UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE

Cryptocurrency Portfolio Optimization: A Hybrid Approach Combining Machine Learning and Mean-Variance Model

Presented by:
Chirine Dexposito
Hella Bouhadda
Charlotte Cegarra

Master 2 Modélisations, Statistiques, Economiques et Financières

Abstract

Portfolio optimization in the field of cryptocurrencies presents a major challenge due to the high volatility and unpredictability of these assets. This study proposes a hybrid approach combining machine learning models with financial optimization techniques to improve the prediction of returns and risk management in a portfolio of digital assets. Using a set of ten cryptocurrencies selected for their liquidity and market influence, we implemented a methodology integrating future return predictions via regression models and price trend classification through supervised learning algorithms.

The adopted approach is based on a multi-step process. First, an in-depth data analysis was conducted, including the creation of new variables and the extraction of relevant technical indicators. Next, cryptocurrency clustering was performed using the K-Means algorithm to optimize risk management and reduce the complexity of the optimization model. Model selection was based on rigorous comparisons using various evaluation metrics. The results demonstrate the superiority of the Random Forest (RF) optimized by the Particle Swarm Optimization (PSO) algorithm for return prediction, as well as the RF Classifier + PSO for market trend classification.

Integrating the predictions into a portfolio optimization framework based on the Mean-Variance Forecasting (MVF) model allowed dynamic adjustment of asset allocation, resulting in a more effective strategy in terms of Sharpe ratio and maximum drawdown. Backtesting of the strategies confirmed the robustness of the model, although some limitations related to reliance on historical data and exogenous shocks remain.

These results highlight the potential of machine learning methods to enhance portfolio management in the cryptocurrency sector. They pave the way for future research integrating more advanced models, including reinforcement learning and real-time capital flow analysis, to better capture market dynamics and refine investment strategies.

Keywords : cryptocurrencies, portfolio optimization, machine learning, CNN, PSO, clustering, Mean-Variance Forecasting, volatility, return prediction.

Contents

1 Introduction

1.1 Context

Portfolio optimization is a central issue in finance, aiming to allocate assets efficiently to maximize returns while minimizing risk. Since the foundational work of Markowitz (1952) on portfolio theory, allocation methods have evolved, incorporating concepts such as diversification, risk management, and the modeling of asset correlations. However, contemporary financial markets have become increasingly complex, making these traditional approaches often unsuitable. The increased volatility, asset interconnection, and growing influence of exogenous factors such as regulations, central bank decisions, and the impact of social media now require more sophisticated models capable of adapting to the constantly evolving market dynamics.

In this context, the cryptocurrency market presents a unique challenge. Unlike traditional asset classes such as stocks or bonds, cryptocurrencies are characterized by extreme volatility, non-normal return distributions, and unstable dynamic correlations. Price fluctuations can reach several tens of percent in a single day, and the market structure is heavily influenced by speculative behavior and network effects. Furthermore, the lack of strict regulation and the impact of political or technological decisions on the market make it difficult to apply traditional financial models. Sensitivity to macroeconomic news, institutional announcements, and social media trends further accentuates the uncertainty surrounding these assets.

Faced with these challenges, traditional portfolio optimization approaches, particularly those based on Markowitz's mean-variance model or the Capital Asset Pricing Model (CAPM), have several limitations. These models assume that returns follow a normal distribution and that asset correlations remain stable over time, which is highly debatable for cryptocurrencies. Additionally, these models do not account for the non-stationary nature of returns and the presence of extreme events, which are frequently observed in this market. Optimization based on a fixed covariance matrix can therefore lead to suboptimal decisions, particularly during periods of high turbulence.

Artificial intelligence and machine learning (ML) offer a promising alternative to overcome these limitations. Unlike traditional statistical models, ML algorithms can capture non-linear relationships and extract complex patterns from financial data. In particular, supervised learning models such as Random Forests, Convolutional Neural Networks (CNN), and boosting models (XGBoost, LightGBM) can improve return prediction and better anticipate market trends. These methods offer the ability to adapt to regime changes and can incorporate a wide range of indicators from technical analysis, on-chain data, and market trends.

In this study, we propose a hybrid approach combining machine learning and dynamic optimization to improve cryptocurrency portfolio management. Our goal is to integrate advanced machine learning techniques for predicting future returns, while using clustering methods to structure the investment universe and optimize asset allocation. The use of Particle Swarm Optimization (PSO) allows us to fine-tune model hyperparameters, ensuring robust performance even in the presence of volatile and noisy time series data. Finally, we leverage the Mean-Variance Forecasting (MVF) model to dynamically adjust portfolio weights based on predictions generated by the ML models.

The originality of this approach lies in the combination of three key elements: return prediction via a Random Forest model optimized by PSO, cryptocurrency clustering using K-Means to simplify the allocation space, and dynamic portfolio optimization by integrating these predictions into an MVF model. This methodology addresses the specific challenges of cryptocurrencies, accounting for their unstable nature while providing greater resilience to market fluctuations.

1.2 Problem Statement

Portfolio optimization in the context of cryptocurrencies presents unique challenges. Unlike traditional asset classes, cryptocurrencies exhibit extreme volatility, unstable correlations, and heightened sensitivity to external factors such as regulations or social media trends. Classical optimization models, such as Markowitz's model, are based on assumptions of normally distributed returns and constant correlations—conditions that are rarely encountered in cryptocurrency markets.

Therefore, the primary issue lies in the ability to develop optimization methods suited to this particular dynamic, taking into account the non-stationary nature of returns and the frequent occurrence of extreme events. Traditional asset allocation techniques become insufficient for anticipating fluctuations and the complexity of the cryptocurrency market.

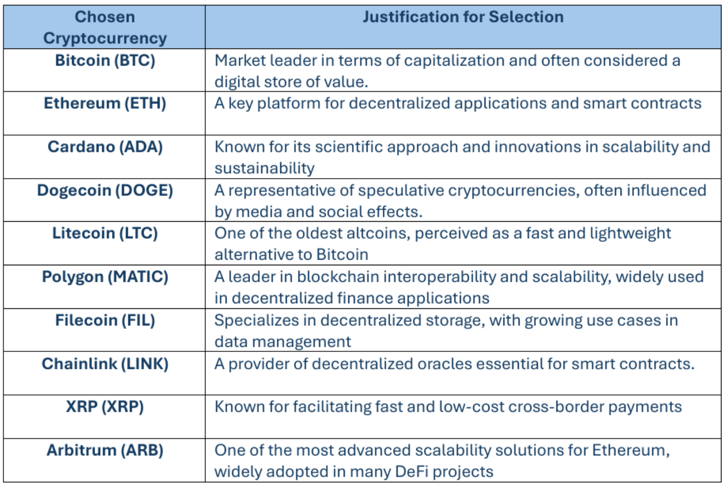
Artificial intelligence, particularly machine learning, could offer a solution by capturing the non-linear and dynamic relationships of returns. However, integrating these methods into a portfolio optimization framework remains challenging, especially when it comes to dynamically adjusting asset weights and better managing risks in such an uncertain environment. The central question of this study is, therefore, how to effectively combine machine learning, clustering, and dynamic optimization to maximize the returns of cryptocurrency portfolios while minimizing risks, in a market characterized by extreme volatility and unpredictability.

1.3 Project Objective

The main objective of this research is to develop a robust methodology for cryptocurrency portfolio management by leveraging advances in machine learning and heuristic optimization techniques. Specifically, we aim to build a model capable of predicting future cryptocurrency returns, efficiently structuring the investment universe by grouping assets with similar behaviors, and dynamically optimizing asset allocation within the portfolio. This methodological framework combines machine learning and multi-objective optimization to maximize risk-adjusted returns.

Our model will be applied to a sample of ten representative cryptocurrencies: Bitcoin (BTC), Ethereum (ETH), Cardano (ADA), Dogecoin (DOGE), Litecoin (LTC), Cosmos

(ATOM), Chainlink (LINK), Polygon (MATIC), XRP (XRP), and Filecoin (FIL).



Chosen Cryptocurrency	Justification for Selection
Bitcoin (BTC)	Market leader in terms of capitalization and often considered a digital store of value.
Ethereum (ETH)	A key platform for decentralized applications and smart contracts
Cardano (ADA)	Known for its scientific approach and innovations in scalability and sustainability
Dogecoin (DOGE)	A representative of speculative cryptocurrencies, often influenced by media and social effects.
Litecoin (LTC)	One of the oldest altcoins, perceived as a fast and lightweight alternative to Bitcoin
Polygon (MATIC)	A leader in blockchain interoperability and scalability, widely used in decentralized finance applications
Filecoin (FIL)	Specializes in decentralized storage, with growing use cases in data management
Chainlink (LINK)	A provider of decentralized oracles essential for smart contracts.
XRP (XRP)	Known for facilitating fast and low-cost cross-border payments
Arbitrum (ARB)	One of the most advanced scalability solutions for Ethereum, widely adopted in many DeFi projects

These assets were selected due to their diversity in terms of market capitalization, volatility, adoption, and specific use cases. They represent a combination of established cryptocurrencies (BTC, ETH, XRP) and innovative or emerging projects (ARB, FIL, MATIC), ensuring comprehensive coverage of different market dynamics.

To assess the robustness of our approach, we will use standard evaluation metrics for each component of the model. For predicting returns, we will evaluate performance using metrics such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and the R-squared (R^2) coefficient. For classifying market direction (price increases or decreases), we will use precision, recall, and accuracy measures. Finally, for portfolio optimization, we will evaluate performance based on the Sharpe ratio, maximum drawdown, and Value at Risk (VaR), which allows for estimating the robustness of the allocation under extreme market conditions.

The period chosen for this study spans from 2020 to 2024, a strategic interval to capture the evolution of the cryptocurrency market through different cyclical phases:

- Bull Market Phase (2020-2021): A period marked by the widespread adoption of cryptocurrencies, historical records in market capitalization, and the entry of institutional investors. This phase allows us to study market behaviors in a high-growth environment.

- Bear Market Phase (2022): Characterized by major events such as the collapse of Terra Luna, the FTX crash, and increased regulatory tightening. This phase highlights the effects of volatility and crises on portfolios.

- Recovery and Stabilization Phase (2023-2024): A gradual return of innovation (Web3, DeFi) and attempts to stabilize the markets after a consolidation phase.

This period offers a diverse sample of market conditions, allowing for the evaluation of the proposed model's robustness in varying configurations. Additionally, the study relies on daily data, an optimal compromise between granularity and data volume, ensuring precise analyses while avoiding biases related to intraday data noise.

1.4 Organization of the Report

This study is structured into several sections detailing the various stages of the methodology. Section two presents a literature review on portfolio optimization models and modern machine learning approaches in finance. Section three describes the methodology employed, including data collection and preparation, extraction of financial indicators, and the application of clustering techniques. Section four focuses on modeling, explaining the prediction algorithms used and their evaluation. Section five presents portfolio optimization and backtesting results, followed by an in-depth discussion in section six. Finally, section seven concludes the study and proposes avenues for future research

2 Literature Review

2.1 Portfolio Optimization in Quantitative Finance

Portfolio optimization is a central field in quantitative finance, aiming to maximize expected returns while minimizing the associated risk of a set of assets. The mean-variance (MV) model, introduced by Markowitz in 1952, remains the historical reference approach for achieving this balance. It is based on the analysis of average returns and asset covariances to construct an efficient frontier, defined as the set of portfolios offering the best risk-return tradeoff.

Mathematical Formulation of the Mean-Variance Model

Markowitz optimization can be formalized as the following problem :

$$\min_{w} \quad w^T \Sigma w$$

Subject to the constraints:

$$w^T \mu = R^*, \quad \sum_{i=1}^{n} w_i = 1, \quad w_i \geq 0$$

This model serves as a solid foundation for portfolio management and is widely used for traditional asset classes such as stocks and bonds. However, when applied to cryptocurrencies, some of the underlying assumptions of the MV model become problematic.

Limitations of the MV Model in the Context of Cryptocurrencies

Cryptocurrencies differ from traditional financial assets due to their extreme volatility, non-normal return distributions, and evolving market dynamics. These characteristics challenge several key assumptions of the MV model:

- **Non-Normal Return Distributions :** Unlike stocks or bonds, cryptocurrency returns often follow asymmetric and leptokurtic distributions with fat tails. This means that extreme events (sharp price increases or crashes) are underestimated by the MV model, which assumes a normal distribution.

- **Dynamic and Unstable Correlations :** The correlation structure among cryptocurrencies, as well as between cryptocurrencies and other asset classes, evolves rapidly under the influence of market trends, regulatory announcements, and technological advancements. The MV model relies on a static covariance matrix that fails to capture these variations, potentially leading to suboptimal portfolio allocations.

- **Sensitivity to Crises :** The MV model is particularly vulnerable to exogenous shocks and periods of high volatility, which can trigger extreme portfolio adjustments in response to sudden market fluctuations.

Despite these limitations, the mean-variance model remains a fundamental optimization tool. However, these challenges justify the exploration of complementary methods and adjustments to better adapt the model to the specific characteristics of cryptocurrencies.

Complementary Approaches to Enhance Portfolio Optimization

- **Risk Parity**
  Instead of minimizing the total portfolio variance, the risk parity method aims to balance each asset's contribution to overall portfolio risk. This approach is particularly useful for cryptocurrencies, where the volatility of certain assets (such as Bitcoin and Ethereum) can dominate the portfolio. By distributing risk more evenly, this method can enhance portfolio resilience to extreme fluctuations.

- **Sharpe Ratio Optimization**
  The Sharpe ratio measures excess return per unit of risk and is a widely used performance metric in portfolio management. Optimizing this ratio involves dynamically adjusting asset weights based on their expected returns and volatility. However, the reliability of this method depends heavily on the accuracy of return forecasts, which remains a challenge in the cryptocurrency market due to the nonlinear nature of price movements.

- **Integrating Machine Learning (ML) into Optimization**
  The emergence of machine learning models enables better capture of complex relationships among assets and the integration of advanced forecasts into portfolio optimization. Machine learning algorithms, such as neural networks, random forests, and clustering methods, can identify nonlinear patterns in price movements and improve asset allocations. These approaches are particularly well-suited for cryptocurrencies, where market trends are often influenced by exogenous factors such as regulatory announcements or sentiment analysis from social media.

The Rise of Hybrid Approaches

Rather than abandoning the MV model, a promising approach is to enhance it by integrating these alternative methodologies. Hybrid optimization, combining the MV model with machine learning-based return forecasts (Mean-Variance Forecasting, or MVF), introduces a dynamic dimension to portfolio construction. Additionally, heuristic optimization techniques, such as Particle Swarm Optimization (PSO), can be used to fine-tune these models and improve their robustness.

In the remainder of this study, we will focus on a hybrid approach combining Random Forest and MVF, leveraging recent advances in machine learning and heuristic optimization to enhance cryptocurrency portfolio management.

2.2 Return Prediction in Finance

Predicting financial returns is a crucial component of quantitative portfolio management, especially for volatile asset classes such as cryptocurrencies. Traditionally, statistical models like ARIMA and GARCH have been the standard tools for time series analysis and return forecasting. However, these approaches face significant limitations in capturing the nonlinear and complex dynamics that characterize cryptocurrencies. These limitations have led to an increasing adoption of machine learning (ML) models, which offer enhanced capabilities for extracting complex patterns and providing more accurate forecasts.

Limitations of Traditional Models

a) ARIMA (AutoRegressive Integrated Moving Average)

ARIMA is a widely used model for analyzing stationary time series and forecasting future trends. It is based on three main components:

- Auto-regression (AR): Uses linear dependencies between past observations
- Integration (I): Makes the series stationary by eliminating long-term trends
- Moving Average (MA): Models past errors to improve forecasts

Although ARIMA is effective for linear and stationary time series, it encounters major challenges when applied to cryptocurrencies. Cryptocurrency price series are often non-stationary, influenced by exogenous factors, and exhibit significant non-linearities, making ARIMA's underlying assumptions unsuitable.

b) GARCH (Generalized AutoRegressive Conditional Heteroskedasticity)

The GARCH model is designed to model and forecast the conditional volatility of financial assets. It is particularly effective in capturing periods of high volatility followed by calm periods, a common phenomenon in traditional financial markets. However, for cryptocurrencies, GARCH has several limitations:

- Inability to predict sudden exogenous shocks, such as regulatory announcements or cyberattacks.
- Dependence on historical dynamics, which does not account for behavioral or on-chain influences.

While these traditional models are useful in conventional markets, they fail to capture the complex and nonlinear patterns of cryptocurrencies, reducing their predictive effectiveness.

Advances in Machine Learning (ML) Models

Given the limitations of traditional approaches, machine learning models have emerged as powerful solutions for analyzing complex financial time series. These models capture the nonlinear dynamics of markets, leverage heterogeneous data, and generate more robust forecasts. Among the most effective architectures are Long Short-Term Memory (LSTM) recurrent neural networks and Convolutional Neural Networks (CNNs). Additionally, ensemble algorithms such as Random Forest, XGBoost, CatBoost, and LightGBM have demonstrated remarkable performance in financial return forecasting due to their robustness and ability to efficiently process large volumes of data.

LSTMs for Capturing Temporal Dependencies

LSTMs, an advanced class of recurrent neural networks, are designed to capture complex temporal relationships in long sequences of data. Unlike traditional RNNs, which suffer from the vanishing gradient problem, LSTMs incorporate memory cells that retain relevant information over long periods, mitigating information loss caused by deep network structures. This architecture is particularly suited to cryptocurrencies, where transaction volumes, active address variations, and on-chain events significantly influence prices. By combining these data with technical indicators, LSTMs enable more precise market movement anticipation, providing a suitable framework for predicting price trends in a volatile environment.

CNNs for Pattern Recognition in Time Series

CNNs, originally developed for image processing, have proven effective in financial time series analysis due to their ability to extract complex local patterns. Their main strength lies in the use of convolutional layers, which identify recurring patterns in data, such as bullish or bearish trends and anomalies in transaction volumes. Furthermore, pooling layers reduce data dimensionality while preserving essential information, improving model robustness and generalization. In the cryptocurrency domain, this approach allows for the integration of multiple data sources, including price variations, traded volumes, and technical indicators, to more effectively detect signals indicating trend reversals.

Ensemble Models for Robust and Interpretable Predictions

In addition to neural networks, ensemble models provide a robust and interpretable alternative for financial return forecasting. Among them:

a) Random Forest relies on an ensemble of decision trees trained on data subsamples using the bagging method. This approach enhances prediction robustness by aggregating multiple models, reducing overfitting risk, and exploiting nonlinear relationships between variables. In cryptocurrencies, Random Forest is particularly effective in identifying key factors influencing prices, such as historical volatility, trading volumes, and on-chain metrics.

b) XGBoost improves traditional boosting models by optimizing tree construction through efficient gradient minimization with regularization. It is fast to train while maintaining strong generalization capabilities. Its efficiency stems from its ability to handle heterogeneous and correlated variables and its robustness to missing values. In cryptocurrency portfolio optimization, XGBoost is often used to predict future returns based on a diverse set of variables, including on-chain data, social media trends, and macroeconomic factors.

c) CatBoost is another gradient boosting variant specifically optimized for categorical data processing. It enhances forecasting accuracy while reducing overfitting risks by effectively handling missing values and variable interactions. In cryptocurrency analysis, CatBoost is particularly relevant for leveraging qualitative data, such as market sentiment signals or token classifications based on fundamental characteristics.

LightGBM is optimized for execution speed and memory efficiency. It features a leaf-wise tree growth approach instead of level-wise, allowing for faster convergence and significantly reducing training time. This characteristic makes it particularly suitable for high-frequency time series, such as cryptocurrencies, where price fluctuations require rapid model adjustments.

Conclusion
Advancements in machine learning offer major benefits for portfolio management in volatile markets like cryptocurrencies. By capturing complex and nonlinear relationships between variables, these models overcome the limitations of traditional approaches. Unlike static models, they dynamically adapt to market variations by integrating multiple heterogeneous data sources. Moreover, their ability to process large datasets in real-time enables more effective anticipation of external shocks, whether regulatory announcements, macroeconomic news, or events within the crypto ecosystem.

Traditional portfolio optimization approaches show their limits when faced with the highly volatile and nonlinear nature of cryptocurrency returns. In response to these challenges, machine learning models provide more flexible and efficient alternatives by integrating complex dynamics and leveraging diverse data sources. Among these models, neural networks (LSTM, CNN) and ensemble algorithms (Random Forest, XGBoost, CatBoost, LightGBM) stand out for their ability to improve return prediction and asset allocation.

In this study, we adopt a **hybrid approach** combining **Random Forest for return prediction and Particle Swarm Optimization (PSO) for dynamic asset allocation optimization.** This methodology leverages the robustness and interpretability of ensemble models while benefiting from the efficiency of heuristic optimization techniques, providing a solution better suited to the unique characteristics of the cryptocurrency market.

2.3 Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) is a heuristic method inspired by the collective behavior of bird flocks and fish schools. Introduced by Kennedy and Eberhart in 1995, PSO has been widely adopted in various fields, including quantitative finance, due to its ability to efficiently optimize complex functions without relying on gradients or convexity. This flexibility makes it particularly suitable for financial problems, where models are often nonlinear and multidimensional.

PSO Principle

PSO is based on the idea of simulating a swarm of particles that explore a solution space. Each particle represents a candidate solution to the problem, and its position in space is adjusted based on :

- Its best position found so far (individual exploration).
- The best position found by the entire swarm (collective exploration).

The particles adjust their positions according to the following equations :

1) Particle velocity update :

$$v_i^{t+1} = w \cdot v_i^t + c_1 \cdot r_1 \cdot (p_i - x_i^t) + c_2 \cdot r_2 \cdot (g - x_i^t) \tag{1}$$

where:

- v_i^t: velocity of particle i at iteration t,
- x_i^t: current position of the particle,
- p_i: best position reached by particle i,
- g: best global position found by the swarm,
- w: inertia factor to maintain movement,
- c_1, c_2: weight coefficients for individual and collective exploration,
- r_1, r_2: random numbers introducing stochasticity.

2) Particle position update

The new position of each particle is determined as follows:

$$x_i^{t+1} = x_i^t + v_i^{t+1} \tag{2}$$

These iterative updates allow the swarm to converge toward an optimal solution while exploring a broad range of possible solutions.

Here is a diagram illustrating the **process of particle updates in the search space used in Particle Swarm Optimization (PSO).**

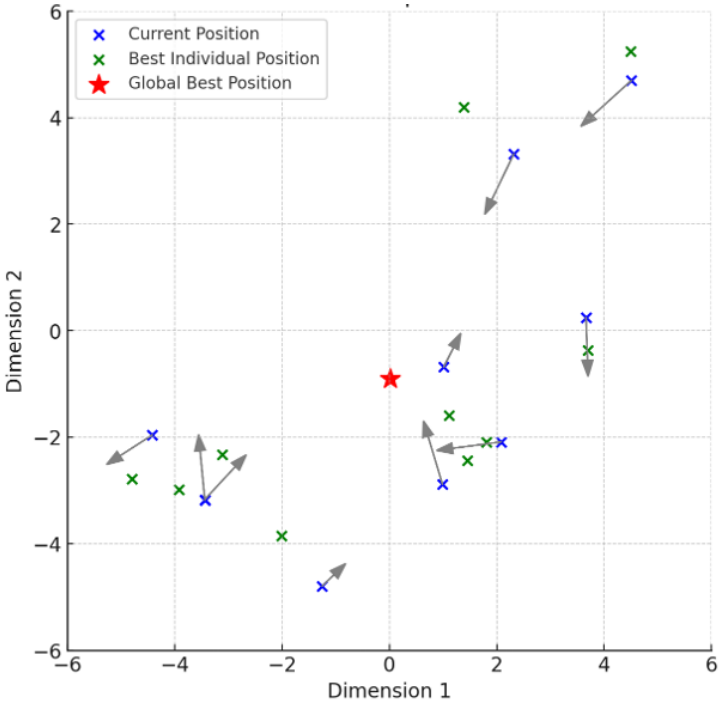


Figure 1: Illustration of Particle Update Process in PSO

Applications of PSO in Finance

PSO is particularly well-suited for financial problems requiring complex optimization, such as tuning hyperparameters of predictive models or portfolio optimization.

a) Optimization of Predictive Models

In machine learning, model performance often depends on the optimal tuning of hyperparameters, such as:

- The number of layers and neurons in a neural network.
- Learning and regularization rates.
- Time window sizes used for training.

PSO is employed to automatically search for the optimal configurations of these hyperparameters. Unlike grid search or random search approaches, PSO enables an intelligent exploration of the solution space by effectively balancing exploration and exploitation. For instance, in this study, PSO is integrated to optimize critical parameters of the Random Forest model, such as the number of filters, kernel sizes, and learning rates, to maximize the accuracy of cryptocurrency return predictions.

b) Portfolio Optimization

In portfolio management, the goal is to maximize expected returns while minimizing risk, often requiring complex optimization solutions. Traditional approaches, such as

Markowitz's mean-variance (MV) model, rely on restrictive assumptions, including the normality of returns and the stability of correlations. In contrast, PSO provides greater flexibility in handling scenarios where these assumptions do not hold, such as in cryptocurrency markets.

More specifically, PSO is particularly useful in portfolio optimization by dynamically adjusting asset weights based on return and risk forecasts. Through its iterative approach, it efficiently explores the search space to identify optimal allocations that maximize performance while minimizing exposure to market fluctuations.

One of PSO's major advantages is its ability to integrate dynamic constraints. Unlike traditional methods, it can account for factors such as liquidity limits, transaction costs, and asset weight restrictions. This flexibility is especially valuable in volatile markets like cryptocurrencies, where conditions evolve rapidly and require continuous allocation adjustments.

Finally, PSO excels in handling multi-objective optimization problems, making it a powerful tool for advanced investment strategies. For example, it can be used to maximize the Sharpe ratio while minimizing maximum drawdown, ensuring a balance between profitability and risk control. This approach enables a more robust portfolio management strategy by simultaneously considering multiple performance criteria.

2.4 Clustering in Finance

Clustering is an unsupervised data analysis method that aims to group assets into homogeneous clusters based on similar characteristics. In finance, this technique is particularly useful for identifying underlying relationships between assets that share common market dynamics, thus facilitating portfolio management and analysis. Among the available clustering algorithms, K-Means has emerged as one of the most popular and effective tools due to its conceptual simplicity and performance.

Principle of the K-Means Algorithm

The K-Means algorithm is based on the idea of partitioning a set of observations into KKK clusters while minimizing intra-cluster variance and maximizing separation between clusters.

The algorithm follows four key steps:

- **Initialization :** KKK centroids are randomly initialized within the data space.
- **Cluster Assignment :** Each observation is assigned to the cluster whose centroid is closest, based on a distance measure (typically Euclidean distance).
- **Centroid Recalculation :** Centroids are updated by computing the mean of observations assigned to each cluster.

- **Convergence :** Steps 2 and 3 are repeated until centroid positions stabilize or a predefined stopping criterion is met (e.g., a maximum number of iterations).

The objective function minimized by K-Means is given by:

La fonction J est définie comme suit :

$$J = \sum_{i=1}^{K} \sum_{x \in C_i} \|x - \mu_i\|^2$$

où C_i est le i-ème cluster, x est une observation, et μ_i est le centroïde du cluster C_i.

Applications of Clustering in Finance

In finance, clustering, and particularly K-Means, is used to simplify complex problems by grouping financial assets that exhibit similar behaviors. This approach is especially relevant for the cryptocurrency market, where asset diversity makes individual analysis both labor-intensive and computationally expensive.

- **Reducing Complexity**

Cryptocurrencies exhibit a wide range of characteristics, such as volatility, market capitalization, and returns. By applying K-Means, assets can be grouped into homogeneous clusters, reducing the number of necessary analyses and allowing predictive models to focus on representative assets from each cluster.

- **Portfolio Segmentation**

By grouping assets based on their market dynamics, it becomes possible to create diversified portfolios that represent different clusters. This improves risk management by limiting intra-cluster correlations and balancing exposure to various asset classes.

- **Anomaly Detection and Atypical Behaviors**

K-Means can also be used to identify anomalies in asset behavior. Cryptocurrencies that significantly deviate from the centroids of their respective clusters may indicate investment opportunities or specific risks.

Methodology Adopted in This Study

In this study, we apply the K-Means algorithm to cluster cryptocurrencies into homogeneous groups based on their financial and behavioral characteristics, such as volatility, market capitalization, and historical returns.

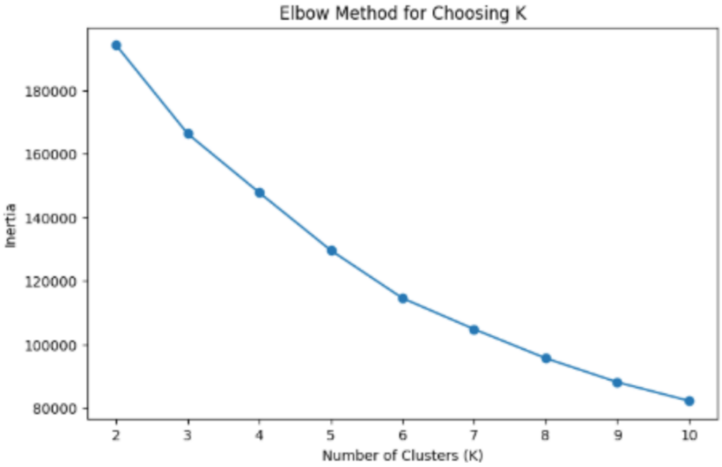
The following steps were undertaken to maximize clustering efficiency:

a) Data Preparation

Raw cryptocurrency data were normalized using a Z-score transformation to ensure that all variables are on a comparable scale.

b) Determining the Optimal Number of Clusters

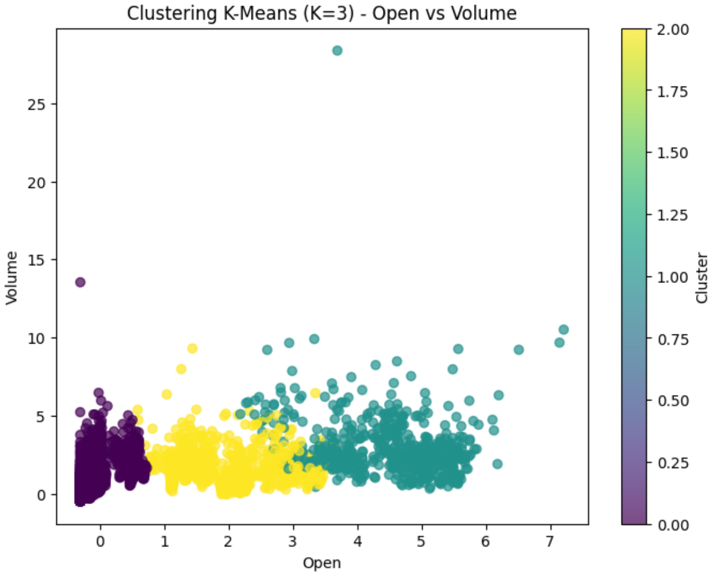
The **Elbow Method** was used to identify the inflection point in the intra-cluster inertia curve. The optimal number of clusters corresponds to the point where adding more clusters no longer significantly reduces inertia. A graphical representation of this method is provided below.



It is observed that the inertia decreases rapidly up to $K = 3$, then the decline slows down, forming an elbow at this point. This change in slope indicates that $K = 3$ is a good compromise between model accuracy and simplicity, as beyond this value, adding more clusters results in only a marginal improvement in inertia.

c) Evaluation of Cluster Quality

The silhouette score was calculated to measure the degree of intra-cluster cohesion and inter-cluster separation. High values indicate that the cryptocurrencies within a cluster are similar to each other and distinct from other groups. Below is a graphical visualization of this.



The interpretation shows a clear segmentation of observations into three distinct groups. The first cluster (purple) groups assets with low opening values and relatively low trading volumes. The second cluster (yellow) consists of assets with slightly higher opening values, but with more marked variability in volumes. Finally, the third cluster (teal) includes assets with higher opening values and greater dispersion in volumes. This classification can be useful for distinguishing different categories of assets based on their market behavior, thus facilitating portfolio optimization strategies tailored to specific risk profiles.

d) Selection of Cluster Representatives

A representative cryptocurrency was chosen for each cluster based on its proximity to the centroid of the group. This selection helps simplify the training of predictive models while ensuring good coverage of market dynamics.

Limitations and Potential Improvements

Although the K-Means algorithm is widely used for data classification and performs well, it suffers from several limitations that can affect the quality of the clusters obtained. One of the main weaknesses of K-Means lies in its **sensitivity to outliers**. Indeed, anomalies present in the data can significantly impact the calculation of centroids, leading to poor cluster distribution and a distortion of the representativeness of the formed groups. In a financial context, where extreme price or volume variations are frequent, this limitation can skew the analysis and lead to incorrect interpretations.

Another major issue with K-Means is the **assumption of spherical cluster shapes**. The algorithm implicitly assumes that the groups formed are convex and isotropic, meaning they have a relatively homogeneous structure in terms of distance from the centroids. However, in financial market analysis, the relationships between variables are often complex and non-linear, which can create irregularly shaped clusters that are poorly represented by K-Means. This restrictive assumption can limit the algorithm's ability to capture more sophisticated underlying structures in the data.

Finally, K-Means requires an arbitrary **choice of the number of clusters**, usually relying on heuristics like the Elbow Method or the silhouette index. This dependence on an a priori selection of the number of groups can be problematic when the actual data structure does not correspond to a clear separation into a fixed number of clusters. In finance, where market dynamics are constantly evolving, this constraint can reduce the model's flexibility and require frequent adjustments to adapt to new conditions.

This approach, combined with machine learning and portfolio optimization, helps improve the accuracy of predictions and the overall performance of investment strategies

3 Methodology

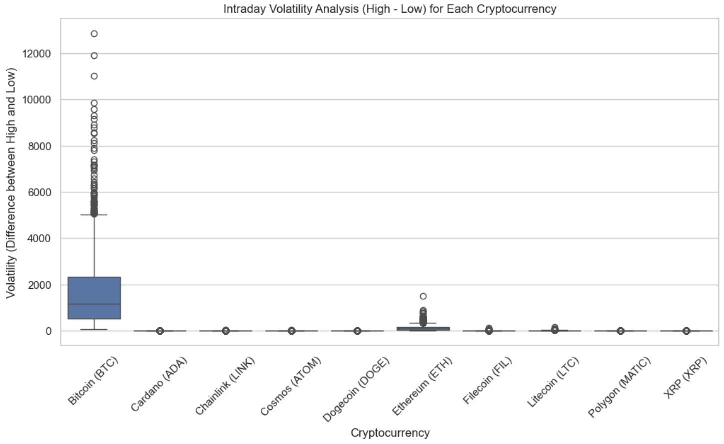
3.1 Data Sources

This study is based on a set of 10 cryptocurrencies selected according to several strategic criteria mentioned in the introduction. It is essential to choose assets that represent different market dynamics to ensure a robust analysis applicable to investors. The selected cryptocurrencies are **Bitcoin (BTC), Ethereum (ETH), Cardano (ADA), Dogecoin (DOGE), Litecoin (LTC), Cosmos (ATOM), Chainlink (LINK), Polygon (MATIC), XRP (XRP), and Filecoin (FIL).**

One of the fundamental aspects of cryptocurrencies is their **high volatility** compared to traditional asset classes. To illustrate this, we analyzed **intraday volatility**, defined as the difference between the highest and lowest price of the day:

$$\text{Intraday Volatility} = \text{High} - \text{Low}$$

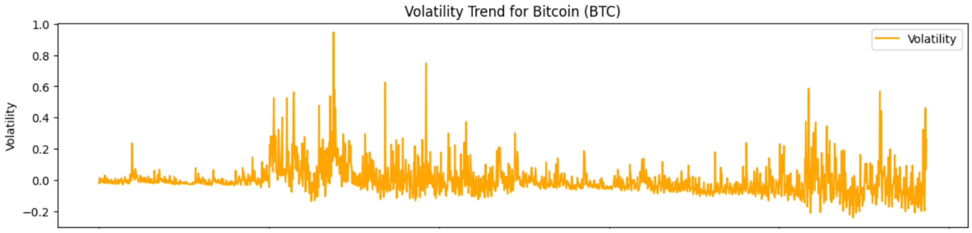
The following boxplot represents the distribution of this volatility for each selected cryptocurrency.



This chart highlights that Bitcoin (BTC) exhibits the highest intraday volatility, as evidenced by the width of the box and the presence of numerous outliers. This observation is consistent with its status as the dominant asset in the crypto market, often subject to massive fluctuations driven by macroeconomic news and institutional decisions.

Other assets, such as XRP and MATIC, display relatively lower volatility, which may be linked to greater stability in their usage and a reduced impact of exogenous factors.

To further this analysis, we examined the temporal evolution of Bitcoin's volatility, as it is one of the primary drivers of fluctuations in the cryptocurrency market.



The temporal analysis of Bitcoin's volatility reveals significant spikes at certain periods. These increases in volatility are generally correlated with major market events, such as regulatory announcements, central bank decisions, or liquidity crises on exchange platforms.

A fundamental aspect of portfolio optimization in cryptocurrencies is incorporating this volatility into the modeling and asset allocation process.

3.2 Data Collection and Preprocessing

The data preprocessing step is essential to ensure the quality and relevance of the analyses conducted. In this context, several transformations and treatments are applied to raw data to improve its usability and eliminate biases that could affect the results.

Feature Engineering

The first phase of preprocessing involves creating new variables derived from raw data. These new variables help extract additional insights into market dynamics and enhance the performance of optimization and forecasting models.
The variables created are as follows:

- **Return (R_t):** Represents the daily return of an asset, calculated as the logarithmic change between the current day's closing price and that of the previous day:

$$R_t = \ln\left(\frac{Close_t}{Close_{t-1}}\right)$$

- **Volatility (V_t):** Measures the dispersion of returns over a given period. It is calculated as the standard deviation of returns over a rolling window of n days:

$$V_t = \sqrt{\frac{1}{n}\sum_{i=t-n+1}^{t}(R_i - \overline{R})^2}$$

  where $\overline{R}$ is the mean return over the period.

- **Amplitude (A_t):** Defines the relative difference between the highest and lowest price of an asset for a given day, allowing an assessment of the magnitude of fluctuations:

$$A_t = \frac{High_t - Low_t}{Low_t}$$

- **Price Change (PC_t):** Indicates the absolute change in closing price between two consecutive days:
$$PC_t = Close_t - Close_{t-1}$$
- **Volatility Ratio (VR_t):** Evaluates relative volatility concerning the average price of the day to detect periods of high instability:
$$VR_t = \frac{V_t}{\frac{High_t + Low_t + Close_t}{3}}$$
- **High-Low Spread (HLS_t):** Measures the absolute value of daily fluctuations by subtracting the lowest price from the highest price:
$$HLS_t = High_t - Low_t$$

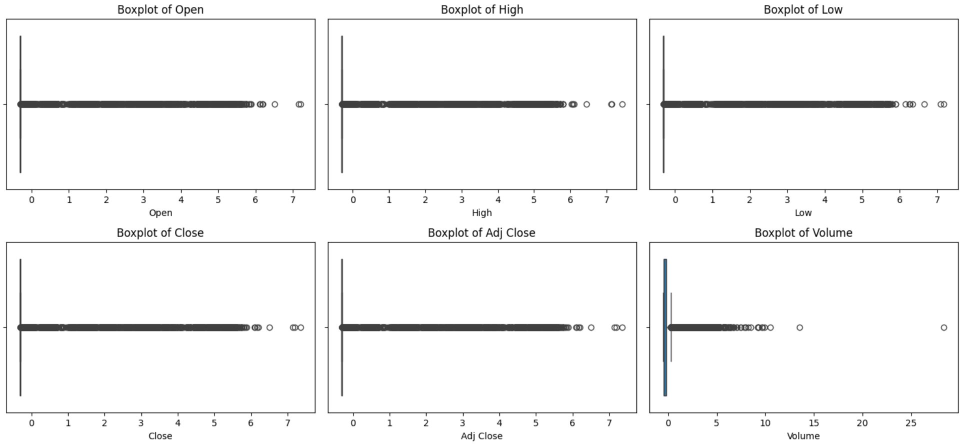
These variables are integrated into the study to improve asset behavior characterization and provide a more robust foundation for cluster analysis and portfolio optimization.

Normalization

Once these variables are generated, the data undergoes a normalization process to standardize the scales of different variables and improve the convergence of clustering and modeling algorithms. This step is particularly important when working with heterogeneous financial indicators, such as prices and trading volumes, which can exhibit significant differences in magnitude.

Outlier Detection

Finally, outlier detection is a critical phase of preprocessing. These values, which may result from market anomalies or data errors, are identified using statistical techniques such as boxplot analysis. The chart below illustrates the distribution of key variables in the study through a series of boxplots, allowing for the visualization of extreme values.



The analysis of these boxplots applied to the main financial variables of the dataset, including Open, High, Low, Close, Adjusted Close, and Volume, highlights a strong concentration of data around central values, with numerous points located outside the whiskers, indicating the presence of outliers. These extreme values are particularly noticeable in the **Volume** variable, where isolated observations suggest unusual spikes in the trading activity of certain assets.

Identifying these anomalies is crucial, as they can distort predictive models and negatively impact the performance of optimization algorithms. In the next stage of preprocessing, appropriate strategies such as **winsorizing** (reducing the impact of extreme values) or **removing the most atypical observations** can be implemented to mitigate their influence and ensure a more robust analysis.

3.3 Extraction of Technical Indicators

To enhance the quality of predictive models and refine the analysis of market dynamics, we have integrated a set of technical indicators. These indicators, commonly used in technical analysis, help better capture trends, volatility, momentum, and the evolution of asset trading volumes. They are grouped into four main categories, each providing complementary information on price and volume behavior.

3.3.1 Trend Indicators

Trend indicators help identify the overall market direction, whether bullish, bearish, or in a consolidation phase. They are particularly useful for smoothing short-term fluctuations and providing a clearer view of underlying trends.

Simple and Exponential Moving Averages (SMA and EMA)

Moving averages are fundamental tools in technical analysis. They help identify price direction by calculating a rolling average over a given period.

- **Simple Moving Average (SMA):** The Simple Moving Average is calculated as the average of the closing prices over a chosen period n:

$$SMA_n = \frac{1}{n} \sum_{i=0}^{n-1} Close_{t-i}$$

  where n is the chosen period (e.g., SMA_7, SMA_{50}).

- **Exponential Moving Average (EMA):** Unlike the SMA, the EMA gives more weight to recent data, making it more responsive to price changes:

$$EMA_t = \alpha \cdot Close_t + (1 - \alpha) \cdot EMA_{t-1}$$

  where $\alpha = \frac{2}{n+1}$ is the smoothing factor.

Rate of Change (RoC)

The Rate of Change is a momentum indicator that measures the percentage change in price over a given period.
It is used to detect accelerations and trend reversals:

$$RoC_t = \frac{Close_t - Close_{t-n}}{Close_{t-n}} \times 100$$

A high RoC indicates strong upward or downward pressure, while a RoC close to zero suggests market consolidation.

3.3.2 Volatility Indicators

Volatility indicators help assess the amplitude of price fluctuations and anticipate periods of high market uncertainty.

Bollinger Bands (BB)

Bollinger Bands measure volatility by plotting a channel around a moving average.
They are defined as follows:

$$BB_{high} = SMA_n + k \cdot \sigma$$

$$BB_{low} = SMA_n - k \cdot \sigma$$

where σ is the standard deviation of prices over n periods, and k is a multiplier, typically set to 2. Prices near the upper or lower bands indicate overbought or oversold conditions.

Average True Range (ATR)

The ATR measures an asset's volatility by considering the amplitude of daily fluctuations. It is defined as the moving average of the **True Range (TR)** over n days:

$$TR_t = \max(High_t - Low_t, |High_t - Close_{t-1}|, |Low_t - Close_{t-1}|)$$

$$ATR_t = \frac{1}{n}\sum_{i=0}^{n-1} TR_{t-i}$$

A high ATR indicates strong volatility, while a low ATR signals price stabilization.

3.3.3 Momentum Indicators

Momentum indicators help evaluate the speed and strength of price movements, making it possible to identify potential market reversal points.

Relative Strength Index (RSI)

The RSI is an oscillator that measures the strength of recent price movements over a given period. It is defined as:

$$RSI_t = 100 - \left(\frac{100}{1 + RS}\right)$$

where

$$RS = \frac{\text{average gains over } n \text{ days}}{\text{average losses over } n \text{ days}}$$

An RSI above 70 indicates an overbought condition, while an RSI below 30 signals an oversold condition. This indicator is particularly useful for identifying potential reversal zones.

3.3.4 Volume Indicators

Volume indicators help measure market activity and identify trends supported by significant trading volumes.

Volume Weighted Average Price (VWAP)

The VWAP is an indicator that measures the volume-weighted average price of transactions over a given period.
It is calculated as follows:

$$VWAP_t = \frac{\sum_{i=1}^{t}(Volume_i \times Close_i)}{\sum_{i=1}^{t} Volume_i}$$

The VWAP is often used to assess whether an asset is trading above or below its weighted average, helping to identify support and resistance zones.

On-Balance Volume (OBV)

The OBV is a cumulative indicator that measures buying and selling pressure by associating price changes with trading volumes:

$$\begin{aligned} & OBV_{t-1} + Volume_t, && \text{if } Close_t > Close_{t-1} \\ OBV_t = {} & OBV_{t-1} - Volume_t, && \text{if } Close_t < Close_{t-1} \\ & OBV_{t-1}, && \text{if } Close_t = Close_{t-1} \end{aligned}$$

An increasing OBV suggests that volume is supporting the price rise, which can serve as a trend confirmation signal.

3.3.5 Summary

The integration of these technical indicators enhances the analysis of financial assets and improves the performance of predictive models. Trend indicators provide insights into price direction, while volatility indicators help assess the magnitude of fluctuations.

Momentum indicators are useful for identifying market reversal points, and volume indicators help measure the underlying strength of price movements.

Thanks to this approach, the developed models benefit from a richer and better-structured dataset, facilitating the identification of market signals and optimizing investment decisions.

4 Modeling

4.1 Preparation for Modeling

In our study, we cover the period from 2020 to 2024, a key timeframe that includes various market phases, ranging from strong bullish trends to sharp corrections. This period encompasses major events that have significantly impacted cryptocurrency volatility, such as the 2021 bull run, the Terra Luna crash in 2022, and the collapse of FTX. The objective of this temporal analysis is to ensure that the models are tested across a broad spectrum of market conditions to evaluate their robustness.

Thus, two target variables have been defined to model market returns and trends :

a) Primary Target: Future Return

This quantitative variable is obtained from the logarithmic change in closing prices between two consecutive days, using the following formula:

$$R_t = \log\left(\frac{P_t}{P_{t-1}}\right)$$

where R_t represents the return on day t, and P_t and P_{t-1} are the closing prices on days t and $t-1$, respectively. This approach captures the relative price variations while mitigating the impact of different price scales across cryptocurrencies.

b) Secondary Target: Price Direction

This categorical variable is a binary version of the primary target and is defined as follows :

$$D_t = \begin{cases} 1, & \text{if } R_t > 0 \text{ (price increase)} \\ 0, & \text{if } R_t \leq 0 \text{ (price decrease)} \end{cases}$$

This transformation converts a regression problem into a binary classification problem, which is useful for predicting market trends.

Time Series Splitting for Model Training

To maintain the temporal integrity of financial time series and prevent data leakage, we used Python's Time Series Split method. Unlike a standard random train/test split, this technique ensures that models are trained on past data and tested on future observations.

This simulates a real-world decision-making scenario where only past information is available to predict the future. This approach prevents the model from benefiting from artificial temporal bias and better reflects real financial market decision-making conditions.

4.2 Model Evaluation

The evaluation of the models is based on metrics adapted to both regression and classification tasks:

For the **primary target (regression)** performance is measured using:

- **R^2 (Coefficient of Determination):** Indicates the proportion of variance explained by the model. An R^2 value close to 1 means that the model effectively captures the relationship between explanatory variables and future returns.
- **MSE (Mean Squared Error):** Measures the average squared error, heavily penalizing large errors.
- **RMSE (Root Mean Squared Error):** Expresses the error in the same unit as the returns, making it easier to interpret and compare.
- **MAE (Mean Absolute Error):** Less sensitive to outliers, it represents the average absolute error between predicted and actual values.

For the **secondary target (classification),** models are evaluated using:

- **Accuracy:** The proportion of correct predictions over the total observations.
- **Precision:** Measures the proportion of predicted positive cases that are actually positive. High precision is important to minimize false buy or sell signals.
- **Recall:** Indicates the proportion of actual positive cases correctly identified by the model. A high recall is essential to avoid missing important market opportunities.

These metrics enable a rigorous evaluation of the models according to their respective objectives.

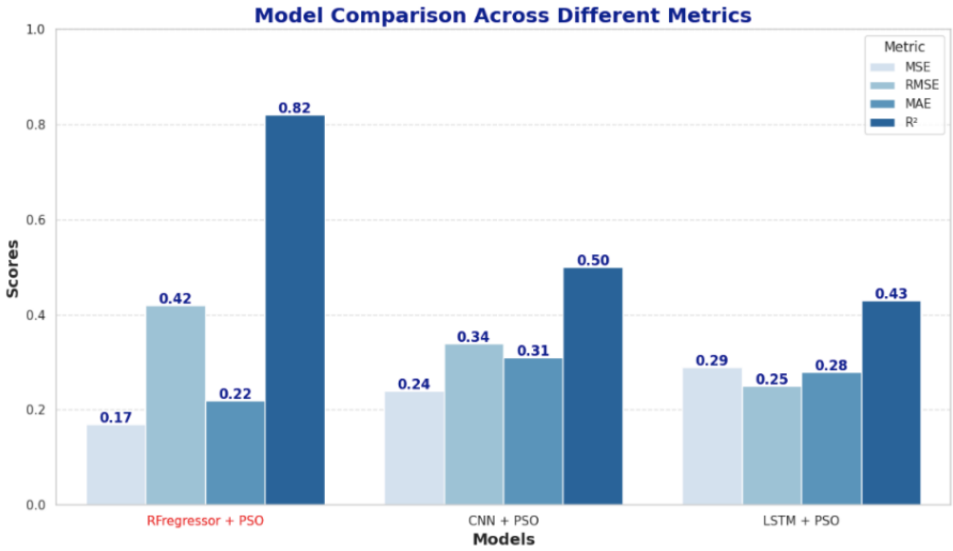
4.3 Model Selection and Performance

4.3.1 Modeling the Primary Target (Regression)

For the regression of future returns, three models were tested:

- **Random Forest (RF) + PSO**
- **Long Short-Term Memory (LSTM) + PSO**
- **Convolutional Neural Network (CNN) + PSO**

The following chart illustrates the performance comparison of these models in terms of R^2, MSE, RMSE, and MAE :



The graph highlights the superiority of the RFRegressor + PSO model, which achieves an R^2 of 0.82, explaining most of the variance in future returns. It also has the lowest MSE (0.17) and MAE (0.22), indicating better accuracy and reduced prediction errors. In contrast, CNN + PSO ($R^2 = 0.50$) and LSTM + PSO ($R^2 = 0.43$) show weaker performance, with higher MSE and MAE values, suggesting greater prediction errors and overfitting. These results confirm that RFRegressor + PSO is better suited for cryptocurrency return prediction, while deep learning models struggle with generalization. **Based on these findings, we will proceed with RFRegressor + PSO for further analysis and implementation.**

Feature Selection and Optimization

To enhance optimization, a feature selection process was conducted to retain only the most informative variables for prediction. The selected features are:
features = {Open, Volume, Volatility, Amplitude, Price Change, Volatility Ratio, High-Low Spread, RoC, ATR, RSI, VWAP, OBV}

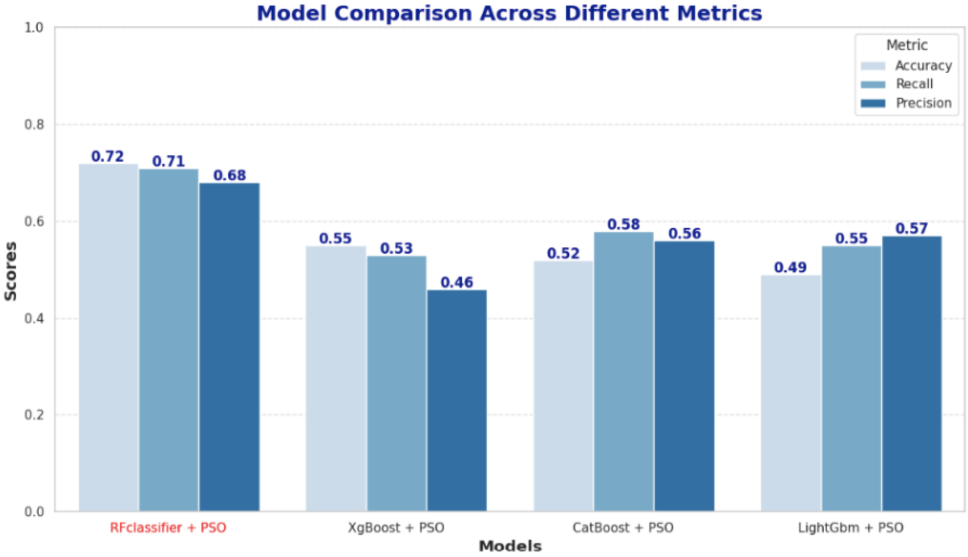
Eliminating redundant variables helps **improve model generalization and reduce computational complexity.**

4.3.2 Modeling the Secondary Target (Classification)

For the classification of price direction, several models were evaluated:

- **RF Classifier + PSO**
- **XGBoost + PSO**
- **CatBoost + PSO**
- **LightGBM + PSO**

The following chart illustrates the comparison of scores in terms of accuracy, recall, and precision:



The **RFClassifier + PSO** model outperforms all other models with the highest accuracy (0.72), recall (0.71), and precision (0.68), indicating better overall classification performance. XGBoost + PSO, CatBoost + PSO, and LightGBM + PSO show weaker results, particularly in precision, suggesting more false positives. These results confirm that RFClassifier + PSO is the most reliable model for market trend prediction. Based on these findings, we will proceed with **RFClassifier + PSO** for further analysis and implementation.

4.4 Portfolio Optimization and Backtesting

4.4.1 Optimization via the Mean-Variance Frontier (MVF) Model

The **Markowitz approach**, also known as the **Mean-Variance Frontier (MVF)**, is based on optimizing portfolios according to expected returns and the associated risk of assets. The **MVF model** aims to maximize the **Sharpe ratio**, which measures excess return per unit of risk taken, defined as follows:

$$\text{Sharpe Ratio} = \frac{E[R_p] - R_f}{\sigma_p}$$

where:

- $E[R_p]$ is the expected return of the portfolio,
- R_f is the risk-free rate,

- σ_p is the standard deviation of the portfolio return (measuring risk).

In our approach, the returns predicted by the machine learning model are integrated into the optimization process to make portfolio management more dynamic. Unlike a static approach based solely on historical data, this methodology allows real-time adjustments of asset weightings based on anticipated market trends.

The **optimization problem** is formulated as follows:

$$\max_{w} \quad \frac{w^T \hat{\mu} - R_f}{\sqrt{w^T \Sigma w}}$$

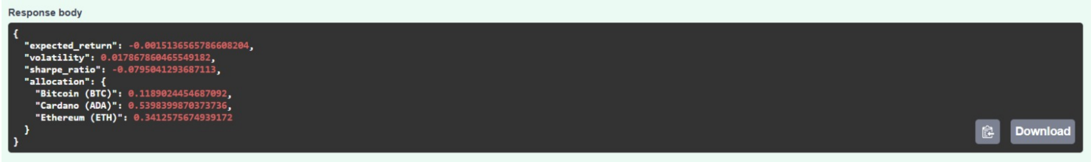
subject to:

$$\sum_{i=1}^{n} w_i = 1, \quad w_i \geq 0$$

where:

- w is the vector of asset weightings,
- $\hat{\mu}$ is the vector of predicted returns,
- Σ is the covariance matrix of assets,
- R_f is the risk-free rate.

An example of the optimal asset allocation obtained through this optimization is presented below:



```
{
  "expected_return": -0.0015136565786608204,
  "volatility": 0.017867860465549182,
  "sharpe_ratio": -0.0795041293687113,
  "allocation": {
    "Bitcoin (BTC)": 0.1189024454687092,
    "Cardano (ADA)": 0.5398399870373736,
    "Ethereum (ETH)": 0.3412575674939172
  }
}
```

Advantages of This Approach

This MVF-based strategy considers the correlation between assets to minimize portfolio volatility while maximizing expected returns. Additionally, it adapts to changing market environments by leveraging machine learning predictions to dynamically rebalance asset allocations based on anticipated trends.

4.4.2 Market Direction Prediction

In addition to optimizing asset weightings, a specific model has been developed to predict the overall market trend over a given period. The goal is to enhance decision-making by classifying the market as either bullish ("Up") or bearish ("Down"), allowing investment strategies to be adjusted accordingly. This model is based on a combination of historical prices and technical indicators integrated into a supervised machine learning architecture. More specifically, it leverages data such as:

- **Moving averages** to capture the underlying trend,
- **RSI and Rate of Change (RoC)** to identify price momentum,
- **Bollinger Bands and ATR** to assess volatility,
- **OBV and VWAP** to account for volume impact on market direction.

The model is trained using a rolling time window, where each observation is labeled based on future price variation: **Label:**

$$Label_t = \begin{cases} 1, & \text{if } Close_{t+h} > Close_t \quad \text{(Bullish market)} \\ 0, & \text{if } Close_{t+h} \leq Close_t \quad \text{(Bearish market)} \end{cases}$$

where h represents the forecast horizon (e.g., 1 day, 1 week).

The algorithm then evaluates the dominant trend over a given period by aggregating successive predictions. If the majority of forecasts indicate a price increase, the model output is "Up", otherwise, it is "Down". This approach helps filter out noisy signals and provides a consolidated view of market trends, thereby improving the robustness of investment decisions.

4.4.3 Market Direction Prediction (Consolidated Approach)

A dedicated model has been developed to predict the overall market trend over a given period. It relies on a combination of past prices and technical indicators to classify the future market direction as either bullish or bearish.

This model works by evaluating the majority trend of predictions over a defined time window. If most predictions indicate an uptrend, the output will be "Up", otherwise, it will be "Down". This approach provides a consolidated market view, making decision-making easier for investors.

The overall results confirm the relevance of our methodology and highlight the value of machine learning models in optimizing portfolios within the highly volatile cryptocurrency market environment.

5 Conclusion and Discussion

The analysis of regression model performance applied to future returns showed that **Random Forest (RF) optimized with PSO** outperforms other approaches, particularly **Convolutional Neural Networks (CNN) and LSTM architectures**. This result can be explained by Random Forest's ability to capture complex interactions between variables without requiring an excessively large dataset. Unlike neural models, **Random Forest is less prone to overfitting**, a frequent issue in financial time series forecasting where price fluctuations are influenced by unpredictable exogenous events. However, although CNN and LSTM produced more modest results, their relevance in contexts requiring **long temporal sequences** should not be overlooked. Their effectiveness could be improved by **increasing the training dataset size** or by integrating hybrid architectures combining these models with **probabilistic approaches or reinforcement learning**.

Regarding price direction classification, the RF Classifier + PSO model proved to be the most effective solution. This is particularly interesting because it enables the use of a robust model to identify market trends with a relatively high level of reliability. However, it should be noted that recall is slightly higher than precision, meaning the model tends to capture a larger number of upward trends, sometimes at the expense of false positives.

The integration of K-Means clustering in the methodology played a key role in reducing the computational complexity of the portfolio optimization problem. This approach allowed us to identify groups of cryptocurrencies with similar behaviors, thereby optimizing asset diversification and mitigating risks associated with strong correlations between certain cryptos. Cluster analysis highlighted three main groups characterized by distinct volatility levels and price trends. This segmentation facilitated resource allocation by prioritizing complementary assets, enhancing the robustness of the optimized portfolio using the MVF model.

The adopted approach presents several advantages that strengthen the credibility of the obtained results. The consideration of the temporal nature of the data via the Time Series Split method prevents information leakage between training and testing. The hyperparameter optimization via PSO ensures better model performance, while the use of multiple evaluation metrics enables a rigorous comparison of model performance.

Limitations and Future Research Directions

However, several limitations should be highlighted. The first limitation concerns the dependence on historical data, as the models remain highly influenced by past trends. Cryptocurrencies, however, are particularly sensitive to exogenous events, such as regulations, cyberattacks, or macroeconomic announcements, which are not captured by traditional statistical models. Another limitation lies in the absence of fundamental variables, as the study primarily relied on technical and volume indicators without incorporating elements such as institutional adoption, regulatory developments, or capital flows. Additionally, market noise represents a challenge, as cryptocurrencies are characterized by high volatility and unpredictability, making it difficult to model their fluctuations accurately. Although optimizing the models helps reduce this bias, a degree of unpredictability remains. Finally, the backtesting process is limited to historical