EXPLORING CRYPTOCURRENCY TIME SERIES DATA USING TOPOLOGICAL DATA ANALYSIS

DION CHAN TECK ONN

This project is submitted in partial fulfillment of
the requirements for the degree of Bachelor of Computer Science with Honors
(Software Engineering)

Faculty of Computer Science and Information Technology
UNIVERSITI MALAYSIA SARAWAK
2025

MENEROKA DATA SIRI MASA WANG KRIPTO MENGGUNAKAN ANALISIS DATA TOPOLOGI

DION CHAN TECK ONN

Projek ini merupakan salah satu keperluan untuk
Ijazah Sarjana Muda Sains Komputer dengan Kepujian
(Kejuruteraan Perisian)

Fakulti Sains Komputer dan Teknologi Maklumat
UNIVERSITI MALAYSIA SARAWAK
2025

UNIVERSITI MALAYSIA SARAWAK

THESIS STATUS ENDORSEMENT FORM

TITLE **Exploring Cryptocurrency Time Series Data Using Topological Data Analysis**

ACADEMIC SESSION: 2024 / 2025

DION CHAN TECK ONN

(CAPITAL LETTERS)

hereby agree that this Thesis* shall be kept at the Centre for Academic Information Services, Universiti Malaysia Sarawak, subject to the following terms and conditions:

1. The Thesis is solely owned by Universiti Malaysia Sarawak
2. The Centre for Academic Information Services is given full rights to produce copies for educational purposes only
3. The Centre for Academic Information Services is given full rights to do digitization in order to develop local content database
4. The Centre for Academic Information Services is given full rights to produce copies of this Thesis as part of its exchange item program between Higher Learning Institutions [or for the purpose of interlibrary loan between HLI]
5. ** Please tick (√)

☐	CONFIDENTIAL	(Contains classified information bounded by the OFFICIAL SECRETS ACT 1972)
☐	RESTRICTED	(Contains restricted information as dictated by the body or organization where the research was conducted)
✓	UNRESTRICTED	

Validated by

Dion

(AUTHOR'S SIGNATURE)

(SUPERVISOR'S SIGNATURE)

DR. PHANG PIAU
Senior Lecturer
Computational Science Programme
Faculty of Computer Science and Information Technology
Universiti Malaysia Sarawak

Permanent Address

169, 2A1C Muara Tabuan,
Marua Tabuan, 93350
Kuching, Sarawak

Date: 22/7/2025

Date: 23/7/2025

Note * Thesis refers to PhD, Master, and Bachelor Degree
** For Confidential or Restricted materials, please attach relevant documents from relevant organizations / authorities

ACKNOWLEDGEMENTS

I would like to thank my supervisor, Dr. Phang Piau, for his continuous guidance and patience throughout this project. His clear advice and helpful feedback gave me the confidence to move forward at each stage and complete my work step by step. I would also like to thank Zielak, the author of the Bitcoin price dataset on Kaggle, which was used in this study and made the analysis possible. I am also grateful to my friends who shared useful ideas and were willing to help when I faced difficulties. Since my English is bad, I also want to thank translation tools and AI assistants, which helped my better understanding and learning during this project. In addition, I would like to thank the Bilibili video creators whose tutorials helped me gain a clearer understanding of the topic in a more visual and easy-to-follow way.

ABSTRACT

Cryptocurrencies, especially Bitcoin, are widely known for their significant volatility and unpredictable price movements, complicating efforts to analyze and predict their behaviour. Traditional time-series models often fail to capture the complex nonlinear dynamics and irregularities in Bitcoin's pricing data, leading to an incomplete understanding of its market trends. This study introduces topological data analysis as a new approach to address these issues. The implementation involved transforming Bitcoin's daily price data into point clouds using Takens embedding and analyzing them with persistent homology. Persistence diagrams were generated for four market phases, which are flat, stable trend, bullish and volatile. The results showed that stable periods produced simpler topological patterns, while volatile periods produced more persistent loops, reflecting complex and repeating behaviours. These findings highlight how topological data analysis reveals structural differences across market conditions. In conclusion, topological data analysis offers a powerful alternative to traditional analysis by uncovering hidden patterns in time series data, providing deeper insights into cryptocurrency price dynamics.

ABSTRAK

Mata wang kripto, terutamanya Bitcoin, terkenal dengan tahap volatiliti yang tinggi dan pergerakan harga yang tidak menentu, yang menyukarkan usaha untuk menganalisis dan meramalkan tingkah lakunya. Model siri masa tradisional sering gagal menangkap dinamik bukan linear yang kompleks serta ketidakteraturan dalam data harga Bitcoin, yang membawa kepada pemahaman yang tidak lengkap terhadap trend pasaran. Kajian ini memperkenalkan analisis data topologi sebagai pendekatan baharu untuk menangani isu-isu ini. Pelaksanaan kajian melibatkan penukaran data harga harian Bitcoin kepada awan titik menggunakan embedding Takens dan menganalisisnya melalui homologi berterusan. Rajah ketekalan telah dihasilkan bagi empat fasa pasaran: mendatar, trend stabil, bullish dan volatil. Hasil kajian menunjukkan bahawa tempoh stabil menghasilkan corak topologi yang lebih ringkas, manakala tempoh volatil menunjukkan lebih banyak gelung berterusan yang mencerminkan tingkah laku kompleks dan berulang. Penemuan ini menunjukkan bahawa analisis data topologi dapat mendedahkan perbezaan struktur antara pelbagai keadaan pasaran. Kesimpulannya, pendekatan ini menawarkan alternatif yang berkuasa kepada kaedah tradisional dengan mendedahkan corak tersembunyi dalam data siri masa, sekali gus memberikan pemahaman yang lebih mendalam terhadap dinamik harga mata wang kripto.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF EQUATIONS

Chapter 1

Introduction

1.1 Background

Cryptocurrency is a digital currency that operates on a decentralized network using blockchain technology. Unlike traditional currencies and financial assets, cryptocurrencies are not controlled by any government or central bank. Instead, their prices are influenced by market speculation, investor behaviour, global economic events and technological changes (Catania & Grassi, 2017). Bitcoin, the first and most recognizable cryptocurrency (Manjula et al., 2022), has grown into a major financial asset, but its price remains extremely volatile and unpredictable. The lack of regulation and frequent price fluctuations make Bitcoin a particularly challenging asset to analyze. As shown in Figure 1 (Sabry et al, 2020), Bitcoin had the largest market share in July 2020.

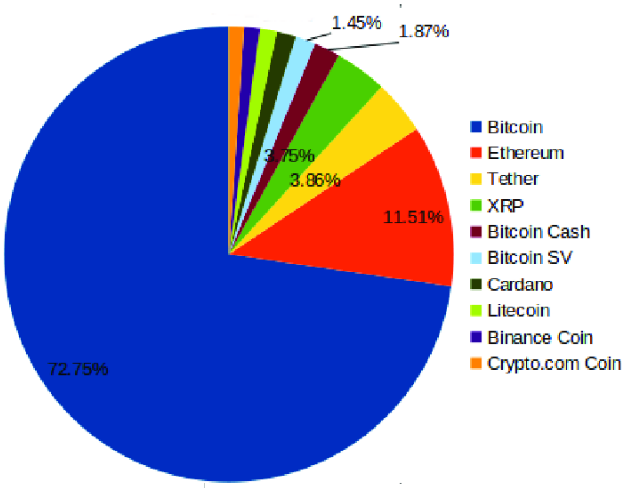


Figure 1.1 Market capitalization of cryptocurrencies in July 2020

Traditional analytical techniques present significant limitations when studying Bitcoin's price movements. Common financial tools, such as statistical models and time series analysis, can identify general trends and fluctuations over time. However, these methods often fail to capture the more complex and underlying patterns that influence Bitcoin's price (Naimy et al., 2021). They may overlook important cycles, trends, or anomalies, leaving gaps in our understanding of Bitcoin's market behaviour.

As a result, accurately predicting Bitcoin’s price movements and reacting to market changes becomes increasingly difficult.

Topological data analysis (TDA) is an emerging approach that focuses on understanding the structure of complex datasets. TDA identifies recurring patterns, clusters, and cycles. This deeper analysis provides a better understanding of Bitcoin's price behaviour and its volatility. While TDA has been used in fields like image processing and biology (Chazal & Michel, 2021), its application in financial analysis, particularly for cryptocurrencies, is still developing.

1.2 Problem Statement

The cryptocurrency market, especially Bitcoin, is highly volatile, making it hard to predict its price reliably. Traditional methods like time series models can identify basic trends, but they cannot capture the complex, unpredictable changes that affect Bitcoin's price. These methods are also not able to identify important topological patterns in the data, such as cycles, clusters and repeating trends, that could provide a deeper understanding of market behaviour. Because of these limitations, existing prediction models often fail to identify key signals that could help forecast price changes. There is a clear need for better methods that can find hidden patterns in cryptocurrency data.

1.3 Objectives

This study aims to use topological data analysis approach to analyze Bitcoin's price data and uncover valuable insights into its market behaviour. The main objectives include:

i. Find hidden patterns in Bitcoin price data
ii. Evaluate Bitcoin's long-term price trends and stability
iii. Analyze Bitcoin’s anomalies and volatility
iv. Examine Bitcoin’s market cycle dynamics

1.4 Scope

Since Bitcoin is the most researched and traded cryptocurrency, this study will focus solely on its price data. Bitcoin serves as a strong case study due to its long history and frequent market activity, which offers enough data to uncover meaningful structural patterns. The analysis applies topological data analysis to daily closing prices and explores different market conditions, which include the stable phases and periods of high volatility.

1.5 Brief Methodology

Knowledge Discovery in Databases (KDD) is the process of extracting useful and meaningful knowledge from large datasets. It involves a series of steps that systematically transform raw data into actionable insights. In this study, KDD is a framework for analyzing Bitcoin price data to discover hidden patterns, trends, and market dynamics.

1.5.1 Data Collection

Bitcoin's historical price data from Kaggle provides a comprehensive view of long-term price trends with a focus on daily closing prices.

1.5.2 Data Preprocessing

The collected data were cleaned by dealing with missing values, outliers and inconsistencies. The cleaned data sets were standardized to ensure uniformity of data in preparation for further analysis.

1.5.3 Data Transformation

One-dimensional time series data is converted to a two-dimensional representation using the Takens embedding theorem. This step allows for the reconstruction of the state space of Bitcoin's price movements, allowing for the analysis of local trends and fluctuations.

1.5.4 Data Mining

Topological Data Analysis (TDA) is the use of persistent homology to generate persistence diagrams. These diagrams capture the birth and death of topological features, providing insight into Bitcoin's price structure, trends, and volatility (Alexander, 2024).

1.5.6 Evaluation

Persistence diagrams are analyzed to assess long-term price trends, stability, anomalies and market cycles. This step focuses on identifying persistent features that reflect stable trends and short-term features that indicate volatility or anomalies.

1.6 Significant of Project

Persistence diagrams will be used to show the patterns and topological features found in Bitcoin’s price data. These visualizations will make it easier to understand the analysis results and provide a clearer view of Bitcoin's price behaviour. By linking these features to real-world market events, such as price crashes or breaking news, this study can provide insight into the relationship between topology and market behaviour. These findings can improve our ability to predict future price changes and enhance decision-making in cryptocurrency trading and analysis.

1.7 Project Schedule

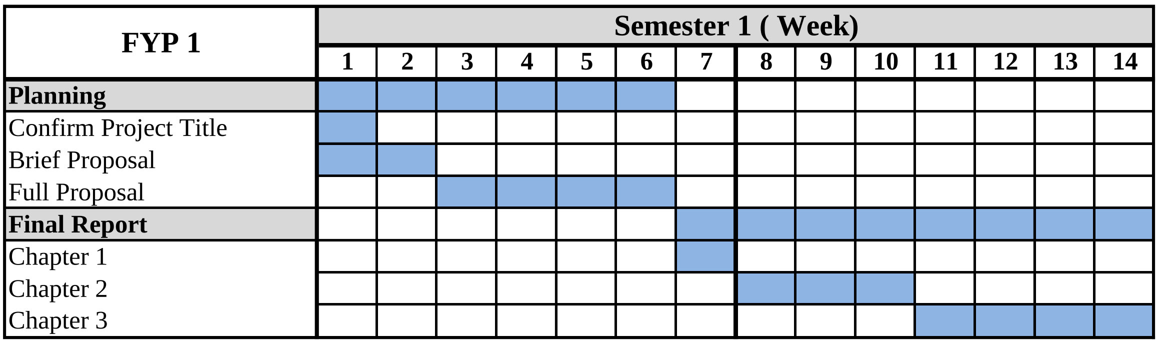


Figure 1.2 Gantt chart for Final Year Project 1

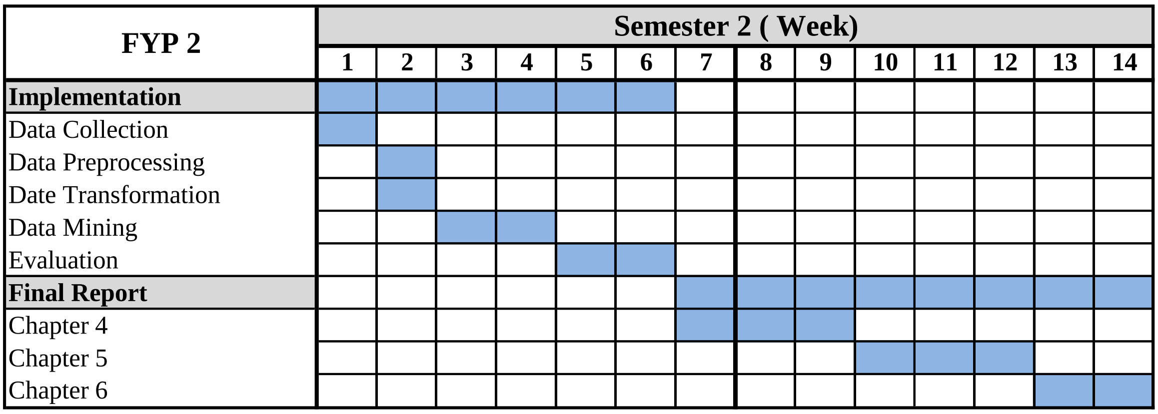


Figure 1.3 Gantt chart for Final Year Project 2

1.8 Expected Outcome

The project will use persistence diagrams to identify recurring patterns in Bitcoin price data, such as cycles or clusters and revealing hidden structures. It will also provide insights into Bitcoin's price volatility, especially during periods of high fluctuation, helping to understand the factors behind large price movements. Simple visualizations will be created to show how the identified patterns relate to real-world market events, helping investors and analysts make better financial decisions.

1.9 Report Outline

This report consists of six chapters. Each chapter outlines a specific part of the project, starting from the introduction, literature review and methodology, followed by implementation, result and discussion, and ending with the conclusion and future work.

1.9.1 Introduction

Chapter 1 introduces the project and summarizes its main aspects. It includes a problem statement, objectives, project scope, project significance, methodology, project schedule, expected results and a report outline.

1.9.2 Literature Review

Chapter 2 reviews prior studies on Bitcoin price prediction, focusing on statistical models like ARIMA and machine learning techniques such as Random Forest and Neural Networks. It also introduces tools for analyzing persistence diagrams, which are used in subsequent chapters for data analysis.

1.9.3 Methodology

Chapter 3 describes the steps taken to complete the project. These include collecting Bitcoin price data, cleaning and preparing the data for analysis, and applying TDA techniques such as persistent homology to study the data.

1.9.4 Implementation

Chapter 4 details the implementation of the methodology. It explains how to set up the analysis environment, the process of finding patterns in Bitcoin price data using TDA and the generation of persistence diagrams.

1.9.5 Result and Discussion

Chapter 5 presents the results of the analysis. It explains the topological features found, such as cycles or clusters and how they relate to Bitcoin price stability and market behaviour. This chapter also uses simple visualizations to make the results easier to understand and discusses the implications of these findings for understanding Bitcoin price patterns.

1.9.6 Conclusion and Future Works

Chapter 6 summarizes the findings of this project and discusses potential directions for future work.

Chapter 2

Literature Review

2.1 Overview

This chapter discusses the various methods and approaches used in Bitcoin price forecasting, highlighting work done by researchers over the past few years. The research papers reviewed employ a range of techniques, from time series forecasting using ARIMA and volatility estimation using GARCH, to machine learning models such as random forests, SVR, neural networks, and logistic regression. This chapter provides a comparative analysis of these studies to provide insight into the strengths and limitations of each approach.

2.2 Time Series Data and Analysis

This section provides an overview of time series data and explains how such data can be analysed to reveal trends, seasonal patterns and unusual behaviour. The section also outlines common analytical methods used to study price changes over time.

2.2.1 Time Series Data

Time series data refers to data collected or recorded at consecutive intervals, usually regular or consistent periods, such as daily, monthly or yearly. It captures variable changes over time so that trends, patterns and cyclical behaviour can be observed. Time series data is essential for forecasting, anomaly detection, and understanding the dynamic relationships between variables in different fields such as finance, economics, and environmental sciences (Hyndman, 2018).

2.2.1 Time Series Analysis

Time series analysis is the process of examining and modeling time series data to extract meaningful statistics and identify underlying structures. This analysis includes techniques to detect trends (long-term movements), seasonality (repeated cycles) and irregular components (random fluctuations). Time series models such as

ARIMA, GARCH and machine learning algorithms can be used to predict future values based on historical patterns and provide insight into the temporal dynamics of the data.

2.3 Methods and Approaches

This section reviews common methods used in the analysis and prediction of cryptocurrency prices. It includes both traditional statistical models and modern machine learning techniques, along with the introduction of TDA as an alternative approach for capturing complex price behaviours across different cryptocurrencies.

2.3.1 Autoregressive Integrated Moving Average Models (ARIMA)

ARIMA model is widely used in time series analysis to forecast future values based on historical data. Its main function is to model and forecast non-stationary time series by combining the autoregressive (AR), integral (I) and moving average (MA) components. The autoregressive (AR) component models the current values of the series as a linear function of its past values, capturing the relationship between current and previous observations. The Integration part solves the non-stationary problem by differencing the series, removing the trend and ensuring that the data becomes stationary, which is essential for reliable forecasting. Finally, the moving average component models the error term of the series by correcting the forecast using past forecast errors (Mondal et al., 2014). The mathematical expression for the ARIMA model is given in Equation (1).

$$\left(1-\sum_{i=1}^{p} a_i L^i\right) X_t = \left(1+\sum_{i=1}^{a} \theta_i L^i\right) \epsilon_{\mathrm{t}} \tag{1}$$

where t is an integer index, X_t are real numbers, L is the lag operator, a_i is the autoregressive coefficients, θ_i are the moving average coefficients, and ϵ_t represents the error terms. By combining these components, ARIMA is effective in capturing time dependence, correcting trends, and improving forecasting accuracy, making it valuable in areas requiring time series forecasting, such as economic forecasting and stock market analysis (Wikipedia Contributors, 2019).

2.3.2 Generalized Autoregressive Conditional Heteroskedasticity (GARCH)

The GARCH model was introduced by Robert F. Engle in 1982 to estimate and predict financial market volatility with variance varying over time. It captures clusters of volatility by modeling volatility as dependent on past error terms and variances. The GARCH model is widely used by financial institutions to help estimate the volatility of returns on assets such as stocks and bonds, to assist in pricing, optimizing portfolios and making risk management decisions. The model involves fitting autoregressive models, calculation of autocorrelation of error terms, and significance testing, making it an important tool for financial forecasting (Kenton, 2020).

2.3.3 Random Forest

A random forest is a classifier that consists of a series of tree-structured classifiers, each of which works independently using random vectors of the same distribution. Each tree in the forest votes for the most common category based on the input data. As each tree is constructed using the training data, a new random vector is generated that is independent of the previous random vectors but follows the same distribution. Random forests use an upper bound to estimate the generalization error, which is based on two factors, accuracy and interdependence between individual classifiers (Mahdi & Mohsin Abdulazeez, 2021). Figure 2.1 shows flow chart of Random Forest

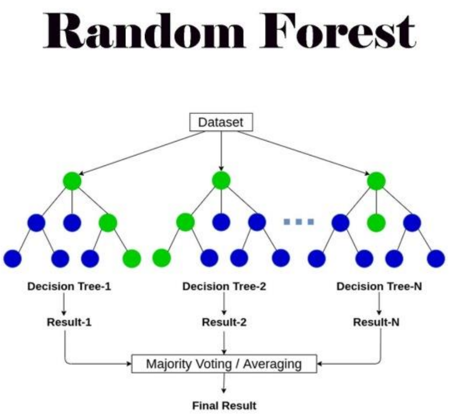


Figure 2.1 Flow chart of Random Forest (Mahdi & Mohsin Abdulazeez, 2021)

2.3.4 Support Vector

Support Vector Machine (SVM) is a supervised learning algorithm for classification that constructs a hyperplane to separate two datasets based on labeled training data. This hyperplane serves as a classification boundary that distinguishes the classes of the data. During the model construction process, support vectors (SVs) are identified as key data points that define the boundary. Its main goal is to maximize the margin between the two categories, which helps prevent overfitting of the training data, a common problem that limits the generalization of machine learning models.

In addition to classification, Support Vector Regression (SVR) applies the fundamentals of SVM but is used for regression tasks. SVR attempts to find the function that best fits the data within specified tolerances. The goal of SVR is not to classify the data points, but to predict continuous values by minimizing the error within a certain threshold, and is therefore suitable for both prediction and regression problems

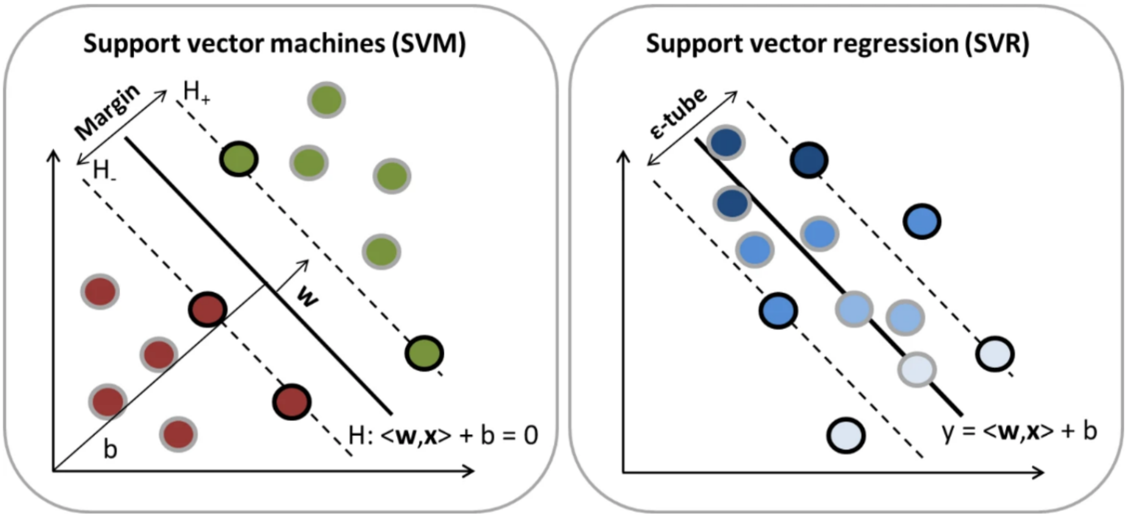


Figure 2.2 Examples of SVM and SVR (Rodríguez-Pérez & Bajorath, 2022)

Figure 2.2 illustrates the difference between SVM, which separates data into distinct classes using a hyperplane, and SVR, which fits a regression line within an ϵ-tube to predict continuous values (Rodríguez-Pérez & Bajorath, 2022).

2.3.5 Neural Network

A neural network is a machine learning model inspired by the human brain, designed to make decisions by mimicking the way biological neurons work together. It consists of multiple layers of nodes (artificial neurons), including an input layer, one or more hidden layers, and an output layer. Each node is connected to the others through associated weights and thresholds, if a node's output exceeds a threshold, it is activated to pass data to the next layer. Neural networks use training data to improve their accuracy, making them effective tools for tasks such as speech and image recognition. Often referred to as artificial neural networks (ANN) or simulated neural networks (SNN), they are the basis of deep learning models. An example of the deep neural network is illustrated in Figure 2.3.

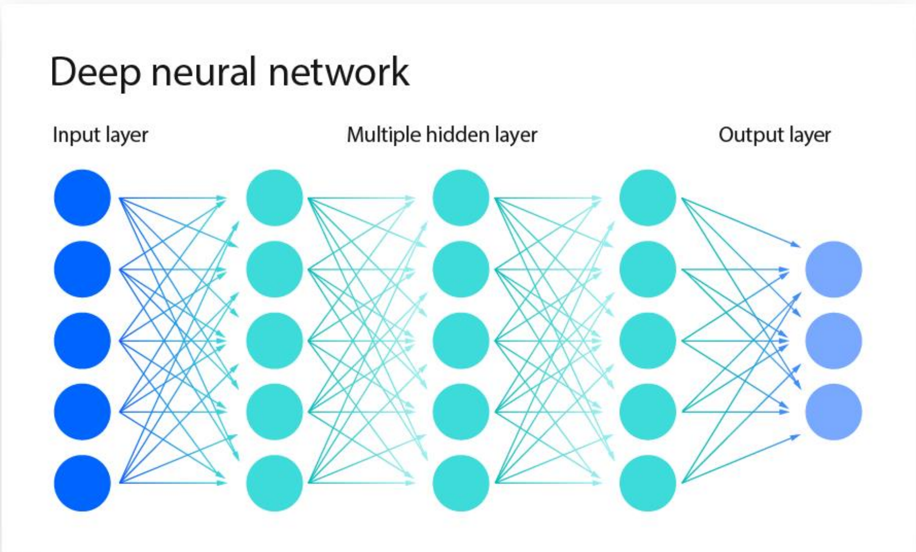


Figure 2.3 Example of Deep Neural Network (igmguru, 2025)

2.3.6 Logistic Regression

Logistic regression is a supervised machine learning algorithm commonly used for binary classification tasks, such as predicting whether an event will occur or not. The model applies the logit transformation to predict the log-odds of an event rather than the event's probability directly. This ensures that the predicted values remain between 0 and 1, which is essential for modeling probabilities. The general formula for the logistic regression model is:

$$\ln\left(\frac{P(y=1)}{1-P(y=1)}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n \tag{2}$$

where $P(y = 1)$ is the probability of the event, x_1, x_2, ..., x_n are the explanatory variables and β_0, β_1, ..., β_n are the model coefficients. The left-hand side of the equation represents the log-odds of the event occurring, and the right-hand side is the linear combination of explanatory variables, each weighted by its corresponding coefficient. This transformation allows for valid predictions of probabilities, as the log-odds are unbounded, unlike probabilities that must lie between 0 and 1 (Buis, 2017).

2.3.7 Topological Data Analysis

Topological data analysis (TDA) is a method used to understand the shape or structure of data. Instead of focusing on just numbers or statistics, topological data analysis looks at how data points are connected to each other, for example, whether form clusters, loops or empty spaces. These patterns help to reveal hidden information in complex or high-dimensional data.

The process begins by transforming the raw data into a point cloud, where each point represents an observation. The location of these points in space depends on how similar they are to each other. TDA then constructs shapes (called simple complexes) from these points to capture their connections at different scales or distances.

Next, TDA uses a tool called persistent homology to track how topological features appear and disappear as the scale changes. The result is a persistence diagram or barcode that shows how long each feature persists. Features that persist at multiple scales tend to be more meaningful. An example of the topological data analysis process diagram is shown in Figure 2.4.

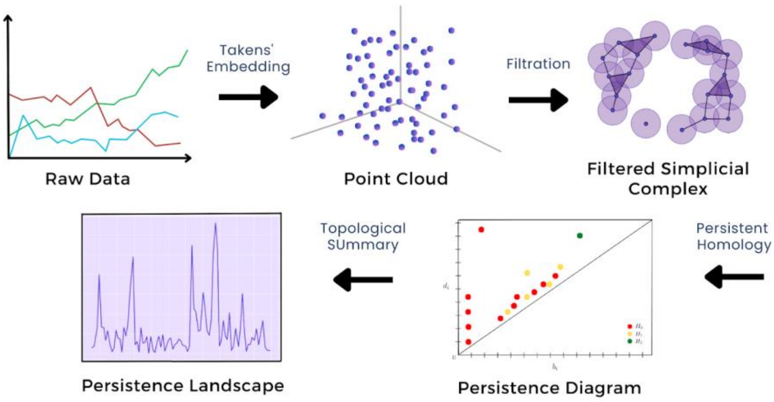


Figure 2.4 Topological Data Analysis (Harvey L. Sapigao, 2024)

2.4 Related Works

This section describes research on Bitcoin price prediction over the past few years. Many researchers have explored various machine learning and statistical methods such as ARIMA, GARCH, Random Forests, Support Vector Machines (SVM), Neural Networks, and hybrid models combining Logistic Regression with Long Short Term Memories (LSTM). These methods aim to improve prediction accuracy by leveraging historical data and addressing the uniqueness of Bitcoin's high volatility and market behaviour.

Azari (2019) used an ARIMA model to predict the price of Bitcoin and found that the model worked well in short-term predictions but struggled to work during periods of high price variability, such as late 2017. The model was trained on three years of data and the optimal ARIMA settings were found through a grid search to minimize prediction errors. However, ARIMA experienced difficulties in dealing with large price declines and increases. The study suggests that while ARIMA may be useful during more stable periods, it may not be sufficient to predict the price of bitcoin during volatile periods, and may require more features or complex modeling.

Besides, Naimy et al. (2021) investigated using GARCH models to predict the volatility of cryptocurrencies and fiat currencies. Their results show that the GARCH model effectively captures the time-varying volatility of cryptocurrencies such as Bitcoin and Ripple, with long term volatility reaching 213% for Bitcoin and 164% for Ripple. Cryptocurrencies exhibit higher and more persistent volatility than fiat currencies, which have volatility between 7% and 10%, and the Japanese Yen, which has a volatility of 55%. The study highlights the advantages of advanced GARCH models such as CGARCH and GJR-GARCH in accurately modeling the volatility of cryptocurrencies, which are more sensitive to market shocks than the more stable and predictable volatility of fiat currencies.

Chen (2023) used Random Forest Regression (RFR) to predict the price of Bitcoin and compared its performance on different datasets. The model achieves a high level of accuracy, with RMSE values of 0.017 and 0.035 for the two periods. The study concludes that RFR outperforms other models during stable price periods, but faces challenges during volatile “bubble” periods. This highlights the strength of

RFR in dealing with stable market conditions but also points to its limitations in capturing dramatic market fluctuations.

In addition, Ali (2020) used SVR to predict cryptocurrency prices, focusing on Bitcoin, XRP, and Ethereum. The study compares three kernels: linear, polynomial, and radial basis function (RBF). The results show that the RBF kernel outperforms the other kernels and has the highest accuracy, with R^2 values of 78% for Bitcoin, 67% for XRP, and 56% for Ethereum. The optimal values of the “Gamma” parameter for the polynomial kernel and the RBF kernel are 0.00001 and 0.00005, respectively, while the optimal error penalty (“C”) values are 100 and 1000, respectively. The study concludes that the RBF kernel is a good choice for nonlinear data modeling and the processing of high-dimensional cryptocurrency datasets is particularly effective.

Jay et al. (2020) explored the use of stochastic neural networks in cryptocurrency price prediction, focusing on their ability to capture complex market patterns. The study found that stochastic neural networks outperformed traditional deterministic neural networks, with performance improvements ranging from 1.56% to 7.41% depending on the dataset and hyperparameter settings. The findings highlight the effectiveness of adaptive learning in modeling market volatility. The authors suggest that future research could further explore optimization techniques for hyperparameters and attempt to use stochastic response functions to better model market behaviour.

Andi (2021) presented a hybrid model combining logistic regression and LSTM for bitcoin price prediction with 97.2% accuracy, 98.32% precision, 98.22% recall, and 95.35% sensitivity. The hybrid approach outperforms the standalone model, demonstrating the effectiveness of combining machine learning techniques to improve predictions. The use of large datasets helps to solve the overfitting problem and improve the reliability of predictions.

Gidea et al. (2018) apply TDA to the study of critical transitions in cryptocurrency markets, using persistence landscapes derived from time-delayed embeddings of logarithmic yield series. Their results show that L_1-norm peaks in the persistence landscape correspond to major market crashes, suggesting that TDA has the potential to detect nonlinear patterns and structural shifts. Despite the method's promise, the study also points out its limitations, such as false positives, which emphasize the need to combine TDA with machine learning to improve the accuracy of predictions for highly volatile markets.

Lastly, Ruiz-Ortiz et al. (2022) apply TDA to financial markets by developing a new turbulence index based on persistence landscapes of log-return time series. They show that this index gives earlier warnings of market crashes, such as during the 2020 COVID-19 crisis, compared to traditional measures like the VIX and Mahalanobis distance. Their TDA-based strategies also outperform standard investment approaches, achieving higher Sharpe ratios and lower drawdown. They further apply the method to cryptocurrency markets and find it effective at detecting upcoming crashes, especially when combined with machine learning, though some false positives remain.

These papers are summarized by categorizing key aspects such as machine learning algorithms, evaluation metrics, limitations, strengths, and conclusions. The summary in Table 2.1 provides a clear comparison of the different approaches to Cryptocurrencies price prediction, highlighting the strengths and challenges of each approach.

2.5 Comparison of Related Works

Table 0.1 Comparison of findings in related works

Author(s)	**Machine Learning Algorithm**	**Evaluation Metrics**	**Strengths**	**Limitations**	**Conclusion**
Azari (2019)	ARIMA	AIC, RMSE, MAPE	Well-suited for time series forecasting, simple and interpretable	Struggles during volatile periods, assumes stationarity	ARIMA works well for short-term predictions, but struggles during high volatility periods like late 2017. It may need more features or complex models to improve accuracy during these times.
Naimy et al. (2021)	GARCH-type models	Volatility prediction accuracy	Effective for capturing time-varying volatility, useful for risk management	Limited in predicting actual price changes, focus mainly on volatility	GARCH models effectively capture time-varying volatility in Bitcoin and cryptocurrencies, providing reliable volatility estimates crucial for risk management.
Chen (2023)	Random Forest Regression (RFR)	RMSE, prediction accuracy	High prediction accuracy in stable periods	Struggles during volatile "bubble" periods, limited by dataset size	RFR outperforms other models during stable price periods but struggles during volatile "bubble" periods.
Ali Alahmari (2020)	Support Vector Regression (SVR)	MAE, MSE, RMSE, R^2	Strong performance with Radial Basis Function (RBF) kernel, handles nonlinearity well	Overfitting risks with large datasets, sensitivity to hyper parameters	SVR, particularly with the Radial Basis Function (RBF) kernel, outperforms other kernels and models for Bitcoin price prediction.

Jay et al. (2020)	Stochastic Neural Networks	Prediction accuracy, relative improvement	Stochastic versions outperform deterministic models, better capturing market volatility	Hyper parameter tuning is crucial for optimal results	Stochastic neural networks outperform deterministic models, with improvements in prediction accuracy.
Andi (2021)	Hybrid Logistic Regression with LSTM	Accuracy, precision, recall, sensitivity	Hybrid model achieves high accuracy, precision, and recall	Overfitting risks with large datasets, needs careful model tuning	The hybrid model combines logistic regression with LSTM to achieve superior accuracy, precision, and recall in Bitcoin price prediction.
Gidea et al. (2018)	TDA + k-means clustering	L_1-norm of persistence landscapes, cluster stability	Detects topological changes indicative of critical transitions; robust to nonlinear behaviour	False positives for non-critical swings; dependent on empirical choices for parameters	Promising method to detect approaching market crashes, further refinement needed to improve precision.
Ruiz-Ortiz et al. (2022)	TDA + k-means clustering	Sharpe ratio, drawdown, early warning accuracy	Detects market regime shifts, strong early warning signals, robust to noise	Some false positives in crypto markets, less common in finance	TDA-based turbulence index detects market crashes earlier than VIX and other indicators. Combined with machine learning, it improves risk management and portfolio returns.

Table 2.1 provides a comparative analysis of various machine learning algorithms, highlighting their strengths and weaknesses in the areas of time series forecasting and cryptocurrency price prediction. ARIMA is widely recognized for its simplicity and ease of interpretation but tends to be problematic during periods of high volatility. GARCH-type models are adept at capturing volatility but fall short of predicting price movement. RFR shows high predictive accuracy under stable market conditions but falls short when the market is in a bubble. SVR, especially when using RBF kernels, is excellent at handling non-linear data, but is prone to overfitting and hyperparameter selection. Stochastic neural networks are good at capturing market fluctuations but require careful fine-tuning to achieve optimal results. Hybrid methods, such as Logistic Regression combined with LSTM, can provide impressive accuracy but can run into overfitting issues when dealing with large datasets. Finally, TDA combined with k-means clustering resists the effects of non-linear patterns and identifies important transitions, but has the potential to false positive on less important fluctuations. When combined with distance-based landscape measurements and applied to equity markets, TDA can also improve early warning signals and support more effective risk-adjusted investment strategies. In summary, TDA is a promising method for detecting key transitions in the time series as it is effective in capturing non-linear patterns and structural changes that precede major market events. Its robustness to noise and ability to reveal hidden topological signals make it particularly useful in volatile environments such as cryptocurrency markets.

2.6 Tools

In this project, Python-based machine learning tools and libraries will be used to develop and implement predictive models.

2.6.1 NumPy

NumPy stands for Numerical Python and provides powerful tools for working with n-dimensional arrays and matrices. It optimizes mathematical operations through efficient array vectorization, which greatly improves computational performance (Knez, 2023).

2.6.2 Pandas

Pandas is a library for working with labeled and relational data using two key data structures: series and data frames. It provides a wide range of features for simplifying data manipulation and visualization (Knez, 2023).

2.6.3 Matplotlib

Matplotlib is a versatile library for creating visualizations. It provides more control over visualizations that require additional coding to achieve more complex or customized visual representations (Knez, 2023).

2.6.4 Persim

Persim is a Python library that specializes in distances and representations of persistence diagrams, providing tools for persistent images, persistent landscapes, and various distance metrics.

2.6.5 Ripser

Ripser is a lightweight Python persistent homology library designed to efficiently compute and visualize persistent homology on sparse and dense datasets.

2.7 Summary

This chapter provides an overview of the various machine learning models and tools used for cryptocurrencies price prediction. The main models discussed include ARIMA for time series forecasting, GARCH for volatility modeling, Random Forest for regression, SVM for classification and regression, neural networks for complex pattern recognition, logistic regression for binary classification tasks and TDA is introduced as a method for detecting critical transitions in Bitcoin price time series. In addition, this chapter introduces basic Python libraries such as NumPy for numerical operations, Matplotlib for data visualization, Pandas for data manipulation, and specialized packages such as Persim and Ripser for persistence diagram analysis and persistent homology computation.

Chapter 3

Methodology

3.1 Overview

This chapter describes the methodology used in the project. It provides an overview of the Knowledge Discovery in Databases (KDD) process as a framework for analyzing Bitcoin price data, detailing the steps from data collection to generating actionable insights.

3.2 Knowledge Discovery in Databases (KDD)

Knowledge Discovery in Databases is finding useful and meaningful knowledge from large amounts of data. It involves seven key steps, which are cleaning, integration, selection, transformation, mining, evaluation, and presentation, and these steps work together to transform raw data into valuable insights. As businesses and organizations collect massive amounts of data, KDD helps transform this data into insights that can guide better decisions and strategies (Tarud, 2024). KDD is needed because the raw data itself is not valuable until it is analyzed and understood, and it uses techniques such as machine learning to uncover hidden patterns and trends that are not readily apparent. Sometimes, organizations may use KDD to discover hidden trends and generate predictions by analyzing historical data patterns.

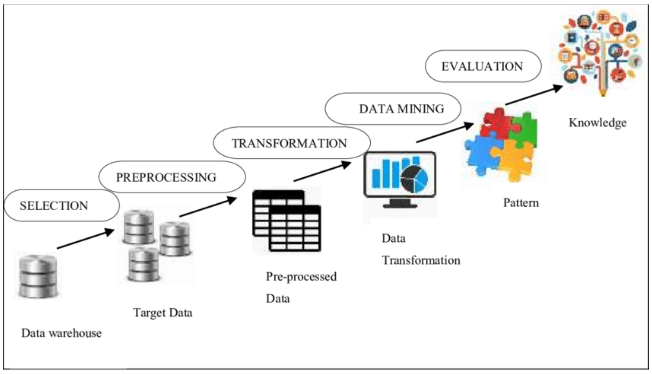


Figure 3.1 Knowledge Discovery in Databases process (Ahmad Sabri et al., 2019)

3.2.1 Data Collection

The dataset used in this study is obtained from Kaggle, specifically historical price data for Bitcoin. The dataset includes market data recorded, such as opening, closing, high and low prices, as well as the daily trading volume for each minute. For this study, only daily closing prices are considered, which are obtained by selecting closing prices at specific times of the day. This data spans from January 1, 2012 to the present, to provide a comprehensive picture of Bitcoin's price trends. This dataset is used as the primary input for applying TDA to explore and analyze the topology of Bitcoin's price trend.

3.2.2 Data Preprocessing

Data preprocessing involves preparing raw Bitcoin data by ensuring its quality and consistency. The first step is to process any missing or incorrect values in the dataset. If any data points such as daily closing prices are missing, they are either imputed or deleted with appropriate values depending on the amount of missing data. Data entry errors or extreme market conditions may result in outliers or anomalies, which will also be identified and dealt with appropriately. After the dataset has been cleaned, it will be standardized to ensure uniformity of format and that all values are consistent in scale and structure in preparation for further analysis.

3.2.3 Data Transformation

In this step, the Takens embedding theorem converts daily closing prices into a suitable format for TDA. The theorem provides a method for reconstructing a multidimensional state space from one-dimensional time series data. By applying this method, one-dimensional bitcoin price data is converted into a two-dimensional representation, which enables the study of its underlying topological features.

The embedding process consists of creating a series of delay vectors from the time series data, each representing a snapshot of the price dynamics within a specific time window. These vectors are constructed by choosing a fixed embedding dimension and delay parameter that determines the number of data points in each vector and the time interval between data points. This transformation ensures that the dataset captures the time-dependent and localized structure of Bitcoin price movements.

By applying the Taken embedding, the transformed two-dimensional data can identify topological features such as clusters, loops and holes that represent key patterns and structures in the price dynamics.

3.2.4 Data Mining

Data mining is the process of discovering patterns, structures and relationships in data that are not initially obvious. In this study, TDA is applied to Bitcoin price data to discover hidden patterns. TDA utilizes persistent homology to study the shape of data at different scales, revealing features that may have been overlooked by traditional methods. The goal of this process is to transform Bitcoin's daily closing price data into topological representations, such as persistence diagrams, which help to identify the underlying structure, trends and changes in price movements. Persistence diagrams capture the birth and death of topological features at different scales and are the main output of this data mining process.

3.2.5 Evaluation

The generated persistence diagram will be analyzed to assess the long-term price trends, stability and volatility of Bitcoin. The evaluation will focus on identifying persistent topological features that reflect stable trends and short-term features that reflect market volatility. By examining the persistence of these features, the stability of the Bitcoin price over time can be assessed. This step also includes detecting anomalies in the price data, focusing on sudden changes or patterns that may signal unusual market behaviour and analyzing Bitcoin's market cycles by examining the evolution of topological features to identify cyclical trends or shifts between different market phases.

3.3 Summary

This chapter provides a detailed description of applying the Knowledge Discovery in Databases (KDD) framework to analyze Bitcoin price data. The process begins with data collection, gathering historical daily closing prices from reliable sources. This is followed by data preprocessing, which deals with missing values and outliers to ensure the consistency and quality of the dataset. Next is the data transformation step, where the cleaned data is converted to a two-dimensional representation using the Takens Embedding Theorem to reconstruct the state space and capture local trends and fluctuations. In the data mining stage, topological data analysis is used to generate persistence diagrams that reveal hidden patterns and data shapes at different scales. Lastly, the evaluation step involves analyzing these persistence diagrams to assess long-term trends, stability, anomalies and market cycle dynamics.

Chapter 4

Implementation

4.1 Introduction

This chapter provides a brief overview of the implementation process and the execution of the methodology. It highlights the specific steps for applying Topological Data Analysis to Bitcoin price data. The implementation process used Python in Visual Studio code. Python was chosen because of its large number of built-in libraries that support data processing and mathematical operations. Python's beginner-friendly and well documentation make it ideal for research and academic projects. In this project, the dataset was collected from Kaggle, a well-known platform for data science competitions and research datasets. Kaggle provides reliable, publicly available financial data, including historical Bitcoin price information, which is well suited for this analysis.

4.2 Environment Setup

The implementation was carried out using Python in Visual Studio Code. Several important libraries were used, including *pandas* and *numpy* for data handling and numerical operations, *matplotlib.pyplot* for plotting, *ripser* for computing persistent homology, *persim* for visualizing persistence diagrams, and *sklearn.metrics* for calculating pairwise distances in the data.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from ripser import ripser
from persim import plot_diagrams
from sklearn.metrics import pairwise_distances
```

Figure 4.1 Import Libraries

4.3 Data Collection

After downloading the bitcoin price dataset from Kaggle, the file was saved as 'bitcoindata.csv'. The dataset was imported into the Python environment by the *pandas* library as shown in Figure 4.2. This dataset contains historical market data for Bitcoin spanning from 1 January 2012 to the present, with data continuously updated. It includes market data recorded, such as opening, closing, high and low prices, as well as the daily trading volume for each minute. Figure 4.3 shows a screenshot of the dataset opened in Excel visualizing the structure and content of the dataset.

```
#Load the CSV file into a DataFrame
df = pd.read_csv("bitcoindata.csv")
```

Figure 4.2 Load Bitcoin dataset into DataFrame

	A	B	C	D	E	F
1	Timestamp	Open	High	Low	Close	Volume
2	1325412060	4.58	4.58	4.58	4.58	0
3	1325412120	4.58	4.58	4.58	4.58	0
4	1325412180	4.58	4.58	4.58	4.58	0
5	1325412240	4.58	4.58	4.58	4.58	0
6	1325412300	4.58	4.58	4.58	4.58	0
7	1325412360	4.58	4.58	4.58	4.58	0
8	1325412420	4.58	4.58	4.58	4.58	0
9	1325412480	4.58	4.58	4.58	4.58	0
10	1325412540	4.58	4.58	4.58	4.58	0
11	1325412600	4.58	4.58	4.58	4.58	0
12	1325412660	4.58	4.58	4.58	4.58	0
13	1325412720	4.58	4.58	4.58	4.58	0
14	1325412780	4.58	4.58	4.58	4.58	0
15	1325412840	4.58	4.58	4.58	4.58	0
16	1325412900	4.58	4.58	4.58	4.58	0
17	1325412960	4.58	4.58	4.58	4.58	0
18	1325413020	4.58	4.58	4.58	4.58	0
19	1325413080	4.58	4.58	4.58	4.58	0
20	1325413140	4.58	4.58	4.58	4.58	0
21	1325413200	4.58	4.58	4.58	4.58	0
22	1325413260	4.58	4.58	4.58	4.58	0
23	1325413320	4.58	4.58	4.58	4.58	0

Figure 4.3 Excel screenshot of the Bitcoin dataset

4.4 Data Preprocessing

Several cleanup steps were performed to prepare the data for analysis. First, the 'Timestamp' column was converted to the appropriate date format so that the data could be arranged by date. Then, this column was set as the index of the dataset to facilitate time based operations. After that, any missing values in the dataset are filled in with the average of the respective columns to ensure that the data is complete and will not cause problems during analysis. Next, the data was resampled to show the average closing price for each day and the 'Close' column was chosen for the final price of Bitcoin at each point in time. The code for these steps is shown in Figure 4.4.

```
#Convert 'Timestamp' column to datetime format
df['Timestamp'] = pd.to_datetime(df['Timestamp'], unit='s')

#Set 'Timestamp' as the index
df.set_index('Timestamp', inplace=True)

#Handle missing values
df = df.fillna(df.mean())

#Resample the data to daily
df_daily = df.resample('D').mean()

# Choose a column (Close)
daily_data = df_daily['Close']
```

Figure 4.4 Code for converting timestamp, handle missing value and resampling data

	A	B	C	D	E	F
1	Timestamp	Open	High	Low	Close	Volume
2	1/1/2012	4.645697259	4.645697259	4.645697259	4.645697259	0.011918951
3	2/1/2012	4.975	4.975	4.975	4.975	0.007013889
4	3/1/2012	5.0855	5.0855	5.0855	5.0855	0.074364778
5	4/1/2012	5.17025	5.170395833	5.17025	5.170395833	0.074467542
6	5/1/2012	5.954291667	5.954361111	5.954291667	5.954361111	0.048839404
7	6/1/2012	6.620236111	6.620333333	6.620236111	6.620333333	0.038788773
8	7/1/2012	6.047777778	6.047777778	6.047777778	6.047777778	0.001941568
9	8/1/2012	6.819444444	6.819444444	6.819444444	6.819444444	0.002777778
10	9/1/2012	6.816513889	6.816513889	6.816451389	6.816451389	0.045048141
11	10/1/2012	6.385722222	6.385722222	6.385722222	6.385722222	0.042147181
12	11/1/2012	7.030256944	7.030333333	7.030256944	7.030333333	0.064810398
13	12/1/2012	6.97925	6.97925	6.97925	6.97925	0.068702522
14	13/1/2012	6.707305556	6.707645833	6.706680556	6.707118056	0.023351959
15	14/1/2012	6.492305556	6.492305556	6.492236111	6.492236111	0.019405086
16	15/1/2012	6.611076389	6.611076389	6.611076389	6.611076389	0.014538403
17	16/1/2012	7.071527778	7.071527778	7.071527778	7.071527778	0.008784233
18	17/1/2012	7.005972222	7.005972222	7.005972222	7.005972222	0.007809984
19	18/1/2012	6.458416667	6.458611111	6.458152778	6.458347222	0.140773889
20	19/1/2012	6.534388889	6.534388889	6.534388889	6.534388889	0.110045823
21	20/1/2012	6.475083333	6.475083333	6.475083333	6.475083333	0.12783044

Figure 4.5 Excel screenshot of the Bitcoin dataset after resampled

Figure 4.6 shows the code used to generate a line graph of the daily closing price over time. Once the plotting was complete, the cleaned and resampled data was saved to a new CSV file for future use. Figure 4.7 display line graph for the daily Bitcoin price data.

```
#Plot the daily data
plt.figure(figsize=(10, 6))
plt.plot(df_daily.index, daily_data, label='Daily Close Price', color='blue')
plt.title('Daily Close Price')
plt.xlabel('Date')
plt.ylabel('Close Price')
plt.legend()
plt.show()

# Save the resampled data to a new CSV file
df_daily.to_csv("daily_data.csv")
```

Figure 4.6 Code for generating line graph and saving daily data

Figure 4.7 Line graph of Bitcoin’s daily close price

From Figure 4.7, it is clear that the Bitcoin's price has gone up a lot over time, but with many ups and downs along the way. The overall trend is upward, but highly volatile. There are about four major high points in the graph. The first big jump happened at the end of 2017, followed by others in early 2021, late 2021 and the most recent one in 2025. The highest peak so far appears in 2025 (now). After each of these peaks, the price dropped again.

In the earlier years, from 2012 to 2016, Bitcoin's price stayed fairly low and steady. After that, the graph shows a repeating pattern of fast growth followed by sudden drops. This ups and downs behaviour suggests that there's a lot of speculation and that the market reacts strongly to different events. Because of this, Bitcoin is a good example to study Topological Data Analysis, which looks at patterns and structures in a complex data.

4.5 Data Transformation

One-dimensional Bitcoin price data must be converted into a high-dimensional format before applying topological data analysis. In this project, takens embedding theorem is used to convert one-dimensional time series into two-dimensional point clouds, which helps to reveal patterns and shapes of the data over time.

The cleaned data was loaded from a saved CSV file and a specific time range (from January 2021 to March 2021) was selected. This specific time range was selected because of the high level of price volatility and the stable rise in the middle of the period. In time series data analysis, this volatility is beneficial because it produces more variation and structure in the data in topological data analysis. This can produce richer persistent diagrams that capture more features and patterns. Then, time_series_to_point_cloud() function was used to perform Takens embedding. In this study dimension (*d*) = 2 and time delay (*tau*) = 10 were used. Finally, the two-dimensional point cloud was plotted to show the structure of the embedded data.

```
#Load the CSV file into a DataFrame
df_daily = pd.read_csv("daily_data.csv", index_col='Timestamp', parse_dates=True)

#Choose a time range (example)
df_daily = df_daily[(df_daily.index >= '2021-01-01') & (df_daily.index <= '2021-03-31')]

daily_data = df_daily['Close']

#Transforms a time series into a point cloud using time-delay embedding.
def time_series_to_point_cloud(ts, d, tau):
    N = len(ts)
    M = N - (d - 1) * tau
    if M <= 0:
        raise ValueError("Time series is too short for the given d and tau.")
    return np.array([ts[i:i + d*tau:tau] for i in range(M)])

d = 2       # embedding dimension
tau = 10    # delay
point_cloud = time_series_to_point_cloud(daily_data.values, tau, d)

# Plot the 2D embedded data
plt.figure(figsize=(6, 6))
plt.scatter(point_cloud[:, 0], point_cloud[:, 1], label='Point Cloud')
plt.xlabel('x(t)')
plt.ylabel('x(t + tau)')
plt.title('Point Cloud')
plt.grid(True)
plt.legend()
plt.show()
```

Figure 4.8 Code for implementing Takens embedding theorem

A visual representation of market dynamics is provided by the point cloud visualization in Figure 4.9, which was created from Bitcoin price data using taken embedding. The pattern shows two distinct point clusters, suggesting that there were two distinct price levels at which Bitcoin prices were concentrated during this period. This clustering suggests a shift in market behaviour, possibly a transition from a stable phase to a rising or volatile phase. The space between the clusters contains scattered points, which may represent price transitions or unstable fluctuations. Within each cluster, the points remain arranged diagonally, indicating a degree of consistency in short-term prices. In essence, this point cloud presents a more complex structure and behaviour compared to a single dense cluster and is therefore particularly suitable for in-depth topological analysis.

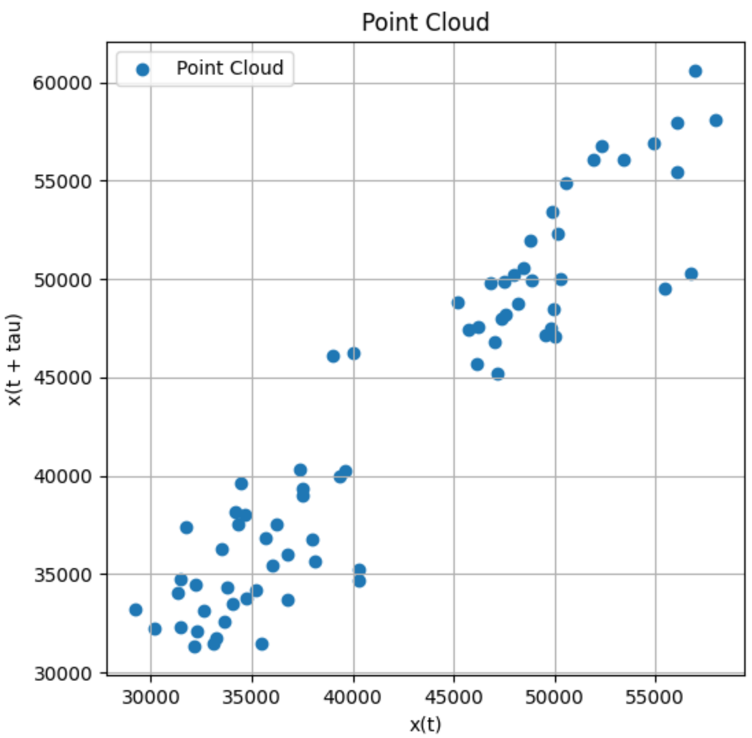


Figure 4.9 Point cloud for Bitcoin dataset from January 2021 to March 2021

4.6 Data Mining

The *ripser* library was used to compute the persistent homology, which reveals the instant when topological features appear and disappear as the scale changes, while the *persim* library was used to visualize the persistent diagrams. Figure 4.10 shows the code used to apply TDA and construct the persistence diagrams.

```
# Compute persistent homology
diagrams = ripser(point_cloud)['dgms']

# Plot the diagrams
plot_diagrams(diagrams, show=True)
```

Figure 4.10 Code for generating and plotting persistence diagrams

Figure 4.11 presents the persistence diagram created from the point cloud using persistent homology. This diagram shows two kinds of topological features, H_0 (blue dots), which represent connected components and H_1 (orange dots), which represent loops. Each dot marks the instantiation when a feature appears (born) and when it disappears (dies) as the filtration scale increases. Features that appear and disappear quickly, which close to the diagonal line are usually considered short-lived or possibly just noise. In contrast, features that are farther from the diagonal may represent more meaningful structures in the data. A more detailed discussion of these patterns will be provided in Chapter 5.

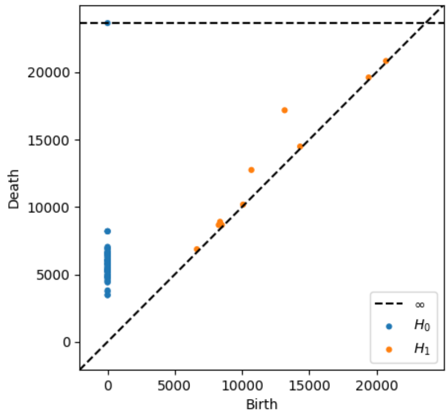


Figure 4.11 Persistence diagram for Bitcoin dataset from January 2021 to March 2021

4.7 Summary

This chapter describes the implementation of topological data analysis, starting from setting up the environment and preparing the Bitcoin dataset. The data was cleaned, resampled and converted into a point cloud using Takens embedding theorem. Finally, topological data analysis was applied using persistent homology to generate persistence diagrams.

Chapter 5

Results and Discussion

5.1 Introduction

This chapter presents the results from the topological data analysis of Bitcoin's daily closing price data. The analysis focuses on the visual patterns found in the point clouds and persistence graphs generated in Chapter 4. Interpretation of these results contributes to a better understanding of Bitcoin's price behaviour, including trends, volatility, stability and market phase transitions.

5.2 Point Cloud Analysis

Takens embedding (time_series_to_point_cloud function) was used to convert Bitcoin price data into point clouds. Depending on the time period, each point cloud reveals a different type of price behaviour (Ravishanker & Chen, 2019). By comparing point clouds for flat, trending and volatile periods, we can have a better observation of how market behaviour affect the overall data structure. These structural differences are important for identifying patterns that may not be apparent in line graphs and serve as a foundation for the persistence diagrams discussed in the next section.

5.2.1 Flat Stable Period (March–May 2012)

Figure 5.1 shows a point cloud from Bitcoin's flat period in early 2012, when the price of Bitcoin stayed low and fluctuated very little. The dots are tightly packed along a narrow diagonal line, showing minimal daily fluctuations. This compact shape reflects a calm market phase with little volatility and highly predictable short-term price behaviour.

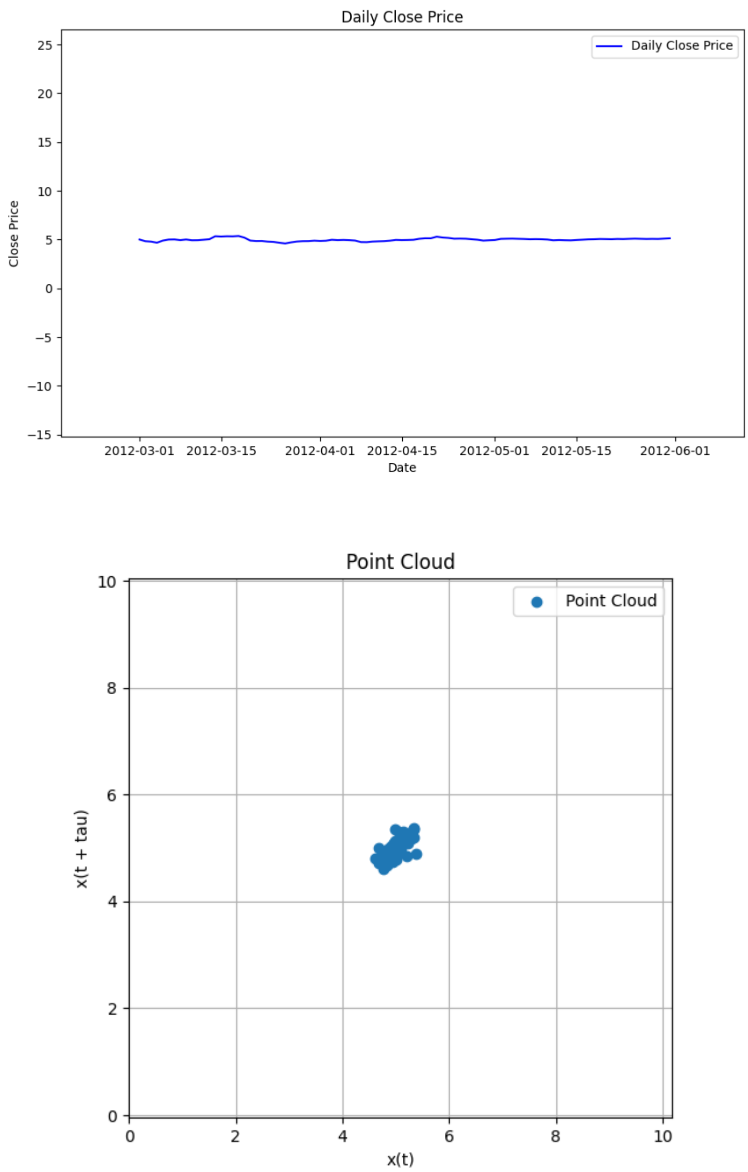
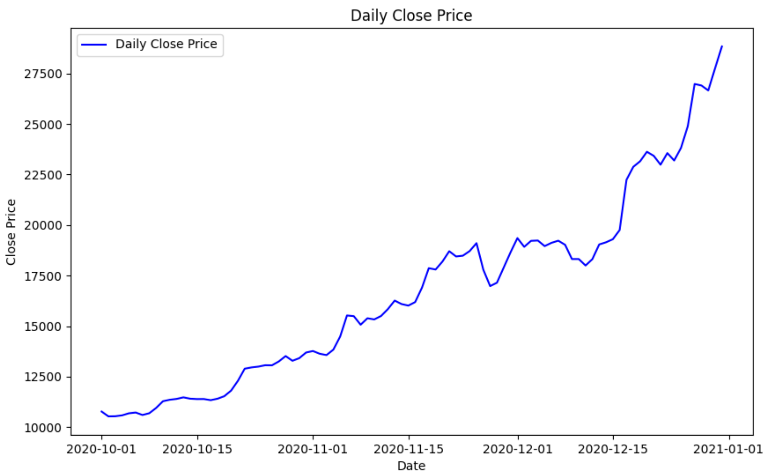


Figure 5.1 Point cloud for flat and low-price stability

5.2.2 Trend-Stable Period (October–December 2020)

Figure 5.2 shows the point cloud for the period of stable growth in the last quarter of 2020. The points are still arranged along the diagonal, but are more loosely distributed than during the flat stable period. This shape reflects a continuing upward trend in prices, with relatively smooth progress, that is no sharp jumps or declines.



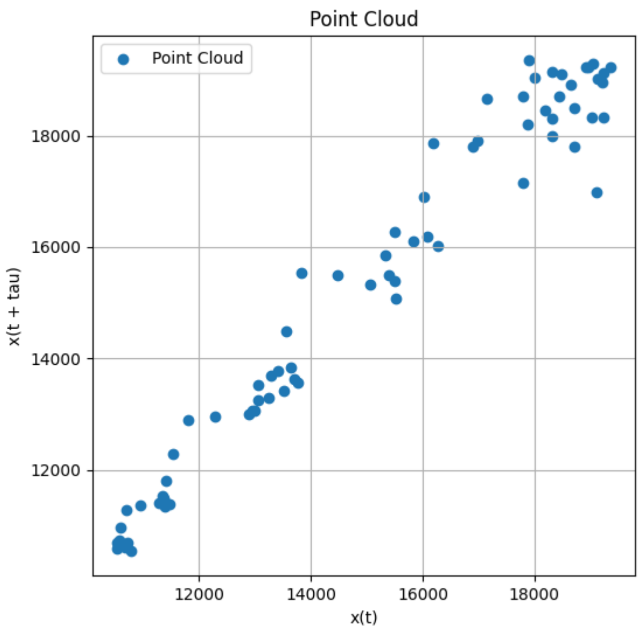
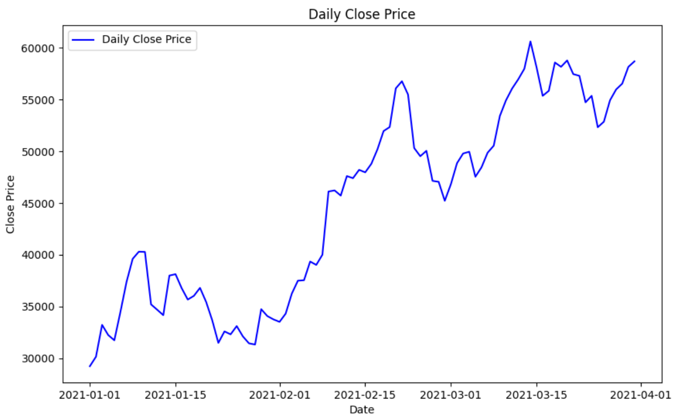


Figure 5.2 Point cloud for stable upward price movement

5.2.3 Strong Bullish Uptrend (January–March 2021)

Figure 5.3 shows a point cloud of the early 2021 phase of rapid price growth for Bitcoin. There is a noticeable gap in the middle, which was caused by the price moving quickly through certain levels without staying there for a long time. This ‘skip range’ indicates a fast-moving market with prices jumping from lower to higher levels with fewer intermediate values. It reflects a strong bullish trend with momentum-driven behaviour.



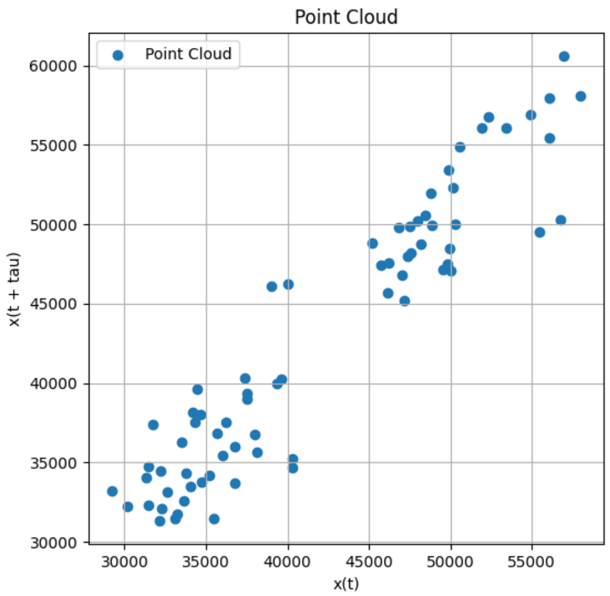


Figure 5.3 Point cloud for strong bullish price movement

5.2.4 Volatile Period (March–May 2024)

Figure 5.4 shows the point cloud for the highly volatile phase in 2024. Compared to the other phases, the points in this phase are more dispersed, forming multiple loose clusters with a less clear structure. This indicates frequent and irregular price fluctuations and erratic short-term behaviour. The shape of this point cloud reflects a chaotic market environment with both upward and downward movements in a short period of time.

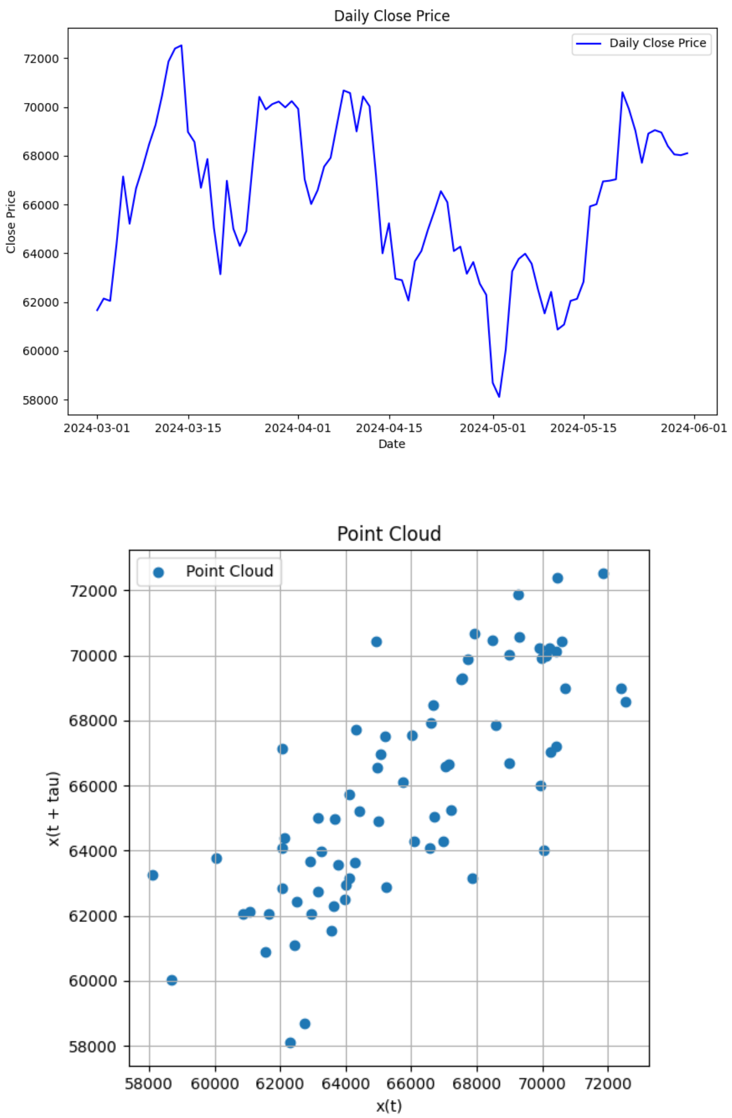


Figure 5.4 Point cloud for volatile or unstable price movement

5.3 Persistence Diagram Interpretation

Persistence diagrams are important in topological data analysis for visualizing how topological features in a dataset. Each point represents a topological feature in these diagrams, with the horizontal-axis indicating the birth time of the feature and the vertical-axis indicating the death time of the feature. Two main types of features are usually represented: H_0 (blue dots), which correspond to connected components of the data and H_1 (orange dots), which represent loop or circular structures. Persistent H_0 features indicate that the connected components in data, while persistent H_1 features indicate the presence of meaningful loops or circular behaviour.

5.3.1 Flat Stable Period (March–May 2012)

The persistence diagram in Figure 5.5 shows that all H_1 features are very close to the diagonal, meaning they are short-lived and likely caused by noise. This indicates that repeatable or cyclic behaviour was almost entirely absent during this flat period. The price stayed relatively constant and no strong topological loops were formed.

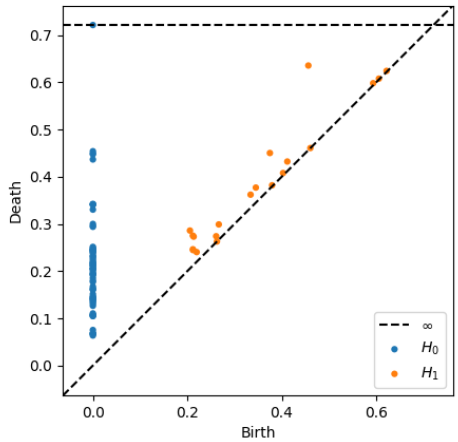


Figure 5.5 Persistence diagram for flat and low-price stability

5.3.2 Trend-Stable Period (October–December 2020)

Figure 5.6 shows persistence diagram for stable upward price movement contains only two H_1 features, both weak and short-lived. This reflects the predictability and stability of the market during this period. The lack of consistent cycling suggests a smooth and consistent upward trend in the price of Bitcoin, with few major reversals or cyclical behaviour. This simplicity explains why only very few topological features appear.

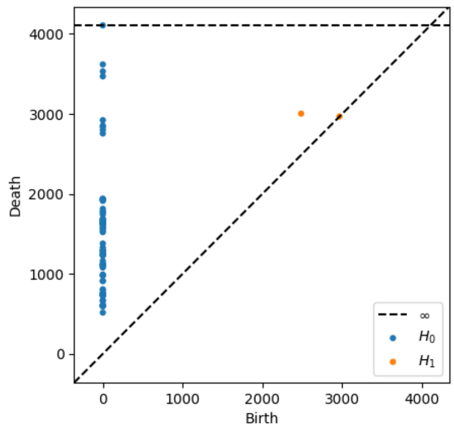


Figure 5.6 Persistence diagram for stable upward price movement

5.3.3 Strong Bullish Uptrend (January–March 2021)

Figure 5.7 shows a diagram with more widely distributed H_1 features, some of which are located much farther from the diagonal. These persistent loops suggest stronger and more significant cyclic patterns in the data, which may be caused by local corrections or temporary reversals within the overall upward movement. The presence of several long-lived features indicates that although the market was rising, there was more complex than a simple straight trend.

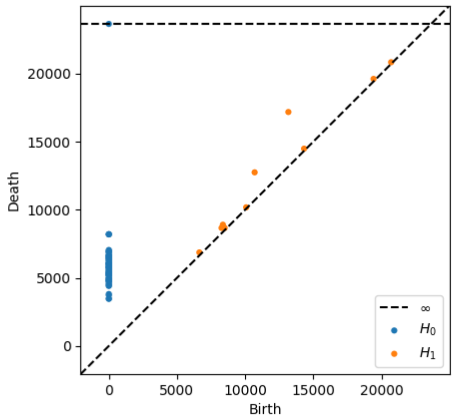


Figure 5.7 Persistence diagram for strong bullish price movement

5.3.4 Volatile Period (March–May 2024)

Figure 5.8 presents the persistent diagram with most significant topological features. The H_1 features are not only more, but many of them are also persistent, as they are located far from the diagonal. This suggests a high number of recurring patterns and strong loops in the data, which aligns with a volatile market with full of sharp reversals, spikes and short-term fluctuations. The density and distribution of these features reflect an unstable phase where price behaviour is less predictable and cycles are more chaotic.

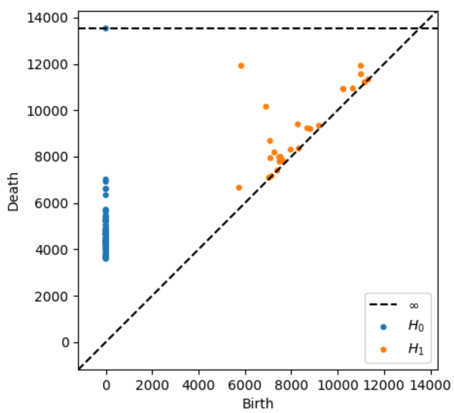


Figure 5.8 Persistence diagram for volatile or unstable price movement

5.3.5 Summary of Diagram Patterns

In all diagrams, the density and distribution of H_1 features increase as market activity becomes more intense. During the flat stable period, almost all features are short-lived and close to the diagonal, mostly can be considered as noises. The trend stable period has only two weak loops, reflecting the steady movement of prices, with minimal structure. The bullish uptrend introduces more persistent features, showing stronger patterns and short-term cycles. The volatile period shows the most persistent loops, indicating complex and unstable price behaviour with meaningful repeating structure.

5.4 Summary

This chapter presented the results of applying topological data analysis to Bitcoin price data across different market conditions. Through point cloud analysis, it is observed how price behaviour changes during flat, stable, bullish and volatile periods. The shape and distribution of the point cloud reflected clear differences between calm and active markets.

Persistence diagrams provided further insights into the hidden topological structures of the data. During stable periods, H_1 features are few and short-lived, indicating simple or noisy behaviour. In contrast, volatile and trending markets show more persistent cycles, suggesting complex patterns and possibly recurring price structures. These findings suggest that TDA is effective in revealing structural differences in Bitcoin's price action.

Chapter 6

Conclusion and Future Works

6.1 Introduction

This chapter summarises the main contributions of the project, discusses limitations and suggests potential directions for future work. The aim of this project was to explore Bitcoin price behaviour using topological data analysis, with a focus on identifying hidden patterns and structural differences across market conditions. The following section reflects on the results achieved in this project and how the findings may support for further research in this area.

6.2 Contributions

This project explores a new way to understand the behaviour of Bitcoin prices by applying topological data analysis, a method rarely used in financial analysis. Instead of relying on traditional time-series models, the project uses Takens embeddings and persistence diagrams to study the shape and structure of bitcoin price data. This allowed for a deeper understanding of hidden patterns and topological features in different market phases.

The main outcome of this project is the demonstration that TDA can successfully capture meaningful structural changes in financial data.The project showed clear differences between stable, trending and volatile markets by analyzing point clouds and persistence diagrams over different periods of time. These findings provide a new way of interpreting how prices behave under different conditions, going beyond basic trend or volatility metrics.

Throughout the project, the intended objectives were met. TDA was effective in identifying hidden patterns in the Bitcoin price, evaluating long-term stability, analyzing price anomalies and revealing the different stages of the market cycle. The results show that TDA not only describes price movements visually, but also reveals deeper behaviours in complex financial time series.

6.3 Limitations

There were several limitations encountered during the process, even though the project successfully demonstrated how topological data analysis can be applied to Bitcoin price data,

The first major limitation is that the time frame of the analysis is too short, as the study focuses on a three-month period. This narrow focus may not adequately capture long-term trends or larger market cycles.

Furthermore, the analysis may overlook smaller price fluctuations that take place throughout the day because it depends only on daily closing prices. More detailed results could have been obtained by a higher frequency of data, but this was not used due to time and resource constraints.

Besides, the research was exclusively centred on Bitcoin, which means the findings may not be relevant to other cryptocurrencies that exhibit varying levels of liquidity, adoption and volatility.

Lastly, selecting embedding parameters like embedding dimension (*d*) and time delay (*tau*) is necessary for the use of persistent homology. Since there is no an ideal way to choose these parameters, the analysis's overall generality is impacted. The selection of these parameters can have a big impact on the results.

6.4 Future Works

Future research could extend this project by applying TDA to other cryptocurrencies such as Ethereum, Solana or newer tokens with different levels of adoption and volatility. Comparing their topological behaviour to Bitcoin could reveal whether certain patterns are common to the cryptocurrency market as a whole or specific to individual assets. This comparison can also allows for a better understanding of how different cryptocurrencies respond to market conditions.

Another improvement is to experiment with different embedding parameters. These parameters affect how time series are converted into point clouds and have a significant impact on the shapes and features detected in persistence diagrams. A systematic approach to adjusting the values of these parameters would lead to a more consistent and accurate result.

More information could be obtained by longer or higher frequency datasets, such as hourly or minute level Bitcoin prices, particularly during times of rapid price movement. TDA would be able to identify deeper patterns that daily data might overlook with high-resolution data. This would be particularly helpful when examining market intraday volatility.

In addition, TDA has the potential to be combined with machine learning models that use topological features like loop persistence or number of components as meaningful input variables. These features can be used to train classifiers or regression models. This fusion of topological and statistical learning may improves the forecasting performance.

6.5 Conclusion

This project explored the use of topological data analysis to examine the structural patterns in Bitcoin price data. By applying Takens embedding and analyzing both point clouds and persistence diagrams, this study was able to uncover hidden behaviours across different market conditions.

The results showed that TDA is capable of identifying meaningful differences in price behaviour, especially in detecting recurring patterns and market complexity through persistent topological features. These findings suggest that TDA provides a valuable and unique perspective on financial analysis that goes beyond what traditional methods typically reveal.

Overall, this study shows that TDA is a useful tool for understanding complex time series data in cryptocurrencies market. By analyzing the geometric structure of price movements rather than just the numerical values over time, TDA provides a fresh perspective on how market behaviour evolves. This highlights the potential of topological methods as an important complement to time series analysis, especially in areas where traditional statistical models may fall short.

References

Ahmad Sabri, I. A., Man, M., Abu Bakar, W. A. W., & Mohd Rose, A. N. (2019). Web Data Extraction Approach for Deep Web using WEIDJ. *Procedia Computer Science*, *163*, 417–426. https://doi.org/10.1016/j.procs.2019.12.124

Ali Alahmari, S. (2020). PREDICTING THE PRICE OF CRYPTOCURRENCY USING SUPPORT VECTOR REGRESSION METHODS. *JOURNAL of MECHANICS of CONTINUA and MATHEMATICAL SCIENCES*, *15*(4). https://doi.org/10.26782/jmcms.2020.04.00023

Andi, H. K. (2021). An Accurate Bitcoin Price Prediction using logistic regression with LSTM Machine Learning model. *September 2021*, *3*(3), 205–217. https://doi.org/10.36548/jscp.2021.3.006

Azari, A. (2019). Bitcoin Price Prediction: An ARIMA Approach. *ArXiv (Cornell University)*. https://doi.org/10.48550/arxiv.1904.05315

Buis, M. L. (2017). Logistic regression: When can we do what we think we can do. *Unpublished Note*, *2*.

Catania, L., & Grassi, S. (2017). Modelling Crypto-Currencies Financial Time-Series. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.3028486

Chazal, F., & Michel, B. (2021). An Introduction to Topological Data Analysis: Fundamental and Practical Aspects for Data Scientists. *Frontiers in Artificial Intelligence*, *4*. https://doi.org/10.3389/frai.2021.667963

Chen, J. (2023). Analysis of Bitcoin Price Prediction Using Machine Learning. *Journal of Risk and Financial Management*, *16*(1), 51. https://doi.org/10.3390/jrfm16010051

Del, A. (2024, August 7). *Topological Data Analysis with Persistent Homology - Alexander Del Toro Barba (PhD) - Medium*. Medium. https://medium.com/@deltorobarba/quantum-topological-data-analysis-the-most-powerful-quantum-machine-learning-algorithm-part-1-c6d055f2a4de

Gidea, M., Katz, Y. A., Roldan, P., Goldsmith, D., & Shmalo, Y. (2018). Topological Recognition of Critical Transitions in Time Series of Cryptocurrencies. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.3202721

Harvey L. Sapigao. (2024, October 8). *UP Mathematicians Introduce a Different Approach in Detecting Potential Stock Market Crashes*. SCIENCE - University of the Philippines Diliman -. https://science.upd.edu.ph/up-mathematicians-introduce-a-different-approach-in-detecting-potential-stock-market-crashes/

Hyndman, R. J., & Athanasopoulos, G. (2018). *Forecasting : Principles and Practice* (2nd ed.). Otexts.

igmguru. (2025, June 25). *What is a Neural Network?* Igmguru.com; igmguru. https://www.igmguru.com/blog/what-is-neural-network

Jay, P., Kalariya, V., Parmar, P., Tanwar, S., Kumar, N., & Alazab, M. (2020). Stochastic Neural Networks for Cryptocurrency Price Prediction. *IEEE Access, 8*, 82804–82818. https://doi.org/10.1109/access.2020.2990659

Kenton, W. (2019). *GARCHP rocess* . Investopedia. https://www.investopedia.com/terms/g/generalalizedautogregressiveconditionalheteroskedasticity.asp

Knez, G. (2023, October 23). *25 Python Machine Learning Libraries*. Sunscrapers. https://sunscrapers.com/blog/python-machine-learning-libraries-for-data-science/

Mahdi, N., & Mohsin Abdulazeez, A. (2021). Machine Learning Classification Based on Radom Forest Algorithm: A Review. *International Journal of Science and Business*. https://doi.org/10.5281/zenodo.4471118

Manjula.B.C, Shilpa.B.S, & Sundaresh. M. (2022). Analysis of Cryptocurrency, Bitcoin and the Future. *East Asian Journal of Multidisciplinary Research*, *1*(7), 1293–1302. https://doi.org/10.55927/eajmr.v1i7.803

Mondal, P., Shit, L., & Goswami, S. (2014). Study of Effectiveness of Time Series Modeling (Arima) in Forecasting Stock Prices. *International Journal of Computer Science, Engineering and Applications*, *4*(2), 13–29. https://doi.org/10.5121/ijcsea.2014.4202

Naimy, V., Haddad, O., Fernández-Avilés, G., & El Khoury, R. (2021). The predictive capacity of GARCH-type models in measuring the volatility of crypto and world currencies. *PLOS ONE*, *16*(1), e0245904. https://doi.org/10.1371/journal.pone.0245904

Nalini Ravishanker, & Chen, R. (2019). Topological Data Analysis (TDA) for Time Series. *ArXiv (Cornell University)*. https://doi.org/10.48550/arxiv.1909.10604

Rodríguez-Pérez, R., & Bajorath, J. (2022). Evolution of Support Vector Machine and Regression Modeling in Chemoinformatics and Drug Discovery. *Journal of Computer-Aided Molecular Design*, *36*(5), 355–362. https://doi.org/10.1007/s10822-022-00442-9

Ruiz-Ortiz, M. A., Gómez-Larrañaga, José Carlos, & Rodríguez-Viorato, J. (2022). A persistent-homology-based turbulence index & some applications of TDA on financial markets. *ArXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2203.05603

Sabry, F., Labda, W., Erbad, A., & Malluhi, Q. (2020). Cryptocurrencies and Artificial Intelligence: Challenges and Opportunities. *IEEE Access, 8*, 175840–175858. https://doi.org/10.1109/access.2020.3025211

Tarud, J. (2021, October 11). *The Importance of Knowledge Discovery*. Koombea. https://www.koombea.com/blog/knowledge-discovery/

Wikipedia Contributors. (2019, April 17). *Autoregressive integrated moving average*. Wikipedia; Wikimedia Foundation. https://en.wikipedia.org/wiki/Autoregressive_integrated_moving_average

Zielak. (2025). *Bitcoin Historical Data*. Www.kaggle.com. https://www.kaggle.com/datasets/mczielinski/bitcoin-historical-data