2 they are {0.15, 0.85}. This means that both models predict that the price will go up, RF_1 with confidence score 0.65 and RF_2 with confidence score 0.85. The added confidence score of both models is 0.65 + 0.85 = 1.50, meaning that for all values of $T \leq 1.50$, this prediction will be included in the output prediction set. For all $T > 1.5$, the prediction will not be included in the output, and the prediction for sample n would instead be considered "inconclusive".

For increasing values of T, there will be more and more predictions that fall into the "inconclusive" category, meaning that there are fewer predictions in the "down" and "up" classes and thus fewer opportunities to do a trade based on the predictions. However, as we are focusing on the smaller sample sizes of 1- or 5-minute trading data, there is seldom a reason to perform a trade every 1 or 5 minutes. As an example: in the case of 1-minute samples, even if 59 out of 60 (~98.3%) of the samples are classified as "inconclusive", we still have, on average, one possible trade to make every hour, or one every 5 hours in the 5-minute sample case. As prediction accuracy increases, we consider this to be a valid trade-off in our method.

5.2. Evaluation metrics

Typical metrics to evaluate a binary ML-classifier are accuracy, precision and recall, which are all defined below. While accuracy is a measurement of the overall correctness of the model, in simple terms the number of correct predictions divided by the total number of output predictions, precision and recall both focuses on the ratio of different subsets of the predictions. While precision and recall are highly relevant in cases where the cost of false positives (e.g. spam detection), respectively false negatives (e.g. cancer detection) is high, they are not as relevant as accuracy in our case. This is due to a false negative being precisely as "costly" as a false positive. Correspondingly, a true positive is just as important as a true negative, as we aim to find the correct direction to trade in, independently of

whether that is "up" or "down". As our goal is to increase accuracy while still maintaining as many predictions in the output as possible, the two measurements accuracy and remaining data are our main metrics when evaluating the results.

Definitions

Accuracy is the proportion of true results (both true positives and true negatives) among the total number of cases examined (Powers, 2020).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Where,

TP = True positive

TN = True negative

FP = False positive

FN = False negative

Precision is the proportion of true positive results among all positive results predicted by the model (Powers, 2020).

$$Precision = \frac{TP}{TP + FP}$$

Recall is the proportion of true positive results among all actual positive cases (Powers, 2020).

$$Recall = \frac{TP}{TP + FN}$$

Remaining data describes the ratio between the size of the output prediction set compared to the amount of total input samples, telling us how many of the predictions that are still classified as "up" or "down" when first verifying that both models gave the same prediction, and then applying a threshold to further limit the predictions to those with highest likelihood of being accurate.

$$Remaining\ data = \frac{Size(output)}{Size(input)}$$

6. Results

In order to test the hypothesis that combining two model outcomes ensemble-style and applying confidence score thresholds can be used to increase the accuracy of price movements, we have firstly chosen to apply our method on the same sample rate and time frame as Chen et al. (2020), i.e. 5-minute samples of the BTC price over the period 2017-07-17 to 2018-01-17. Secondly, we have applied our method to ten different cryptocurrencies on the 2022-01-01 to 2023-12-31 time range, spanning over two years of data, also here using 5-minute samples. This time range was chosen as it represents a period where the crypto market was both bullish (moving up) and bearish (moving down) over prolonged periods of time, and thus can be used to evaluate the method's performance in both increasing and decreasing markets. It is also aimed at investigating whether the method is applicable to more recent data, as market conditions and trading behaviour could have changed since the initial benchmark period. All simulations in this work have been carried out with Python 3.12 on a modern laptop with an Intel Core Ultra 7 CPU and 32 Gb RAM.

6.1. Benchmark Period Results (July 2017 – January 2018)

This period contains 53 280 samples of 5-minute BTC trading data, and while Chen et al. do not specify how far into the future they are predicting the price direction, we decided to maintain our n+60 target variable setup, i.e. that for sample n we predict if the price will have moved up or down in sample n+60. For the iterative rolling window approach, we are using a 7-day training window (N = 2016 samples) and a 5-hour test window (K = 60). This gives 854 iterations in total, and the resulting metrics for all iterations of the simulation are presented in Table 2.

Model	Accuracy			Precision			Recall		
	Average	Standard deviation	Variance	Average	Standard deviation	Variance	Average	Standard deviation	Variance
RF_1	0.614	0.275	0.08	0.567	0.403	0.163	0.632	0.412	0.169
RF_2	0.631	0.265	0.07	0.568	0.403	0.163	0.623	0.412	0.170

Table 2: Average accuracy, precision and recall for the two models RF_1 and RF_2, which together constitute the ensemble model detailed in the method chapter.

As observed, both underlying models perform very similarly in terms of precision and recall, with RF_2 having a slightly higher average accuracy compared to RF_1, and both models are a few percentage points below the 65-67% accuracy presented by Chen et al. When combining the predictions of the two binary models into the three-class outcome space where all predictions that differ between the models are classified as "inconclusive", we end up with approximately 56% correct, 31% incorrect and 13% inconclusive (see Figure 4), giving a combined accuracy of 64%, which is slightly higher than the models by themselves, but not significantly so.

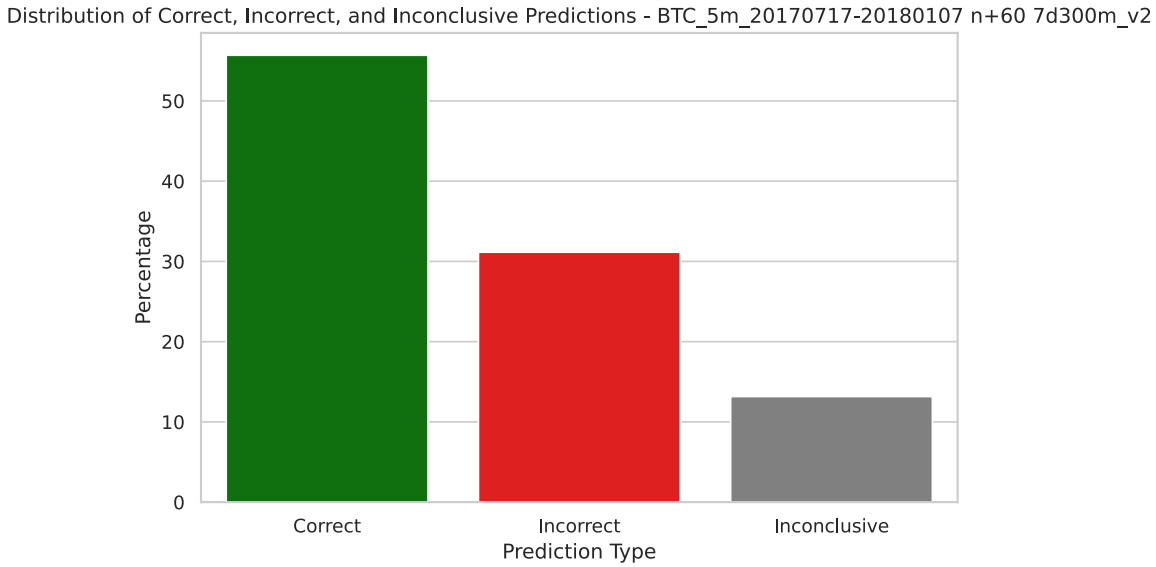


Figure 4: Distribution of correct, incorrect and inconclusive predictions for the benchmark period samples

Taking also confidence scores into consideration, as shown on the y-axis in Figure 5, we observe how the total confidence scores for the correct predictions are, on average, higher than the total confidence scores for the incorrect predictions, thus indicating that higher confidence scores will give a higher total accuracy.

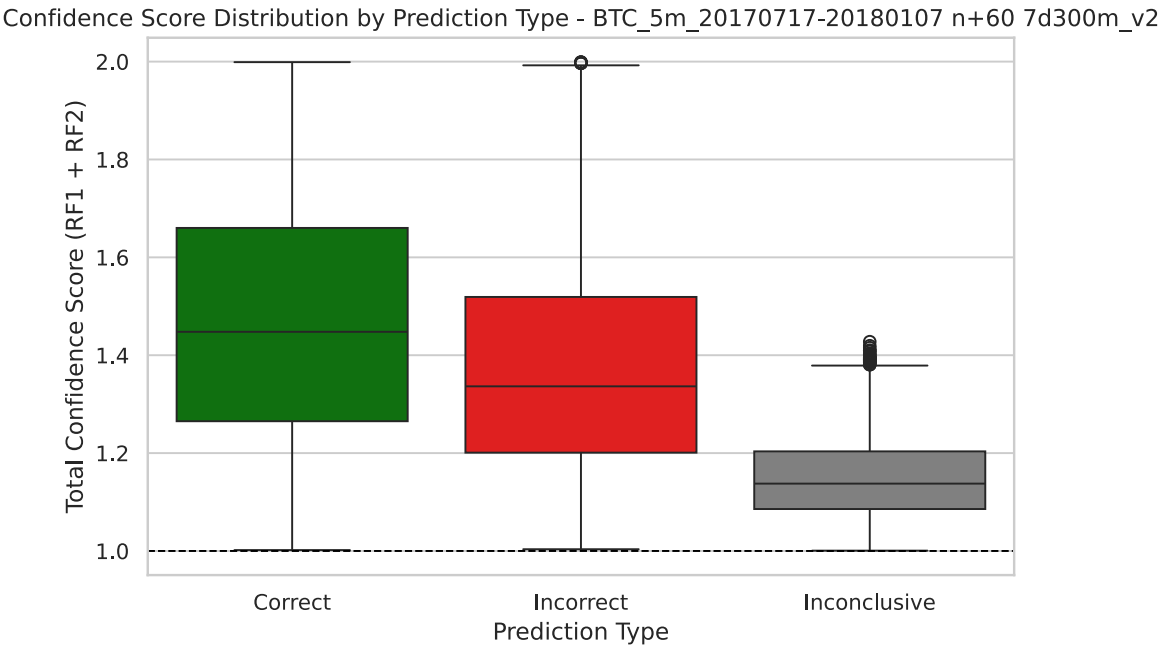


Figure 5: Boxplot of correct, incorrect and inconclusive predictions for the benchmark period samples and their distribution across the total confidence score of models RF_1 + RF_2

To investigate further, we observe how accuracy changes over different bins of confidence scores, both for the two models separately and combined. In Figure 6 and Figure 7, the number of correct respective incorrect predictions has been binned per confidence score for the two models RF_1 and RF_2, respectively, and in Figure 8 for the combination of the models – in the case where both models gave the same prediction. Here we clearly see that for confidence scores of ~0.75-0.80 for the RF_1 and RF_2 models and an aggregated confidence score of ~1.5-1.6 for the combination of models we are well above the Chen et al. benchmark top result of ~67% accuracy. However, and as noted earlier, this increase in accuracy comes with the cost of discarding a large number of the predictions, which leads to fewer trading opportunities. This effect is particularly impactful when decreasing the sampling frequency from 1 to 5 minutes or further.

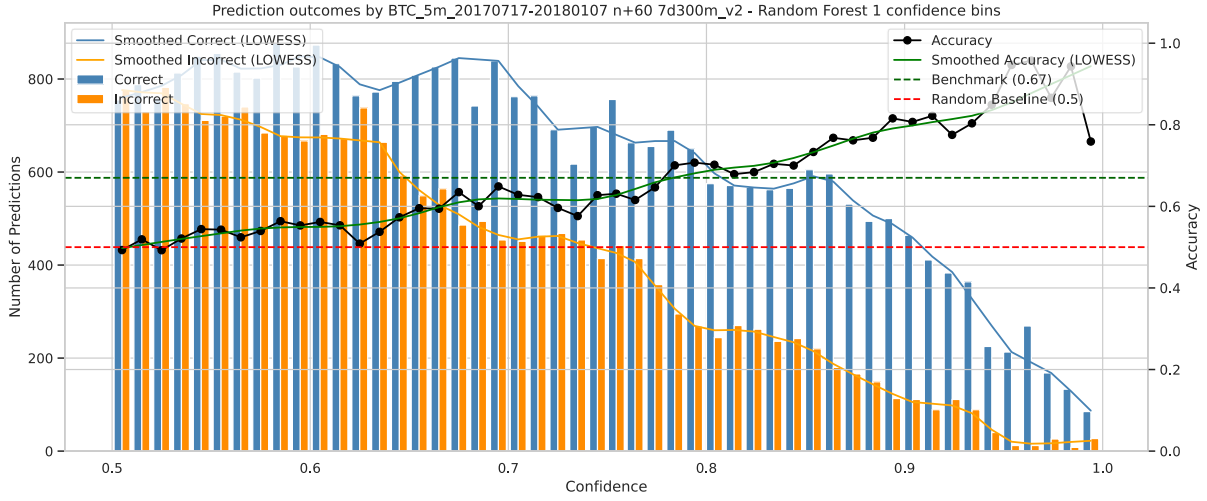


Figure 6: Number of correct versus incorrect predictions over increasing bins of confidence score for the RF_1 model

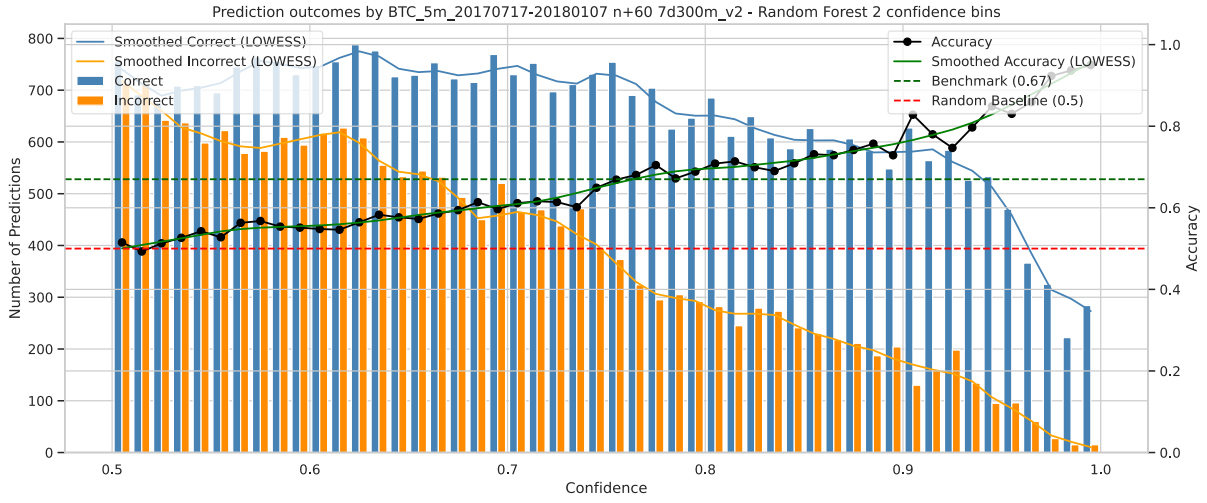


Figure 7: Number of correct versus incorrect predictions over increasing bins of confidence score for the RF_2 model

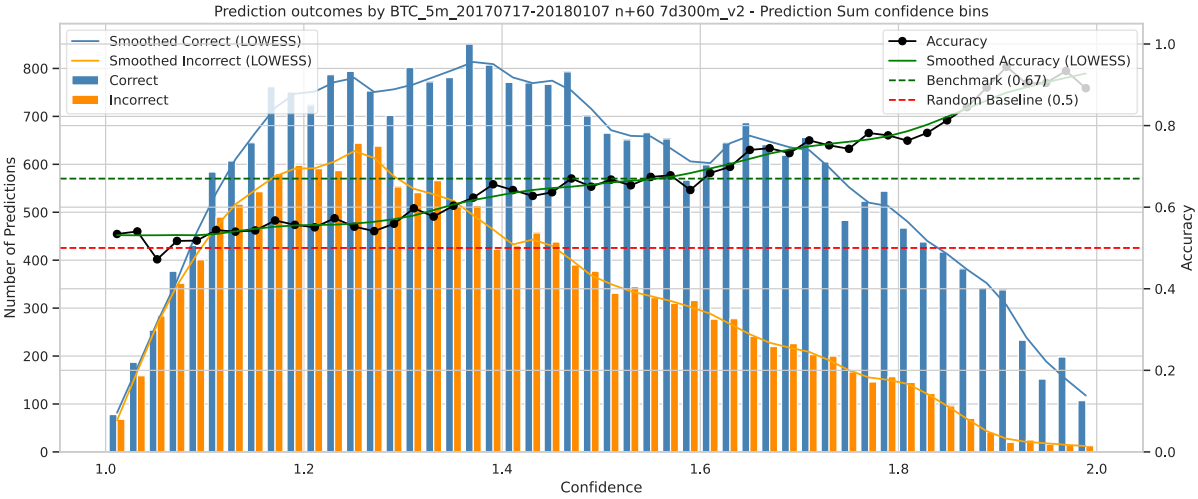


Figure 8: Number of correct versus incorrect predictions over increasing bins of confidence score for the combined predictions of RF_1 and RF_2, where both models predict the same price direction

Investigating different values for *T* in more detail, we construct a Table 3 which details how the threshold value impacts accuracy, but also the ratio of remaining data. The same data can also be seen in Figure 9. Worth noting is that while the absolute number of correct samples decreases with increasing values for *T*, the absolute number of incorrect samples decreases faster, thus giving a higher accuracy ratio in total.

Threshold	Correct %	Incorrect %	Inconclusive %	Accuracy %	Remaining data %
1.0	55.69	31.15	13.16	64.13	86.84
1.1	53.17	28.72	18.11	64.93	81.89
1.2	46.64	23.42	29.94	66.57	70.06
1.3	39.29	17.52	43.19	69.16	56.81
1.4	31.44	12.50	56.05	71.55	43.95
1.5	23.99	8.41	67.60	74.04	32.40
1.6	17.72	5.23	77.05	77.23	22.95
1.7	11.48	2.80	85.72	80.42	14.28
1.8	5.98	1.09	92.94	84.61	7.06
1.9	1.98	0.17	97.85	92.08	2.15
2.0	0.00	0.00	100.00	N/A	0.00

Table 3: Ratios of correct, incorrect, and inconclusive observations for different threshold values, and the corresponding accuracy ("Correct" versus "Incorrect") on the remaining data ("Correct" & "Incorrect" versus "Inconclusive")

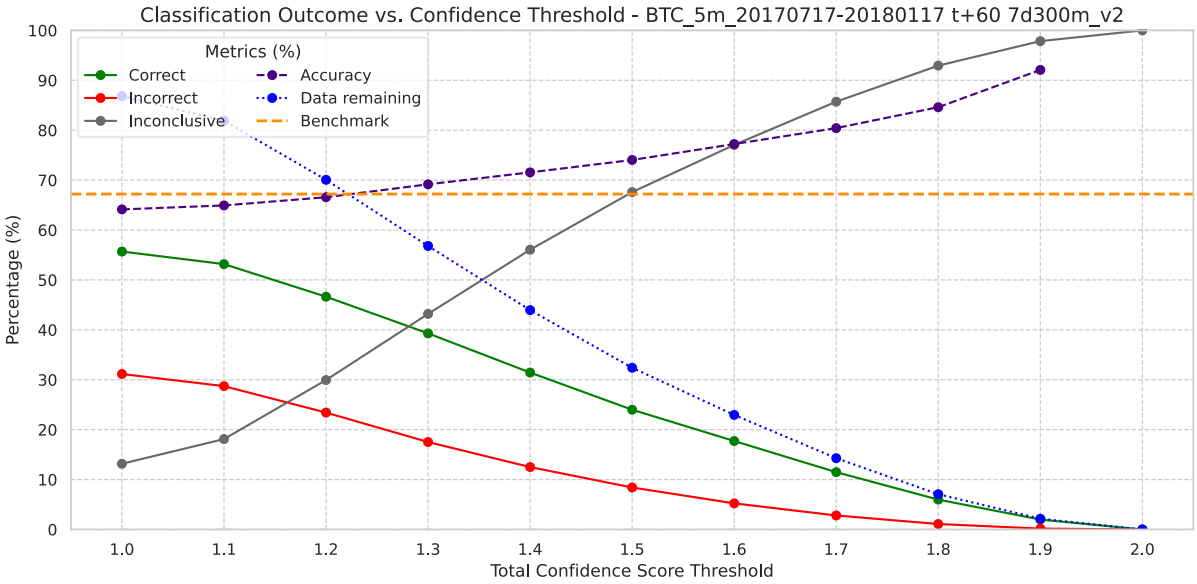


Figure 9: Changes in accuracy for increasing values of the threshold T, with the maximum accuracy of 67% achieved by Chen et al. included as a benchmark line for comparison

In summary, when applying the suggested method, we achieve results in terms of accuracy higher than those presented by Chen et al. for values of the threshold T>1.3. This comes with the trade-off of having 56.8% of the data samples discarded as "inconclusive" for T=1.3, and increasing for higher values of T. With more than 50% of the data remaining at T=1.3, corresponding to only being able to take trading decisions on roughly half of the samples, this is in our view a reasonable trade-off when working with high-frequency trading data such as 1- or 5-minute samples.

6.2. Multi-currency Results Over Two Years (January 2022 – December 2023)

For the period spanning from 2022-01-01 to 2023-12-31 we have applied our method to ten different cryptocurrencies, sampled at 5-minute intervals, equalling to 210,240 samples per currency pair. The sampled data for each pair then went through the same two-model frequent retraining method as applied earlier, with the training window length N = 2016 (7 days) and training window K = 60 (5 hours), giving a total of 3,465 iterations over the entire two-year period. The visual representation of the results of indexed price development for the ten sampled currencies, including an equal-weighted market index for comparison of overall market development between 2022 and 2023, is depicted in Figure 10. As seen in Table 4, our evaluated method gives very similar results also for other cryptocurrencies than BTC, and for the entire period of 2022 and 2023. This suggests that the method does provide a reliable and predictable increase in accuracy, also when applied to other currencies and timeframes and thus can be applied with similar results also outside the scope of this thesis. For further details on each examined pair in the dataset, see Appendix A.

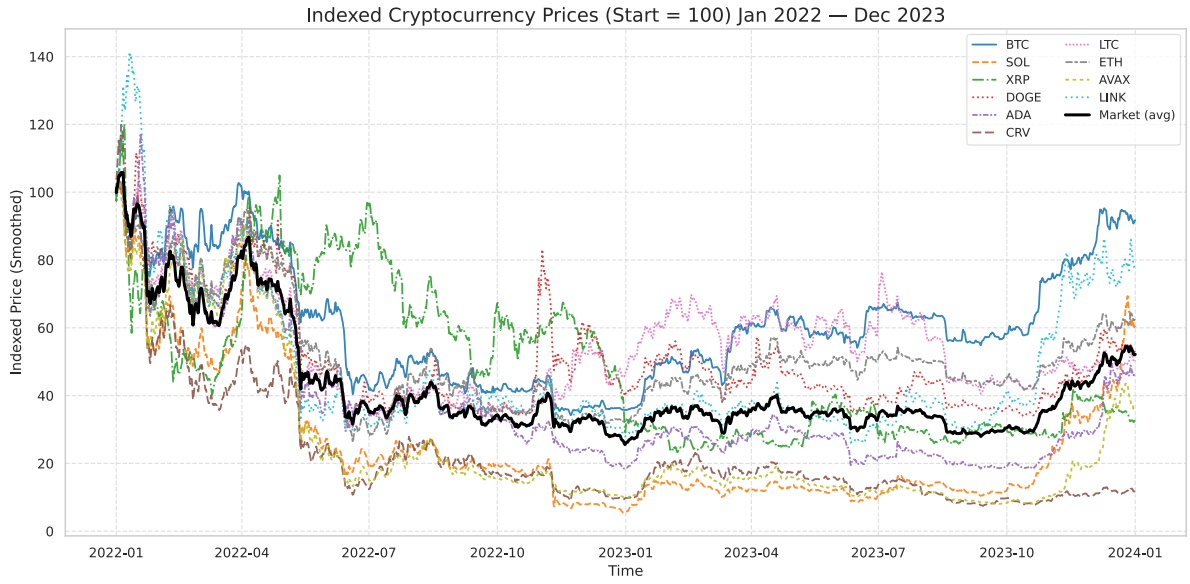


Figure 10: Indexed price development (start=100) for the sampled ten currencies, including an equal-weighted market index for comparison of overall market development 2022-2023

Accuracy %

Threshold	BTC	SOL	XRP	DOGE	ADA	CRV	LTC	ETH	AVAX	LINK	Market Average	Avg. Remaining Data %
1.0	65.8	65.5	65.5	65.7	65.6	66.1	66.0	65.3	65.4	65.5	65.6	87.9
1.1	66.7	66.4	66.5	66.7	66.5	67.1	67.0	66.2	66.3	66.5	66.6	82.4
1.2	68.8	68.4	68.7	68.8	68.6	69.2	69.3	68.3	68.3	68.7	68.7	70.0
1.3	71.4	71.2	71.2	71.3	71.3	71.9	72.0	70.7	70.8	71.2	71.3	56.5
1.4	74.0	74.4	74.2	74.1	74.3	75.0	75.0	73.5	73.7	74.2	74.2	43.6
1.5	77.1	77.4	77.2	77.4	77.9	78.3	78.1	76.6	76.9	77.2	77.4	32.1
1.6	80.7	80.4	81.1	81.2	81.8	81.9	81.2	80.8	79.6	81.1	80.8	22.0
1.7	84.0	83.7	84.6	85.4	85.6	85.7	85.0	84.4	84.0	84.6	84.6	13.5
1.8	88.0	88.1	87.7	90.2	88.3	89.7	88.9	88.3	88.1	87.7	88.5	6.7
1.9	92.0	93.5	93.7	94.5	93.3	92.3	95.1	93.7	91.1	93.7	93.0	1.8
2.0	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.0

Table 4: Accuracy % for the ten crypto currencies in the 2022-2023 sample for each threshold 1.0-2.0, including overall average accuracy % and average remaining data %

7. Discussion

While the area of research on ML-based methods for cryptocurrency price predictions has seen an increased interest in the scientific community over the past couple of years, few previous publications address the shorter timeframes such as 1- or 5-minute samples, and none of them the possibility of increasing the accuracy further using conformal classification and confidence score thresholding. In comparison with Chen et al., who thoroughly evaluated the current state of the art ML-methods and yielded a highest price prediction accuracy of around 67% on 5-minute samples, our research shows that by applying a confidence score threshold, the level of accuracy can reach 72% or even above 80%, while still keeping 44% respective 14% of the outcome prediction set. When examining the 2022 – 2023 dataset, the results look even more promising, yielding an average accuracy of 71% with nearly 57% remaining data, and 81% accuracy with 22% of the samples kept. This shows that traders who are willing to discard low-confidence-score predictions can expect a much higher accuracy in their model outputs at the samples where confidence scores are high, consequently leading to traders or trading bots that take more informed trading decisions. While the increased accuracy does come with a trade-off, this shortcoming can be mitigated by trading several parallel assets or by doing fewer trades, as a method to decrease risk and possibly achieve a better risk-reward ratio.

One of the conclusions drawn from the Efficient Market Hypothesis (EMH) is that even for weak-form-efficiency, future market prices should not be possible to predict using historical trading data alone. However, our results suggest that at certain times and in certain markets, such as the cryptocurrency market, which is still very much evolving, the Adaptive Market Hypothesis (Lo, 2004) may be a better theoretical fit given the results presented in this report. We can nevertheless expect that this, or the results from our method, will hold true forever, as evolving markets also mature over time. The fact that the results from the 2022 – 2023 dataset have not deteriorated compared to the 2017 – 2018 dataset is, however, promising and suggests that the markets for the ten investigated assets are still in a highly evolving and adapting state, with ample opportunities to trade on market inefficiencies. But to be practically applicable at the current date and beyond, we expect that the method will require continued and iterative improvements in order to evolve with the market.

7.1. Limitations

We have only explored the prediction of the price movement between the samples at positions *n* and *n*+60, corresponding to 60 minutes for 1-minute samples and 300 minutes (5 hours) for 5-minute sample data. There might be other prediction timeframes that are more beneficial, especially when decreasing the sample frequency further. If we want to maintain the goal of making intra-day trades that are less affected by macro-economic trends and signals, a significantly shorter prediction window should be considered, especially if higher thresholds are applied and fewer samples become tradable as a result. However, as shown by Nyquist H. (1928) and Shannon C.E. (1949), a signal cannot be reconstructed from its samples if the sampling rate is below twice the maximum frequency of the signal. This implies that variations faster than twice the sampling interval lie beyond the resolvable frequency range. While this concept originates from signal processing and does not translate directly to machine learning predictions, it nevertheless suggests that attempting to forecast only one sample ahead may be unreliable, since it is below the Nyquist-Shannon frequency where a signal cannot be reconstructed and is likely to contain sampling noise. As the input data to the model is sampled at a certain rate, Nyquist-Shannon also tells us that the input data does not reliably describe signals below

twice the sampling rate, and thus is unlikely to contain high-frequency signals that can be trusted for near-future-predictions. Therefore, it is more meaningful to predict at least two samples ahead, where the signal is less affected by under-sampling uncertainty.

A further limitation in the scope of this thesis is the number of coins examined, as they only represent a subset of the market, albeit chosen among the largest-volume coins for the studied time period. Additionally, only historical trading data is utilised for feature engineering and model training, where information about especially macro-economic factors could have contributed with additional predictive power to the model. Lastly, we chose to focus solely on the Random Forest algorithm for classification, a limitation made based on the high accuracy achieved in previous research in combination with the relative computational efficiency of the model.

7.2. Recommendations for Future Research

Our analysis focuses exclusively on predicting the direction of price movements, not their magnitude. Future research could address this limitation by also modelling the size of price changes, which would enable more advanced trading strategies and risk management. Further restricting positions to high risk-to-reward setups could significantly accelerate the growth of the account balance.

Preliminary simulated trading results based on the presented findings - which lie well beyond the scope of this thesis - indicate that for the exact same trading strategy and threshold value, the profitability still varies heavily between currencies and time periods. This suggests that further research is still required to refine both the method and the trading strategy before the findings in this report are implemented in algorithmic trading setups. To be able to trade with higher consistency and acceptable risk, we suggest that a detailed analysis of simulated trading results and the concurrent market conditions is performed to be able to further decide when a certain prediction should be considered for live trading.

7.3. Practical Applications

While algorithmic trading systems are typically designed to execute every signal generated by the underlying models, this approach may not always align with the practical needs of conventional day traders. In live trading environments, signal generation can be used as an additional tool for the traders. Instead of taking every model generated trade, the ones with high confidence score might be used by human traders. Traders can then integrate these signals with their own analysis, market context, and risk management preferences, choosing whether to complement their existing strategies with model-driven entry or exit points. This approach leverages the strengths of machine learning models in identifying predictive outputs with high confidence scores, while retaining the flexibility and judgment of experienced traders to account for factors not captured by quantitative models.

8. References

Al-Chalabi, H., Al-Douri, Y. K., & Lundberg, J. (2018). Time series forecasting using ARIMA model: A case study of mining face drilling rig [Master's thesis, Luleå University of Technology]. Luleå University of Technology Publications.

Anghel, D.-G. (2020). A reality check on trading rule performance in the cryptocurrency market: Machine learning vs. technical analysis. *Finance Research Letters, 36*, 101655.

Baur, D. G., Hong, K., & Lee, A. D. (2018). Bitcoin: Medium of exchange or speculative assets? *Journal of International Financial Markets, Institutions and Money, 54*, 177–189.

Biau, G. (2012). Analysis of a random forests model. *Journal of Machine Learning Research*, 13, 1063–1095.

Bitget. (2024, December 30). What was the price of Bitcoin when it first came out. *Bitget*. https://www.bitget.com/wiki/what-was-the-price-of-bitcoin-when-it-first-came-out

Bitnob Technologies. (2024, September 23). Bitnob | Transforming global financial pathways. *Bitnob*. https://bitnob.com/blog/usdt-meaning-use-cases

Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32.

Chen, Z., Li, C., & Sun, W. (2020). Bitcoin price prediction using machine learning: An approach to sample dimension engineering. *Journal of Computational and Applied Mathematics, 365*, 112395.

CoinMarketCap. (2024). Top cryptocurrency exchanges ranked by volume. *CoinMarketCap*. https://coinmarketcap.com/rankings/exchanges/

De Bondt, W. F., & Thaler, R. H. (1985). Does the stock market overreact? *Journal of Finance, 40*(3), 793–805.

Dudek, G., Fiszeder, P., Kobus, P., & Orzeszko, W. (2023). Forecasting cryptocurrencies volatility using statistical and machine learning methods: A comparative study. *Applied Soft Computing, 142*, 110409.

Fama, E. F. (1970). Efficient capital markets: A review of theory and empirical work. *Journal of Finance, 25*(2), 383–417.

Géron, A. (2019). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow* (2nd ed.). O'Reilly Media.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning* (2nd ed.). Springer.

Howarth, J. (2023, March 14). How many cryptocurrencies are there in 2024? *Exploding Topics*. https://explodingtopics.com/blog/number-of-cryptocurrencies

Jang, H., & Lee, J. (2017). An empirical study on modeling and prediction of Bitcoin prices with Bayesian neural networks based on blockchain information. *IEEE Access, 6*, 5427–5437.

Jaquart, P., Köpke, S., & Weinhardt, C. (2022). Machine learning for cryptocurrency market prediction and trading. *The Journal of Finance and Data Science, 8*(4), 292–307.

Jensen, M. C. (1978). Some anomalous evidence regarding market efficiency. *Journal of Financial Economics, 6*(2–3), 95–101.

Johansson, U., Boström, H., & Löfström, T. (2013). Conformal prediction using decision trees. In *2013 IEEE 13th International Conference on Data Mining* (pp. 330–339). IEEE.

Kaufman, P. J. (2013). *Trading systems and methods* (5th ed., pp. 847, 851, 1065). John Wiley & Sons.

Lahmiri, S., & Bekiros, S. (2019). Cryptocurrency forecasting with deep learning chaotic neural networks. *Chaos, Solitons & Fractals, 118*, 35–40.

Lo, A. W. (2004). The adaptive markets hypothesis: Market efficiency from an evolutionary perspective. *Journal of Portfolio Management*, 30(5), 15–29.

McNally, S., Roche, J., & Caton, S. (2018). Predicting the price of bitcoin using machine learning. In *2018 26th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP)* (pp. 339–343). IEEE.

Murphy, J. J. (1999). *Technical analysis of the financial markets: A comprehensive guide to trading methods and applications* (pp. 162, 196, 199, 240, 253). New York Institute of Finance.

Nakamoto, S. (2008). *Bitcoin: A peer-to-peer electronic cash system* [White paper]. https://bitcoin.org/bitcoin.pdf

Nyquist, H. (1928). Certain topics in telegraph transmission theory. *Transactions of the American Institute of Electrical Engineers, 47*(2), 617–644.

Orte, F., Mira, J., Sánchez, M. J., & Solana, P. (2023). A random forest-based model for crypto asset forecasts in futures markets with out-of-sample prediction. *Research in International Business and Finance, 64*, 101824.

Powers, D. M. W. (2020). Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. *International Journal of Machine Learning Technology, 2*(1), 37–63.

Sathyanarayanan, S. (2024). Confusion matrix-based performance evaluation metrics. *African Journal of Biomedical Research, 27*(4s), 4023–4031.

Sebastião, H., & Godinho, P. (2021). Forecasting and trading cryptocurrencies with machine learning under changing market conditions. *Research in International Business and Finance, 56*, 101383.

Shannon, C. E. (1949). Communication in the presence of noise. *Proceedings of the IRE, 37*(1), 10–21.

Smith, A. (1776). *An inquiry into the nature and causes of the wealth of nations*. W. Strahan and T. Cadell.

Soloviev, V., Bielinskyi, A., Serdyuk, O., Solovieva, V., & Semerikov, S. (2020). Lyapunov exponents as indicators of the stock market crashes. In *Proceedings of the 8th International Conference on Monitoring, Modeling & Management of Emergent Economy (M3E2 2020), CEUR Workshop Proceedings* (Vol. 2732, pp. 455–464).

Stoltzfus, J. C. (2011). Logistic regression: A brief primer. *Academic Emergency Medicine, 18*(10), 1099–1104.

TradingView. (2025). Balance of power (BOP). *TradingView*. https://www.tradingview.com/support/solutions/43000589100-balance-of-power-bop/

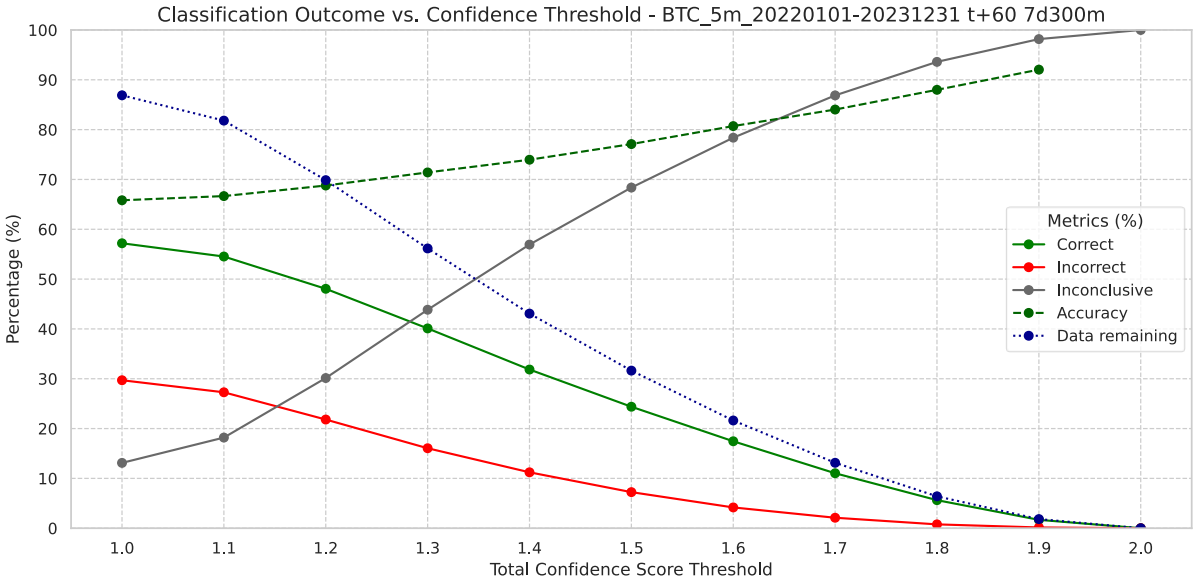
Urquhart, A. (2016). The inefficiency of Bitcoin. *Economics Letters, 148*, 80–82.

Young, L.-S. (2013). Mathematical theory of Lyapunov exponents. *Journal of Physics A: Mathematical and Theoretical, 46*(25), 254001.

Appendix A: Detailed Accuracy Data for Each of the Ten Cryptocurrency Pairs in the January 2022 – December 2023 Dataset

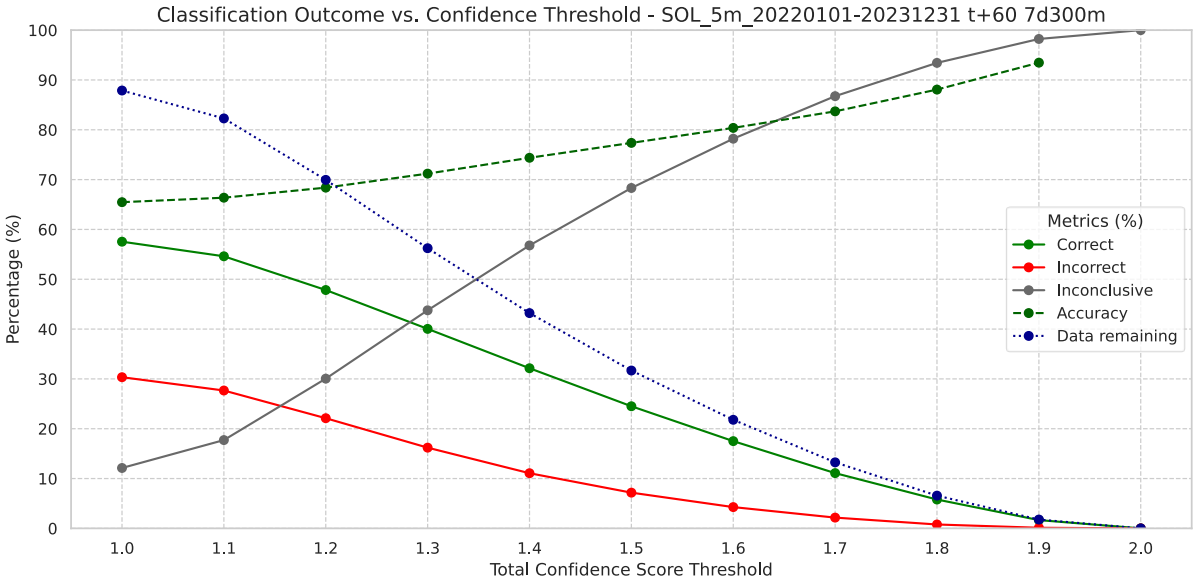
A.1 Bitcoin (BTC)

Threshold	Correct %	Incorrect %	Inconclusive %	Accuracy %	Data remaining %
1.0	57.19	29.70	13.11	65.82	86.89
1.1	54.53	27.28	18.19	66.66	81.81
1.2	48.05	21.81	30.14	68.78	69.86
1.3	40.10	16.05	43.85	71.41	56.15
1.4	31.85	11.22	56.93	73.96	43.07
1.5	24.38	7.24	68.37	77.10	31.63
1.6	17.45	4.17	78.38	80.71	21.62
1.7	11.03	2.10	86.87	84.03	13.13
1.8	5.63	0.77	93.60	87.99	6.40
1.9	1.68	0.15	98.17	92.04	1.83
2.0	0.00	0.00	100.00	N/A	0.00



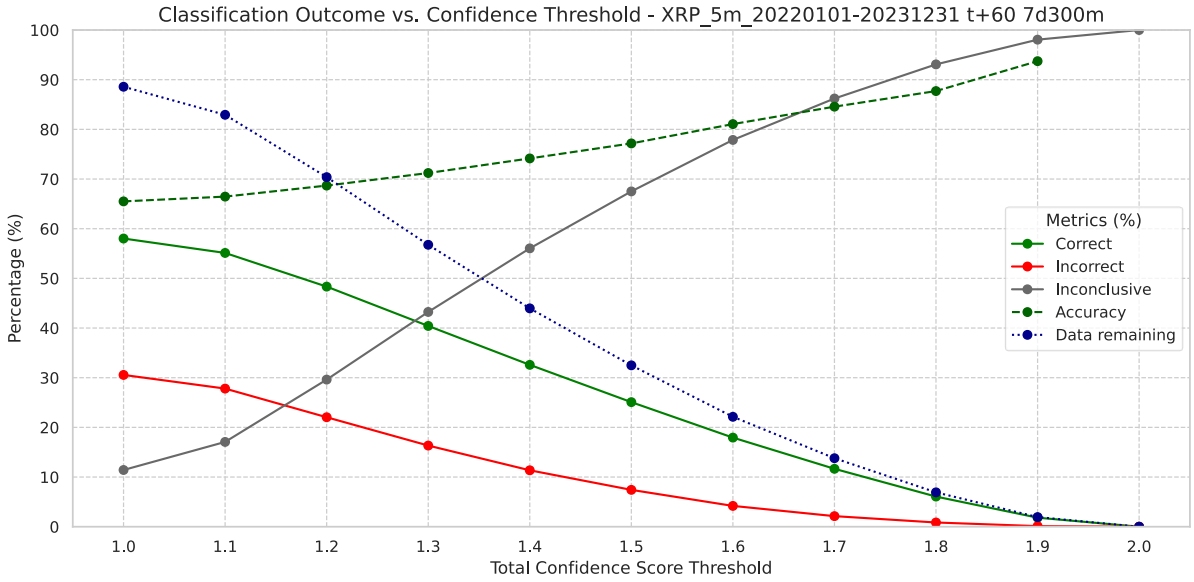
A.2 Solana (SOL)

Threshold	Correct %	Incorrect %	Inconclusive %	Accuracy %	Data remaining %
1.0	57.54	30.34	12.12	65.47	87.88
1.1	54.61	27.67	17.72	66.37	82.28
1.2	47.84	22.11	30.05	68.39	69.95
1.3	40.04	16.20	43.77	71.20	56.23
1.4	32.14	11.07	56.79	74.38	43.21
1.5	24.51	7.17	68.32	77.37	31.68
1.6	17.51	4.27	78.21	80.38	21.79
1.7	11.09	2.16	86.75	83.70	13.25
1.8	5.79	0.78	93.43	88.07	6.57
1.9	1.66	0.12	98.23	93.47	1.77
2.0	0.00	0.00	100.00	N/A	0.00



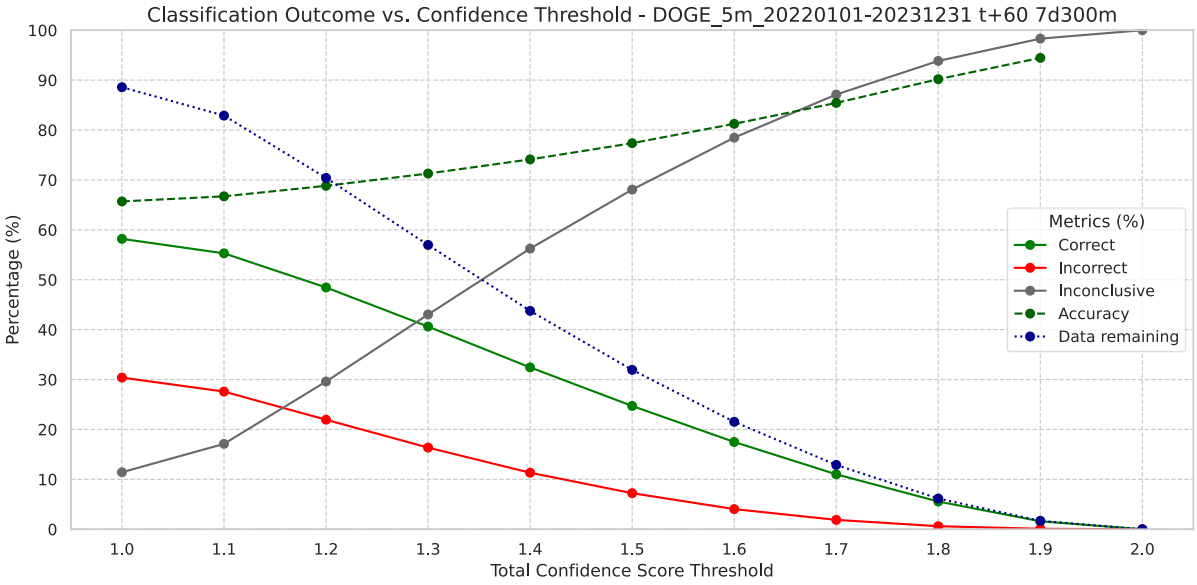
A.3 Ripple (XRP)

Threshold	Correct %	Incorrect %	Inconclusive %	Accuracy %	Data remaining %
1.0	58.02	30.56	11.42	65.50	88.58
1.1	55.12	27.81	17.07	66.46	82.93
1.2	48.34	22.04	29.62	68.68	70.38
1.3	40.42	16.34	43.24	71.21	56.76
1.4	32.59	11.36	56.04	74.15	43.96
1.5	25.08	7.42	67.51	77.17	32.49
1.6	17.94	4.19	77.87	81.06	22.13
1.7	11.66	2.13	86.21	84.57	13.79
1.8	6.07	0.85	93.08	87.71	6.92
1.9	1.82	0.12	98.06	93.73	1.94
2.0	0.00	0.00	100.00	N/A	0.00



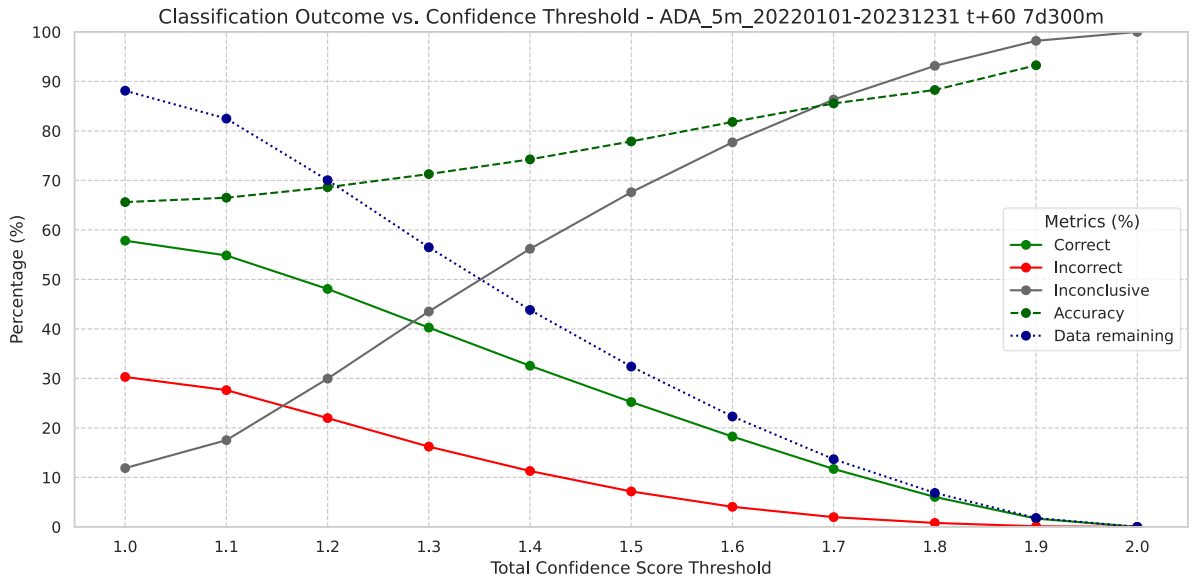
A.4 DogeCoin (DOGE)

Threshold	Correct %	Incorrect %	Inconclusive %	Accuracy %	Data remaining %
1.0	58.19	30.39	11.42	65.69	88.58
1.1	55.29	27.59	17.11	66.71	82.89
1.2	48.45	21.95	29.60	68.82	70.40
1.3	40.61	16.36	43.03	71.28	56.97
1.4	32.43	11.33	56.24	74.11	43.76
1.5	24.71	7.23	68.06	77.36	31.94
1.6	17.49	4.04	78.47	81.24	21.53
1.7	11.01	1.88	87.11	85.43	12.89
1.8	5.56	0.61	93.84	90.18	6.16
1.9	1.60	0.09	98.31	94.46	1.69
2.0	0.00	0.00	100.00	N/A	0.00



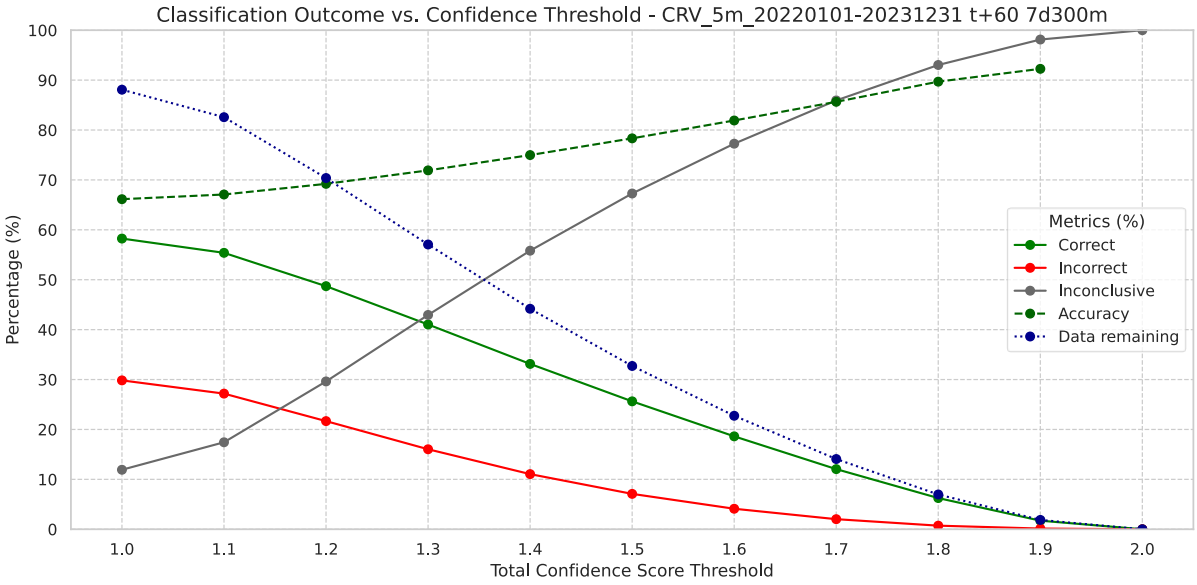
A.5 Cardano (ADA)

Threshold	Correct %	Incorrect %	Inconclusive %	Accuracy %	Data remaining %
1.0	57.82	30.29	11.88	65.62	88.12
1.1	54.84	27.63	17.52	66.50	82.48
1.2	48.07	21.98	29.95	68.62	70.05
1.3	40.26	16.22	43.52	71.28	56.48
1.4	32.55	11.29	56.17	74.25	43.83
1.5	25.23	7.16	67.61	77.88	32.39
1.6	18.25	4.06	77.69	81.81	22.31
1.7	11.70	1.97	86.33	85.55	13.67
1.8	6.05	0.80	93.14	88.27	6.86
1.9	1.68	0.12	98.20	93.26	1.80
2.0	0.00	0.00	100.00	N/A	0.00



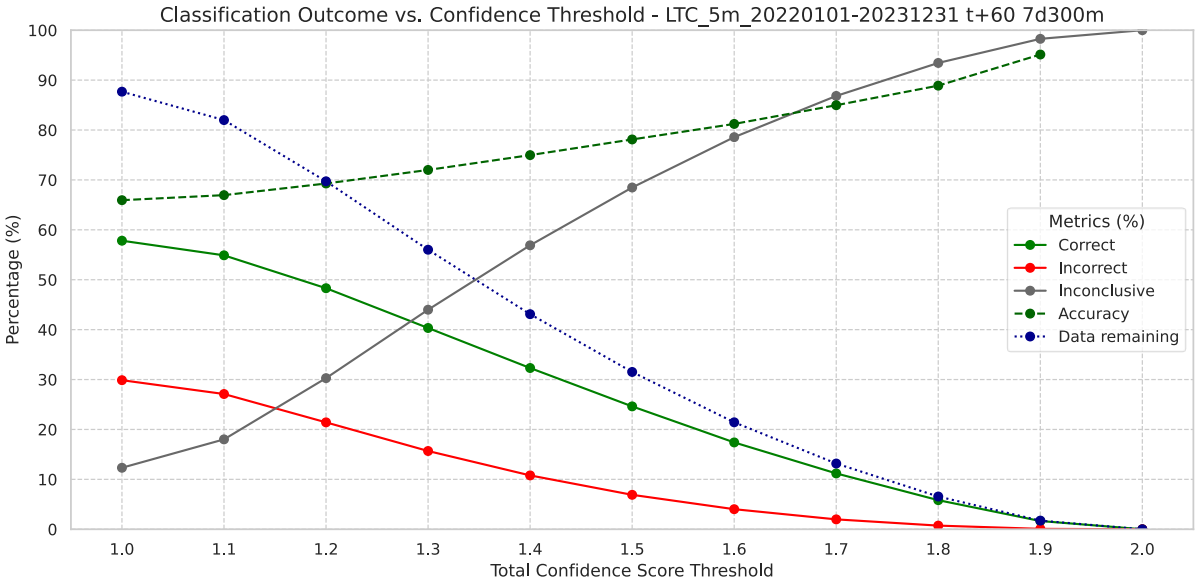
A.6 Curve DAO Token (CRV)

Threshold	Correct %	Incorrect %	Inconclusive %	Accuracy %	Data remaining %
1.0	58.25	29.83	11.92	66.14	88.08
1.1	55.39	27.18	17.43	67.08	82.57
1.2	48.71	21.67	29.62	69.21	70.38
1.3	41.03	16.03	42.94	71.91	57.06
1.4	33.13	11.05	55.82	74.98	44.18
1.5	25.63	7.09	67.28	78.33	32.72
1.6	18.62	4.11	77.26	81.91	22.74
1.7	12.05	2.02	85.93	85.65	14.07
1.8	6.25	0.72	93.03	89.69	6.97
1.9	1.72	0.14	98.13	92.26	1.87
2.0	0.00	0.00	100.00	N/A	0.00



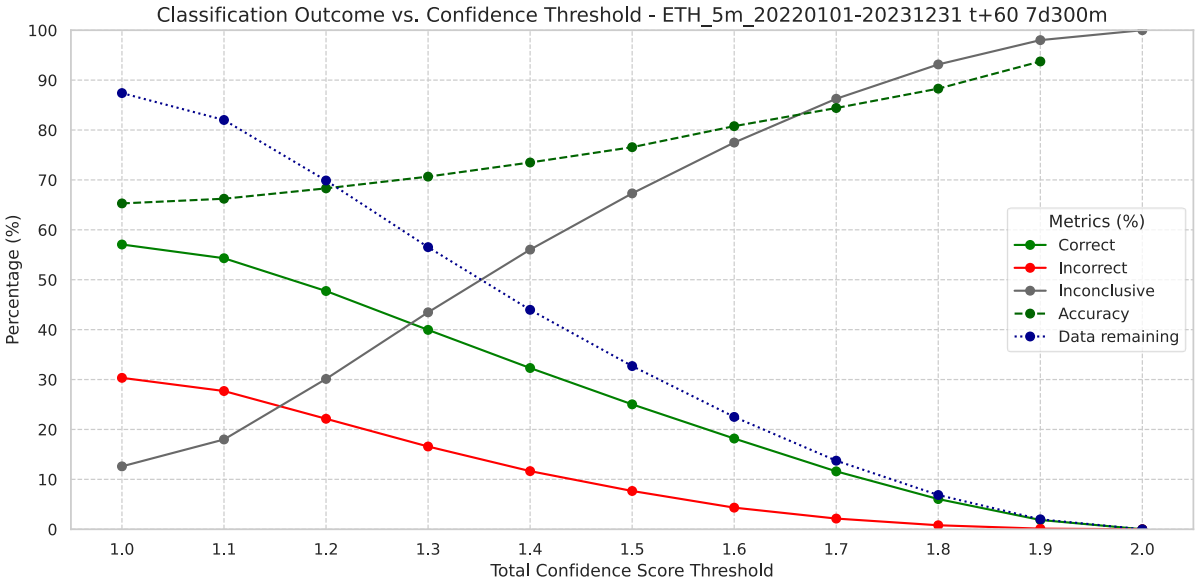
A.7 Litecoin (LTC)

Threshold	Correct %	Incorrect %	Inconclusive %	Accuracy %	Data remaining %
1.0	57.82	29.87	12.32	65.94	87.68
1.1	54.89	27.10	18.01	66.95	81.99
1.2	48.30	21.42	30.28	69.28	69.72
1.3	40.33	15.68	43.99	72.00	56.01
1.4	32.31	10.79	56.90	74.97	43.10
1.5	24.62	6.90	68.48	78.11	31.52
1.6	17.40	4.02	78.57	81.22	21.43
1.7	11.19	1.98	86.83	84.97	13.17
1.8	5.83	0.73	93.44	88.89	6.56
1.9	1.65	0.08	98.27	95.13	1.73
2.0	0.00	0.00	100.00	N/A	0.00



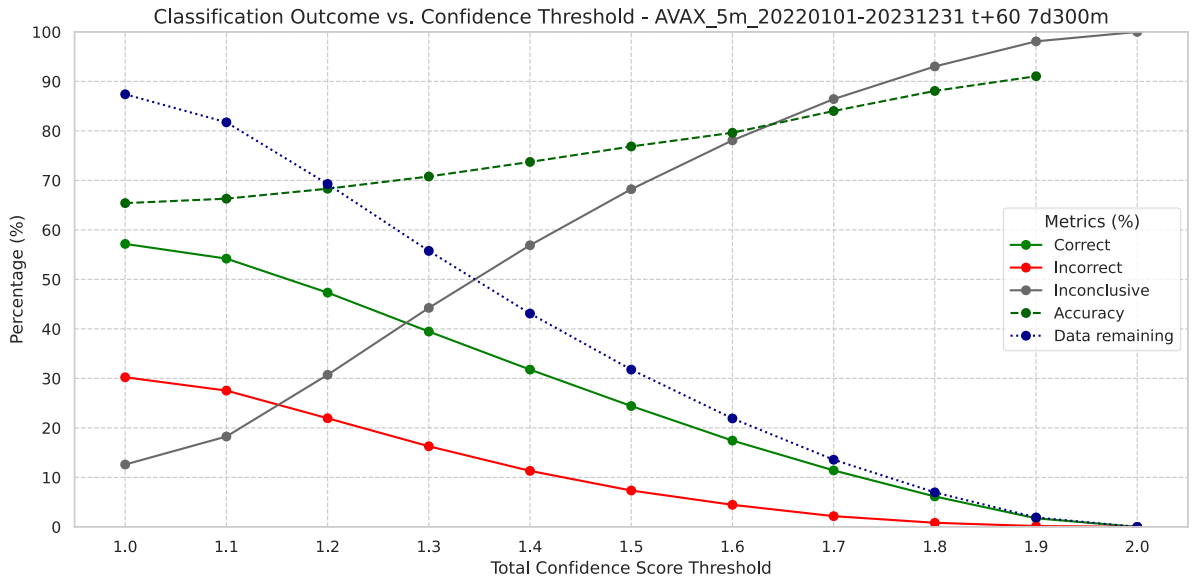
A.8 Ethereum (ETH)

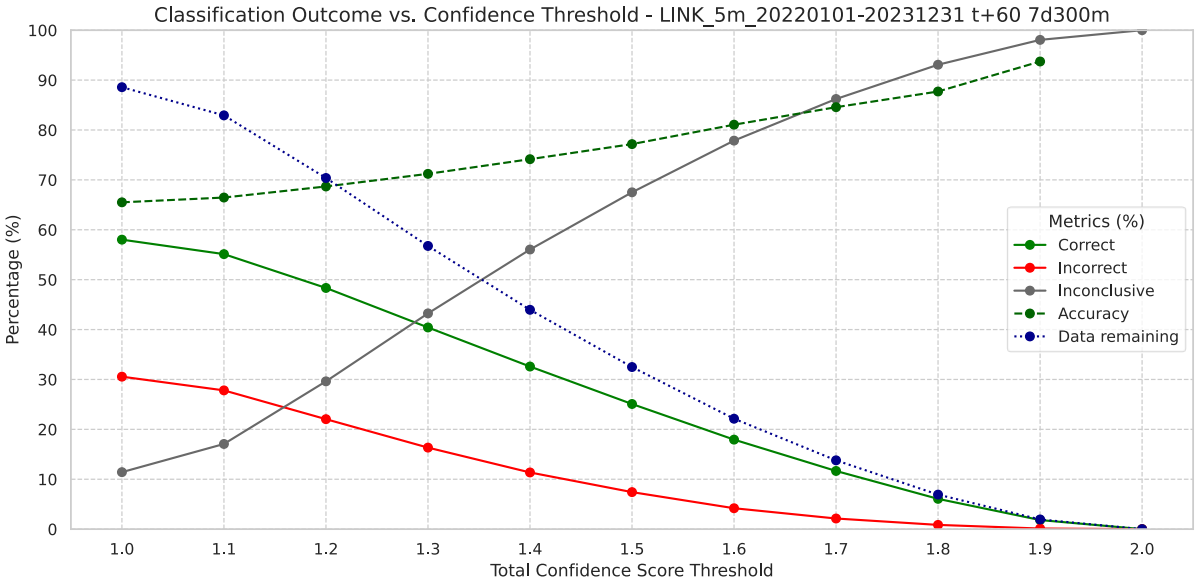
Threshold	Correct %	Incorrect %	Inconclusive %	Accuracy %	Data remaining %
1.0	57.06	30.34	12.60	65.29	87.40
1.1	54.31	27.70	17.99	66.23	82.01
1.2	47.74	22.14	30.12	68.31	69.88
1.3	39.95	16.58	43.47	70.66	56.53
1.4	32.31	11.65	56.03	73.49	43.97
1.5	25.03	7.67	67.30	76.56	32.70
1.6	18.18	4.33	77.49	80.77	22.51
1.7	11.60	2.14	86.26	84.41	13.74
1.8	6.05	0.80	93.14	88.29	6.86
1.9	1.86	0.12	98.01	93.74	1.99
2.0	0.00	0.00	100.00	N/A	0.00



A.9 Avalanche (AVAX)

Threshold	Correct %	Incorrect %	Inconclusive %	Accuracy %	Data remaining %
1.0	57.17	30.23	12.60	65.41	87.40
1.1	54.20	27.54	18.26	66.31	81.74
1.2	47.33	21.95	30.72	68.32	69.28
1.3	39.47	16.29	44.24	70.79	55.76
1.4	31.78	11.32	56.90	73.73	43.10
1.5	24.42	7.35	68.22	76.86	31.78
1.6	17.44	4.46	78.09	79.63	21.91
1.7	11.41	2.17	86.43	84.02	13.57
1.8	6.15	0.83	93.02	88.08	6.98
1.9	1.74	0.17	98.09	91.05	1.91
2.0	0.00	0.00	100.00	N/A	0.00



A.10 Chainlink (LINK)

Threshold	Correct %	Incorrect %	Inconclusive %	Accuracy %	Data remaining %
1.0	58.02	30.56	11.42	65.50	88.58
1.1	55.12	27.81	17.07	66.46	82.93
1.2	48.34	22.04	29.62	68.68	70.38
1.3	40.42	16.34	43.24	71.21	56.76
1.4	32.59	11.36	56.04	74.15	43.96
1.5	25.08	7.42	67.51	77.17	32.49
1.6	17.94	4.19	77.87	81.06	22.13
1.7	11.66	2.13	86.21	84.57	13.79
1.8	6.07	0.85	93.08	87.71	6.92
1.9	1.82	0.12	98.06	93.73	1.94
2.0	0.00	0.00	100.00	N/A	0.00