

**NOVA**

**IMS**

Information  
Management  
School

# MGI

Master Degree Program in  
**Information Management**

## **SELF-ORGANIZING MAPS TRADING STRATEGY**

BRAZILIAN STOCK MARKET

Rodrigo Arend Vaz Ferreira

Master Thesis

presented as partial requirement for obtaining a Master's Degree in Information Management

**NOVA Information Management School**  
**Instituto Superior de Estatística e Gestão de Informação**

Universidade Nova de Lisboa



**NOVA Information Management School**  
**Instituto Superior de Estatística e Gestão de Informação**  
Universidade Nova de Lisboa

**SELF-ORGANIZING MAPS TRADING STRATEGY**

Brazilian Stock Market

by

Rodrigo Arend Vaz Ferreira

Master thesis presented as partial requirement for obtaining the Master's degree in Information Management, with a specialization in Business Intelligence

**Supervised by**

Fernando Bação, PhD, NOVA Information Management School

April, 2025

## **STATEMENT OF INTEGRITY**

I hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism, any form of undue use of information or falsification of results along the process leading to its elaboration. I further declare that I have fully acknowledged the Rules of Conduct and Code of Honor from the NOVA Information Management School.

*Lisbon, 29/04/2025*

*Rodrigo Arend Vaz Ferreira*

## **DEDICATION**

To my wife Daria, whose unwavering love, patience, and encouragement have been my greatest source of strength throughout this journey. Thank you for supporting and believing in me.

To my father, mother, and brother, thank you for your support, sacrifices, and for always being there.

This work is for you.

## **ACKNOWLEDGEMENTS**

I would like to express my deepest gratitude to all those who supported me throughout the development of this thesis.

To my wife, Daria, for her unwavering love, patience, and encouragement, thank you for being my anchor through every step of this journey.

To my family, your constant support and belief in me have been invaluable.

To my friends and colleagues, thank you for the insightful discussions, shared challenges, and moments of motivation that helped me move forward.

To my professors and academic advisors at NOVA IMS, your guidance, expertise, and dedication to teaching have been instrumental in shaping this work.

To the staff and everyone involved in the administrative and academic support at the University, thank you for creating an environment where learning and growth are truly possible.

Finally, to all who contributed anonymously building the tools that made the computational part of this work possible.

## **ABSTRACT**

This thesis presents a novel trading strategy for the Brazilian stock market, leveraging Self-Organizing Maps (SOMs) to uncover hidden patterns in the daily behavior of the Ibovespa index from 2000 to 2023 and tested in data from 2024 to March 2025. By transforming raw price and technical indicators data into a set of Perceptually Important Points (PIPs), the model captures essential market movements while discarding less relevant fluctuations. A comprehensive time-series cross-validation approach is employed, ensuring rigorous performance evaluation on unseen data. The proposed method automatically clusters similar market conditions and assigns trading signals based on the expected future return within each cluster. Experimental results show that this SOM-based strategy outperforms both a simple buy-and-hold benchmark and other widely used technical analysis approaches in terms of overall returns and risk-adjusted performance. Notably, the inclusion of technical indicators provides a richer description of market trends, volatility, and momentum, enhancing the SOM's ability to distinguish profitable patterns. The framework also incorporates risk management by evaluating the Sharpe ratio and other metrics, highlighting the robustness of the strategy across diverse market regimes. These findings suggest that a well-tuned unsupervised learning approach, combined with carefully selected financial features, can systematically exploit market inefficiencies. Moreover, the methodology is flexible and can be extended to other stock markets or asset classes with minimal adjustments, underlining the versatility and practical value of the presented solution.

## **KEYWORDS**

Self-Organizing Maps; Trading Strategies; Brazilian Stock Market; Technical Analysis; Perceptually Important Points

# TABLE OF CONTENTS

Statement of Integrity.....	1
Dedication .....	2
Acknowledgements.....	3
Abstract .....	4
1. Introduction.....	9
2. Literature Review .....	11
2.1. Self-Organizing Maps (SOMs).....	11
2.2. Technical Analysis for Trading Assets.....	13
2.3. State of the Art .....	14
2.4. Literature Gap.....	17
3. Methodology .....	19
3.1. Data Acquisition .....	19
3.2. Data Preprocessing.....	20
3.3. Technical Indicators.....	21
3.4. Perceptually Important Points (PIPs) .....	22
3.5. Model Validation .....	24
3.6. Backtesting And SOM Evaluation .....	27
3.7. Model Comparison .....	28
3.8. Empirical Study.....	29
4. Results and Discussion.....	31
4.1. Best Overall Configuration in Backtesting.....	31
4.2. Comparison With Other Trading Strategies .....	35
5. Conclusions and Future Research .....	38
5.1. Key Findings.....	38
5.2. Theoretical and Practical Implications .....	38
5.3. Future Studies.....	38
Bibliographical References.....	40

## LIST OF FIGURES

Figure 1 – SOM Training Process. Adapted from Shirani Faradonbeh et al. (2020) .....	12
Figure 2 – Graphical Abstract .....	19
Figure 3 - PIP iteration process. Adapted from Fu et al. (2001).....	23
Figure 4 – Cross-validation where Validation Data is between two other Train Sets .....	26
Figure 5 – Cross-validation where Validation data is at the beginning .....	27
Figure 6 – Cross-validation method where Validation data is at the end .....	27
Figure 7 – Strategies Comparison .....	36
Figure 8 – SOM Model vs RSI and Buy and Hold with trade signals .....	37

## LIST OF TABLES

Table 1 – State of the art papers.....	14
Table 2 – Technical indicators.....	21
Table 3 – Metrics Calculated for each Cluster .....	27
Table 4 – Comparing Trading Strategies .....	28
Table 5 – Changing settings on grid search.....	29
Table 6 – Immutable setting on grid search.....	30
Table 7 – Grid search result table sample.....	31
Table 8 – Top 20 configurations.....	32
Table 9 – Parameters of the top 20 best configurations .....	33
Table 10 – Results by year using best overall configuration.....	34
Table 11 – Strategies Comparison Metrics .....	36

## LIST OF EQUATIONS

Equation 1 – Tanh Estimator .....	20
Equation 2 – Determine Candidate Point Significance .....	24
Equation 3 – PIPs Extraction.....	24

# 1. INTRODUCTION

Stock markets have long fascinated investors and traders due to their potential for generating substantial returns. This allure has driven market participants to develop various strategies in an attempt to decode market behavior and capitalize on price movements. Broadly, those who seek profit in stock markets can be classified into two primary groups: fundamental investors and technical traders.

Fundamental investors base their decisions on financial statements, economic data, industry analysis, and company reports. They operate under the belief that the true value of a security (its intrinsic value) often deviates from its market price. Over time, they expect market prices to adjust to reflect intrinsic value, creating opportunities for profit.

In contrast, technical traders rely on historical price movements and patterns, assuming that past trends tend to repeat. They use technical indicators, which are quantitative tools primarily derived from past price and volume data and used to identify potential trading opportunities. By analyzing market trends, momentum, and other patterns, technical traders seek to uncover hidden signals that may lead to profitable trades (Petrusheva & Jordanoski, 2016).

Although these groups tend to differ in their approaches and opinions, both are equally aiming for the same goal, profit in a complex and dynamic market. Moreover, both these approaches defy the Efficient Market Hypothesis (EMH) and the Random Walk Hypothesis (RWH), the first being introduced by Eugene F. Fama in the 60's, asserts that all financial markets' available information is quickly and accurately reflected in asset prices due to its fully rational numerous participants, leaving no room for consistent excess returns through market prediction. Complementary to this, the RWH, popularized by works such as Paul H. Cootner's and Fama's own studies, posits that stock prices follow a stochastic path, rendering future price movements independent of past behavior, meaning they are random and therefore unpredictable (Malkiel, 2003).

While EMH and RWH have empirical support Malkiel (2003), anomalies, behavioral biases, and inefficiencies observed in markets reveal their limitations, leaving room for technical and fundamental investing approaches to, even if to a limited extent, predict the market (Hoover, 2015; Leković, 2018; Lo & MacKinlay, 1988). Moreover, a comprehensive review and classification of 150 studies, challenges the strong form of the efficient market hypothesis (Atsalakis & Valavanis, 2010), pointing that market inefficiencies exist and can be successfully exploit by conventional models and trading strategies for excess returns.

Although traditional analytical approaches such as moving averages, trading range breakouts, momentum-based strategies and strategies based on indicators like Relative Strength Index (RSI) and Stochastic Oscillator Indicator (SOI) can generate excess returns as shown in Brock et al. (1992), Chiang et al. (2012) and Luukka et al. (2016), they often struggle to capture the

nuanced relationships and hidden patterns in vast, high-dimensional financial data. To address this challenge, machine learning techniques have emerged and popularized as powerful tools to analyze data and optimize trading strategies, as seen in the majority of the 37 publications on deep learning techniques for stock price prediction reviewed by Li and Bastos (2020) and most of the over 100 articles on neural networks and other techniques reviewed by Atsalakis & Valavanis (2009) both casting more doubt on the validity of EMH and RWH.

Among these sophisticated techniques, Self-Organizing Maps (SOMs) stand out for their ability to visualize and interpret complex datasets in an intuitive and meaningful manner, but more importantly, they are capable of learning complex relationships and patterns from data without the need for human intervention nor the need for labeled data. Developed by Kohonen (1982), Self-Organizing Maps are a type of unsupervised neural network designed to reduce the dimensionality of data while preserving its topological properties. SOMs project high-dimensional data onto a two-dimensional grid, which in a stock market perspective can enable traders to cluster data points that may inform trading decisions.

This paper explores the potential of Self-Organizing Maps (SOMs) in building analytical trading model that can outperform buy and hold strategies and other technical trading strategies. By leveraging SOMs, traders can take their hands off the wheel and let the machine decide for them either it is time to buy, sell or stay out of the market.

This study is heavily based on the technical trading ideal, but instead of having a human trying to identify patterns and hidden information in the price action and technical indicators of equities, it explores the potential of Self-Organizing Maps (SOMs) in building analytical trading model that can outperform the buy and hold strategy and other technical trading strategies, the first often treated as the passive benchmark, that simply involves purchasing the asset at the start of the investment horizon and keeping it untouched until the horizon ends, thereby minimizing turnover and transaction costs (Cohen & Cabiri, 2015), and the latest composed of following simple rules around one or more technical indicators. The desired result for a strategy based on SOM is to allow traders to take their hands off the wheel and let the machine decide for them whether it is time to buy, sell or stay out of the market.

## 2. LITERATURE REVIEW

The following review examines the literature on Self-Organizing Maps (SOMs) and their applications in financial markets, the key concepts of technical analysis, and the integration of SOMs with technical indicators for trading strategies. This review establishes a foundation for understanding SOMs usage in stock market analysis while identifying gaps in existing research that this paper seeks to address.

### 2.1. SELF-ORGANIZING MAPS (SOMs)

A Self-Organizing Map (SOM) is an unsupervised neural network technique developed by Kohonen (1982) for mapping high-dimensional data into a typically two-dimensional grid, a map, while preserving the original data's topological structure. It takes data with many features (dimensions) and produces a flat map, often 2D, where points that were close or similar in the original space end up near each other on the map. The main intuition is that similar data should activate nearby neurons on the map. This makes patterns in complex data easier to visualize and interpret, since distance on the map reflects similarity in the original space. An SOM can be thought of as a form of dimensionality reduction or clustering tool that preserves relationships: data points in proximal map locations have more similar features than those far apart on the map.

A SOM consists of a grid of neurons, each associated with a weight vector of the same dimensionality as the input data. During the training process, the neurons compete for the responsibility of representing the input data. The neuron whose weight vector is closest to the given input is identified as the "winner". Both the winner and its neighboring neurons are updated in a way that brings their weight vectors closer to the input data. This iterative process allows the map to self-organize by grouping similar data points together, producing a two-dimensional representation that reflects the topology of the original high-dimensional input space (Kohonen, 1982).

The initial weight vectors are often randomly generated to give the map a starting configuration spread across the data space. An iteration process then takes place for each data point for several iterations where the distance between each neuron and the data point is computed. The neuron whose weight is closest to the input data is the "winner" for that input and called the Best Matching Unit (BMU). Once the BMU is identified, the SOM "learns" by updating the BMU's weight vector to become more like the input vector. In other words, the BMU's weight moves a small step toward the data point for which it was elected the winner. But uniquely, neighboring neurons in the grid are also updated toward the input, but by a smaller amount. Neurons farther away on the grid get a smaller adjustment or none at all, depending on the configurations. The amount of adjustment typically follows a neighborhood function (often a Gaussian centered on the BMU) which decays with distance on the grid, meaning that neurons closer to the BMU have a more drastic adjustment in comparison to those more far spread. As the learning process goes on, the learning rate and

effective neighborhood radius decays (Kohonen, 2013). Displays a graphical representation of the SOM training process.

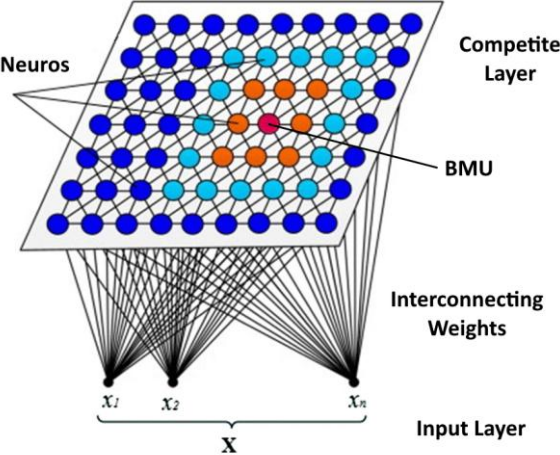


Figure 1 – SOM Training Process. Adapted from Shirani Faradonbeh et al. (2020)

Although not used in this paper for reasons discussed in later chapters, two of the most common metrics for assessing how well SOM maps the input data are the quantization error and the topological error. The first measures how accurately the SOM neurons represent the original data by reflecting the average distance between each input data point and its BMU. The latest measures how well the SOM preserves the topology (spatial relationships) of the original data. Specifically, it calculates the proportion of data points for which the two closest neurons are not neighbors on the SOM grid (Uriarte & Martín, 2008).

One of the defining features of SOMs is their unsupervised nature, which means they do not require predefined outputs or prior assumptions about data distribution. This makes SOMs particularly effective for discovering patterns and latent structures in complex datasets, even in cases where the relationship between variables are non-linear or poorly understood (Deboeck & Kohonen, 1998).

Because these qualities make SOMs well-suited to settings where hidden relationships drive decision-making, SOMs have been extensively applied in finance, a field rich in complex, high-dimensional, and often non-linear data. Deboeck and Kohonen (1998) presented a collection of financial applications of SOMs, such as statements to support corporate strategy formulation and solvency diagnostics, the establishment of bond ratings, the prediction of long-term interest rates, the selection of investments in mutual funds and stocks as being employed in the appraisal of real estate properties and the segmentation of customers.

Self-Organizing Maps (SOMs) offer distinct advantages over other machine learning approaches in financial market predictions, especially due to their ability to visualize and cluster complex data intuitively. But a characteristic that is more relevant for this study is that unlike supervised learning algorithms, such as Support Vector Machines (SVMs) or Artificial Neural Networks (ANNs), SOMs do not require labeled training data, making them particularly

suitable for exploratory data analysis and identifying hidden patterns within stock market data. Furthermore, SOMs preserve the topological properties of the input space, thus capturing the relationships between different market conditions effectively (Kohonen, 2013).

In financial prediction, SOMs offer unsupervised clustering and visualization capabilities that differ fundamentally from supervised learners like SVM, RF, and LSTM. SOMs are particularly suitable for applications like market regime detection, portfolio segmentation, and pattern discovery – tasks where understanding the clustering of input data is key. For example, mapping stocks or trading days into clusters can reveal relationships that drive a trading strategy, as noted by Deboeck (1998).

## **2.2. TECHNICAL ANALYSIS FOR TRADING ASSETS**

Technical indicators are tools used to analyze stock price movements and can be classified into various types, including trend-following, momentum, volatility, and volume-based indicators. These indicators provide valuable insights into market trends, price momentum, risk levels, and trading volume.

Examples of widely used technical indicators include Moving Averages (MA), which smooth price fluctuations to identify trends, and the Exponential Moving Average (EMA), which reacts more sensitively to recent price changes (Murphy, 1999). Momentum indicators such as the Relative Strength Index (RSI) help identify overbought or oversold market conditions (Wilder, 1978), while Bollinger Bands, a volatility indicator created to identify potential price levels of overbought and oversold conditions (Bollinger, 1992). Volume-based indicators like On-Balance Volume (OBV) analyze trading volume to assess buying and selling pressure (Murphy, 1999).

Prior research suggests that technical indicators can be effective in forecasting stock price movements and guiding profitable trading strategies. For example, Brock et al. (1992) found that even relatively simple rules, such as moving averages and trading volume, can generate excess returns, thereby challenging the efficient market hypothesis. Building on these findings, Chiang et al. (2012) showed that more complex indicators, including RSI oscillators and parabolic strategies, can outperform buy-and-hold approaches where the trader adopts a passive approach taking a long position in an asset or index and passively keeping it untouched for a long period (Agarwal & Naik, 2000), especially in emerging markets. Furthermore, combining multiple indicators, such as moving averages with RSI or MACD, can enhance the accuracy of signals and reduce false positives (Chio, 2022).

Despite the known capabilities of technical indicators in creating lucrative trading strategies, the information they hold can be especially valuable when combined with advanced machine learning algorithms, as it allows for the identification of non-linear relationships and subtle market patterns that may not be evident through traditional analysis.

Technical indicators are essential tools in financial analysis as they transform raw market data into more meaningful metrics that highlight trends, momentum, volatility, and other critical aspects of price behavior.

By incorporating these indicators, the models gain access to features that can reveal hidden relationships and patterns in the data, thereby improving their ability to classify and predict market movements. This step is particularly important in this study, as the added complexity and richness of the data enable the SOM model to better represent market behavior and achieve more robust results.

As we will later see in more detail in the methodology section, the data used to create the SOM model originates from Yahoo Finance, consisting of basic Open, High, Low, Close (OHLC) prices of an asset. While these raw data points provide a fundamental view of market activity, they lack the depth required to fully capture underlying market dynamics and patterns. To address this limitation and enhance the models' learning capacity, technical indicators derived from the raw data are integrated into the dataset.

### 2.3. STATE OF THE ART

In this section we would like to present a few papers and studies that are closely related to what we present on this thesis and therefore have a higher impact on the decisions made and paths taken.

Their main ideas and results are presented individually in the following paragraphs, but to ease the reader, Table 1 shows a quick comparison between them all.

Table 1 – State of the art papers

Reference	Objective	Method	Key Takeaways
<b>Chen &amp; He (2000)</b>	Use SOM to identify chart patterns in TAIEX and test trading	6x6 SOM on 125-day windows; compares SOM trading vs. buy-and-hold	- SOM clusters correspond to classic chart patterns. - Strategy beat buy-and-hold.
<b>Fu et al. (2001)</b>	Discover frequent time-series patterns with reduced data	Use Perceptually Important Points (PIPs) then SOM for pattern clustering	- PIPs simplify data yet preserve shapes. - SOM clusters reveal recurring “sub-patterns.”
<b>Resta (2007)</b>	Develop intraday SOM-based trading signals for S&P MIB Index	100 SOMs per market regime; combine outputs for signals	- SOM signals often produce higher Sharpe ratios than buy-and-hold. - Multiple SOMs enhance robustness.
<b>Kossakowski &amp; Bilski (2017)</b>	SOM-driven investment decisions on Warsaw Stock Exchange	Various SOM sizes/configurations; includes “conscience” mechanism	- Conscience SOM improves coverage & performance. - Removing correlated inputs can boost returns.

<b>Pei et al. (2023)</b>	Enhance trading via multi-scale fuzzy SOM on candlestick images	Two-layer Deep Fuzzy SOM + GRU; uses price + volume candlestick charts	<ul style="list-style-type: none"> <li>- Fuzzy SOM helps handle noise.</li> <li>- Multi-scale approach + GRU yields strong results on futures &amp; forex.</li> </ul>
--------------------------	---	--	---

Chen and He (2003) developed a SOM-based trading strategy that, for much of their test period, outperformed a simple buy-and-hold (B&H) benchmark, achieving at the end of the period a 60% higher result than B&H. Their study used 7,435 daily closing prices of the Taiwan Stock Index (TAIEX) and subdivided them into overlapping windows of 125 observations, each representing a short-term price movement. These windows were then fed into a 6x6 SOM to produce 36 representative patterns. Crucially, most of these SOM-identified patterns were followed by clear buy or sell signals, suggesting they could be harnessed for profitable trading. It is also notable that the 36 patterns can map onto familiar chartist categories such as uptrend, downtrend, V-formation, rounding bottom, rounding top, double tops, and island reversal, reinforcing the idea that recognizable technical chart patterns can indeed signal market predictability.

The concept of segmenting the dataset into smaller windows and using a SOM to analyze these windows, rather than individual data points, is particularly compelling. By subsequently matching these learned patterns against the test data, we gain a deeper appreciation for how market movements evolve over time. This underscores the importance of looking beyond a single day's price action and examining the market's behavior in the days leading up to that point.

In a different attempt to use SOM for trading, Resta (2007) proposed a SOM-based trading framework that uses five-minute intraday OHLC data from the Italian S&P MIB Index to generate buy, sell, or standby signals. Similar to Chen and He (2003), Resta (2007) also segments the data into blocks of periods representing, but in a particular different way, having upward, downward and sideward market phases, dedicating 70% of each block to train the SOMs and 30% to test them. Within each phase, 100 SOMs were trained and their outputs averaged to enhance robustness. Three distinct trading rules are then applied to convert SOM cluster assignments into trading signals. Across most market blocks, these SOM-based signals yield higher Sharpe ratios (Sharpe, 1998) and greater annualized returns than a straightforward buy-and-hold strategy, underscoring the potential of self-organizing maps in leveraging intraday price movements.

The idea of first pinpointing the market's overall trend and then training multiple SOMs on the same dataset in Resta (2007) is particularly interesting, as it addresses two key challenges. First, it enables the SOM algorithm to interpret different market conditions more effectively; second, it mitigates the randomness from individual map initializations, thereby enhancing the statistical reliability of the results. Moreover, evaluating the model against a buy-and-hold

(B&H) benchmark using financial performance measures provides a more straightforward and interpretable way to assess its effectiveness.

Continuing this exploration of SOM-driven pattern discovery, Fu et al. (2001) presented a method for discovering frequently occurring patterns in time-series data using self-organizing maps (SOM). Like Chen and He (2000), the authors begin by splitting the time series into subsequences (segments) via a sliding window. However, rather than training the SOM directly on raw data, they introduce an additional step employing Perceptually Important Points (PIPs), which identify the most significant turning points within each segment. By focusing on only these critical points, PIPs substantially reduce the number of data points while preserving the essential shape of the time series. The reduced representations are then used to train a SOM that groups similar patterns more efficiently. Although the authors note that these pattern clusters could inform trading strategies, they do not develop or test a standalone trading strategy based on their discovered patterns.

Perceptually Important Points (PIPs) may offer a straightforward yet effective way to enhance the performance of SOMs in spotting patterns within stock data that may indicate predictable price movements. In essence, the PIP algorithm works by iteratively locating the most significant “turning points” within a segment of the time series. It begins by marking the first and last points of the segment as critical; then, among the unmarked points, it selects the one farthest (in vertical distance) from the straight line connecting the two adjacent PIPs. This newly chosen point becomes an additional PIP. The process continues until the desired number of PIPs is reached. By distilling a potentially long sequence down to only these crucial points, PIPs preserve the segment’s most pronounced fluctuations while eliminating less salient details, allowing the subsequent SOM to more efficiently cluster and analyze the essential shape of the time series.

Evaluating the need for robust parameter tuning and broader pattern coverage, Kossakowski and Bilski (2017) also proposed an investment decision framework that centers on Self-Organizing Maps (SOM) to guide trading decisions in equity markets, configuring SOMs with various dimensions and parameters to cluster past price and technical indicator data into distinct “patterns.” Each pattern is then associated with a specific buy, sell, or hold signal. Notably, the authors introduce a “conscience” mechanism to ensure that rarely activated SOM nodes still participate in the learning process, thereby enhancing the variety of patterns captured. In their experiments on the Warsaw Stock Exchange, they test multiple SOM configurations using daily or intraday data, documenting how changes in the map size, training epochs, and learning strategies affect predictive performance.

The study by Kossakowski and Bilski (2017) highlights that SOM parameters, including the number of neurons and the learning scheme, must be carefully tuned to balance classification accuracy and robustness of trading signals. In many cases, conscience-enhanced SOMs provide improved rate-of-return outcomes by ensuring broader coverage of historical volatility. Moreover, removing highly correlated inputs before SOM training is found to reduce

model complexity and potentially boost performance. Their results underscore that SOM-based trading strategies, while requiring thorough calibration, can outperform simpler methods by intelligently mapping price histories into clusters that reflect actionable market patterns.

Pushing the frontier further with deep architectures, Pei et al. (2023) extended the SOM framework into a Deep Fuzzy Self-Organizing Map (DFSOM) architecture tailored to financial time series forecasting and trading. Their method first transforms raw price and volume data into extended candlestick charts, encoding both directional price movement and volume activity into image-like representations. They then compute histogram-of-gradient (HOG) features from these candlestick images and feed them into a two-layer DFSOM, which can cluster the resulting feature vectors across multiple time scales. The fuzzy component in DFSOM refines traditional SOM learning by handling uncertainties and high noise, common in volatile market data, through partial memberships.

After clustering, Pei et al. (2023) train separate predictive models—gated recurrent units (GRUs)—for each identified cluster, leveraging context-specific behaviors in each pattern. Their empirical tests on both commodity futures and foreign exchange markets show that DFSOM can isolate meaningful fluctuation modes and improve overall trading performance. Crucially, combining fuzzy logic with multi-scale SOM layers facilitates recognition of subtle price-volume patterns that would be obscured by simpler feature engineering methods. This research underscores the advantages of integrating fuzzy clustering, deep SOM architectures, and dedicated neural predictors to adapt trading rules dynamically to shifting market regimes.

## **2.4. LITERATURE GAP**

Although studies suggest that SOMs can be efficient in financial data and seen in the examples by Deboeck and Kohonen (1998) or in Back et al. (1998) and have been successfully applied to build trading models based on technical indicators as in Chi et al. (2000), Afolabi and Olude (2007), Mohamed (2019), Hsu (2011), even with intraday data as in Resta (2007) and with image recognition on candlesticks as in Pei et al. (2023) the examples are not numerous as most authors chose to rely on other machine learning models for their analysis, most relying on deep learning models (Atsalakis & Valavanis, 2009; Li & Bastos, 2020).

Along with that, most papers that do use SOMs do not have a method to differentiate between stocks when applying the models. According to Li and Bastos (2020) most papers on stock price prediction and algorithmic trading strategies do not show a comparison against other classical technical trading strategies, such as RSI or Crossing Moving Averages. Additionally, most of these works are not focused on Brazilian stocks.

Unlike other works that rely on SOMs and technical indicators, such as Hsu (2011) where a hybrid SOM and genetic programming model is proposed to predict stock prices, or in Afolabi and Olude (2007) where the authors compare three different methods, namely SOM, Backpropagation and hybrid SOM to predict stock prices, the model presented in this paper

has no intention on predicting the future price of trading assets with precision, but rather create a trading model that can predict the general direction of the market and upon it, signal when to enter and exit trading positions in order to generate profitable trading strategies.

This paper builds on top of, specially, the papers mentioned in section 2.3 to fill the gaps presented by providing a relatively simple model using SOMs that can predict the general direction of the market and provide the basis for an algorithmic trading strategy that can outperform not only a buy-and-hold strategy but also traditional technical trading strategies.

### 3. METHODOLOGY

This chapter outlines the conceptual framework and methodological rationale used to develop a Self-Organizing Map (SOM) based trading strategy for the index of the Brazilian stock market. It describes how the data is obtained and prepared, how technical indicators and Perceptually Important Points (PIPs) are employed to enhance pattern discovery, how the SOM algorithm is used to cluster market behavior, and how the resulting clusters are translated into trading signals for backtesting.

The goal of the methodology deployed was to identify a single, robust trading model that delivers the best results in average across the entire 24-year training dataset (2000–2023). By applying grid search and evaluating each candidate model via backtesting on unseen data, we ensured that the chosen configuration would be selected solely for its real-world trading performance rather than any fitting measure. The best configuration for the SOM would result in a generalist yet specialized model: one that is sufficiently adaptable to broader market conditions but still finely tuned to the Brazilian Stock Market Index (IBOV).

To outlines a broad overview of the methodology used to develop the proposed trading strategy Figure 2 – Graphical Abstract, captures the data flow and the essential analytical stages that form the foundation of the experimental design and implementation of the approach.

Each icon in the graphical abstract is later explained in more detail in the sections of this chapter.

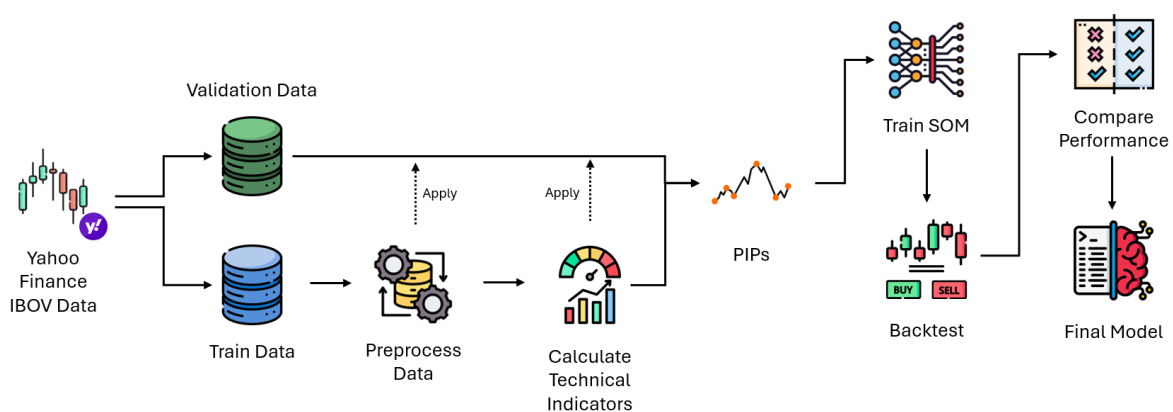


Figure 2 – Graphical Abstract

#### 3.1. DATA ACQUISITION

The study uses daily historical market data from the Ibovespa (Índice Bovespa), the primary benchmark stock index for the Brazilian exchange (B3 – Brasil, Bolsa, Balcão). By focusing on

the Ibovespa, which tracks the most liquid and actively traded stocks in Brazil, the research gains a comprehensive view of Brazilian equity market trends.

Data was collected over an extended time horizon (2000–2025) to ensure the strategy's robustness across multiple market regimes.

To implement and test the approach, daily historical data on the Ibovespa (Índice Bovespa) was retrieved using the yfinance Python library (Ran Aroussi, 2017/2024). The data spans from January 2000 to April 2025, capturing multiple market cycles and price regimes. For each trading day, the dataset includes Open, High, Low, and Close prices (OHLC).

After downloading, each year's dataset was stored in individual CSV (comma separated values) files to ensure reproducibility and easy reference.

The resulting dataset contained 5,843 daily records (after cleansing).

### 3.2. DATA PREPROCESSING

The training data undergoes preprocessing steps that include cleansing and normalization. This ensures consistency and prepares the data for the extraction of meaningful patterns. The same preprocessing is subsequently applied to the validation data to ensure comparability and consistency across training and evaluation phases.

Beyond removing rows with missing values, no extensive data-correction or imputation procedures were taken. Missing entries were too infrequent to justify more invasive fixes, and abstaining from them avoids injecting assumptions that might distort the model's behavior.

To ensure that all features have a similar scale, preventing certain variables from dominating the learning process, we have chosen the Tanh Estimator for data normalization as for forecasting applications where data is highly dynamic and non-linear, such as stock market indices, data normalization directly impacts predictive performance and empirical studies have shown that the Tanh Estimator normalization is a more effective in scaling raw data, leading to improved model accuracy. Bhanja and Das (2019) demonstrated that, among various normalization techniques, the Tanh Estimator yielded the lowest prediction errors in DRNN models. In parallel, Nayak et al. (2014) emphasize that non-linear normalization methods outperform linear techniques like Min-Max when dealing with complex, volatile time series data. Therefore, the Tanh Estimator is the normalization method of our choice. It can be mathematically expressed as follows.

$$x_{norm} = \frac{1}{2} \cdot \left( \tanh \left( scale \cdot \frac{x - \mu}{\sigma} \right) + 1 \right)$$

Equation 1 – Tanh Estimator

Where:

- $x$  represents an individual data point
- $\mu$  is the mean of the dataset
- $\sigma$  is the standard deviation

The scale is a parameter that controls the steepness of the tanh function, a factor of 0.01 was used.

The hyperbolic tangent function,  $\tanh(\cdot)$  compresses the data values into the interval  $(-1,1)$ . By applying the transformation  $\frac{1}{2} \cdot (\tanh(\cdot) + 1)$ , the output is subsequently scaled into the range  $(0,1)$ . This approach mitigates the influence of outliers and improves the robustness of the data, which can enhance the performance of subsequent predictive models.

### 3.3. TECHNICAL INDICATORS

To enrich the feature set, a representative set of technical indicators is calculated based on the raw price and volume data. Indicators such as Bollinger Bands, Relative Strength Index (RSI), Moving Average Convergence Divergence (MACD), and selected moving averages serve to capture different dimensions of market behavior—trend, momentum, and volatility.

In principle, these indicators provide the SOM with more nuanced signals than raw price data alone. They highlight overbought/oversold conditions, emerging trends, and price volatility, all of which can be significant in identifying patterns with predictive power for trading.

During grid search only two possibilities were explored with technical indicators, they were either all used together, or none would be used at all. When used they always had the same attributes, which are presented in Table 2.

Nine technical indicators were calculated using the Technical Analysis (Pedial, Darío López, 2018/2025) library in Python, providing a richer feature set than raw price alone. As summarized in Table 2 – Technical indicators, the selected indicators include measures of volatility (Bollinger Band width, Average True Range), trend (various moving averages, MACD histogram), and momentum (Relative Strength Index). Each indicator was computed using default or commonly used parameter windows. Their selection was guided by popularity, documented efficiency in literature, and diversity of indicator types.

Table 2 – Technical indicators

Name	TA Name	Parameters	Type
Bollinger Band Channel Width	bollinger_wband	Window = 20 Std Deviation = 2	Volatility
Bollinger Band Moving Average	bollinger_mavg	Window = 20 Std Deviation = 2	Volatility
Average True Range	atr	Window = 14	Volatility
Relative Strength Index	rsi	Window = 14	Momentum
Moving Average Convergence Divergence Histogram	macd_diff	Window Slow = 26 Window Fast = 12 Windows Signal = 9	Trend

Exponential Moving Average 10	ema_indicator	Window = 10	Trend
Exponential Moving Average 30	ema_indicator	Window = 30	Trend
Simple Moving Average 10	sma_indicator	Window = 10	Trend
Simple Moving Average 30	sma_indicator	Window = 30	Trend

Although the complete set up of Bollinger Bands (BB) and the Moving Average Convergence Divergence (MACD) include multiple curves, we have opted for their most representative components only. For Bollinger Bands, we excluded the lower and upper bands, any signals triggered when prices close beyond these thresholds, and the band percentage channel. Retaining only the channel width and the midpoint moving average. Similarly, for MACD, we have omitted both the MACD line and the signal line, relying solely on the histogram.

Four moving averages were employed as they are commonly analyzed in pairs, where signals are generated when one moving average crosses another.

The decision to limit our model to a streamlined set of indicators is well justified by prior studies, which indicate that incorporating too many technical indicators can dilute the predictive signal and introduce noise and unnecessary complexity to a model, and the best is to stick with a set of 5–10 technical indicators (He, 2023). Alsubaie et al. (2019) provides evidence that a compact set, particularly when it includes key indicators like Bollinger Bands, Moving Averages, and the Relative Strength Index, yields superior performance by enhancing signal clarity and reducing misclassification costs, the usage of these 3 indicators is also supported by Chio (2022). Similarly, He (2023) demonstrates that a focused feature set optimizes cumulative returns and minimizes drawdowns, reinforcing our approach of leveraging a minimal yet effective set of indicators.

### **3.4. PERCEPTUALLY IMPORTANT POINTS (PIPs)**

Perceptually Important Points (PIPs) condense a time series into its most critical turning points. By focusing on peaks, troughs, and major inflection points, PIPs emphasize structural changes in price movements or technical indicator trajectories rather than every minor fluctuation.

Perceptually Important Points (PIPs) are strategically selected data points that succinctly capture the essential movements of a time series by preserving significant peaks, troughs, and critical turning points. Instead of using every minor fluctuation, PIPs allow the algorithm to summarize important shifts in trends and movements that might hold predictive significance. This technique was chosen due to its effectiveness in reducing computational overhead while retaining the interpretability and predictive power of market patterns as shown in Fu et al. (2001). This helps the SOM capture meaningful patterns without being overwhelmed by irrelevant micro-movements. In effect, PIPs act as a dimensionality-reduction strategy tailored to time-series data.

A representation of how PIPs work can be seen in Figure 3 - PIP iteration process. Adapted from Fu et al. (2001).

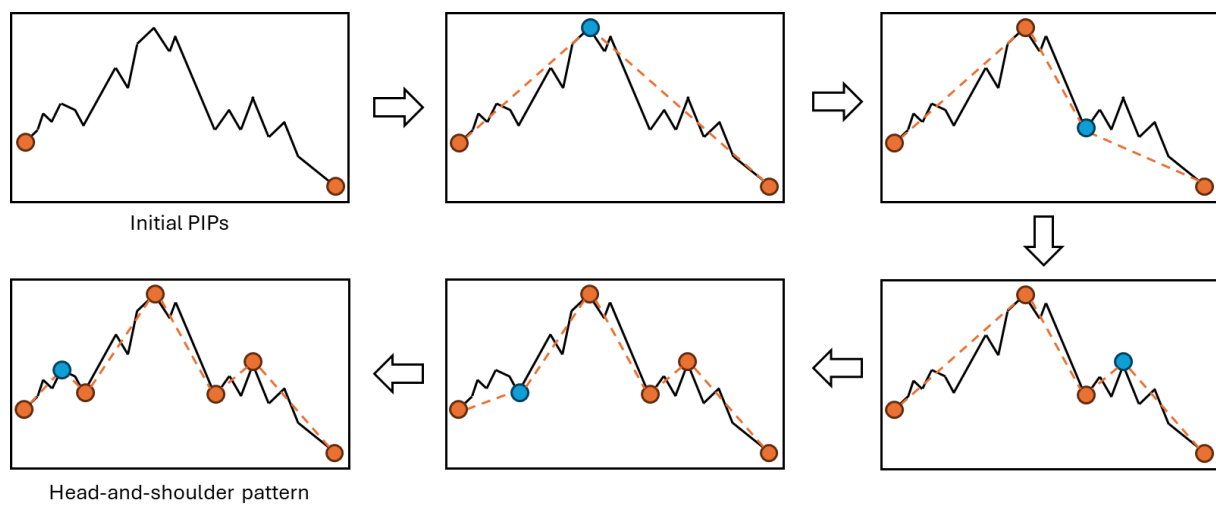


Figure 3 - PIP iteration process. Adapted from Fu et al. (2001)

The extraction of PIPs was implemented algorithmically as follows:

1. Defining the Window Size:

The extraction process begins by selecting a sliding window of a fixed size over the time series. We used a single sliding window of 90 trading days.

2. Initial Selection of Points:

Within each window, the first and last points are initially selected as PIPs. These anchor points represent the starting and ending positions of the local trend under analysis.

3. Iterative Selection Process:

A recursive algorithm then iteratively identifies additional points within each window. Specifically:

- The vertical distance between each intermediate point in the series and the linear interpolation between previously selected PIPs (anchor points) is calculated.
- The point exhibiting the greatest vertical distance (i.e., the most significant deviation from the linear interpolation) is identified as an additional PIP.
- This process continues iteratively, selecting further points as new anchors, until the desired number of PIPs for each window is reached.

4. Normalization of Selected Points:

After selection, each PIP is normalized to a relative index within the window (from 0 at the window's start to 1 at the end). This normalization facilitates consistent comparisons and inputs to subsequent clustering processes.

A single window-size of 90 trading days with a total of 9 PIPs per window was used for both the close price and each technical indicator.

To determine the significance of candidate points during the extraction of Perceptually Important Points (PIPs), a two-step calculation was used. First, for a candidate point  $i$  located between two anchor points previously selected at positions  $y_{start}$  and  $y_{end}$ , a linear interpolation value  $y_{interp}(i)$  is computed as:

$$y_{interp}(i) = y_{start} + \frac{(y_{end} - y_{start}) \times (i - start)}{end - start}$$

Equation 2 – Determine Candidate Point Significance

Then, the vertical distance  $d(i)$  between the actual value  $y_i$  of the candidate point and this interpolated value is calculated using the following equation:

$$d(i) = |y_i - y_{interp}(i)|$$

Equation 3 – PIPs Extraction

Where:

- $y(i)$  is the actual value of the point at index  $i$ .
- $y_{end}$ ,  $y_{start}$  are values of previously selected anchor points (PIPs).
- $y_{interp}(i)$  represents the linearly interpolated value at position  $i$  between the two anchor points.

The selection process is iterative and greedily selects points with maximum distance at each step until the desired number of PIPs is identified, effectively maximizing the perceptual importance of each selected point.

### 3.5. MODEL VALIDATION

To ensure the model generalizes well and does not overfit, we first reserved data from 2024 to March 2025 as our final test data, not including it in any part of the model building, and applied a leave-one-out cross-validation on the remaining data for training, below is the process of how this was achieved.

- Select one year as the validation set  
That year's data, e.g., 2000, is withheld for out-of-sample validation.
- Train on all other years  
SOM is trained on PIPs derived from the remaining years, e.g., 2001 through 2023, ensuring that the validation year (2000) remains unseen during training.

- Repeat for each year

This procedure is repeated 24 times, once for each year in the range 2000–2023, so that each year eventually serves as the validation year.

The first reason for adopting this leave-one-year-out approach is that SOM was built with Perceptually Important Points (PIPs) to identify structural market shapes, such as peaks, troughs, and inflection points. Because these shapes can appear at various points in time, training on all but one year while reserving the excluded year for validation still allows the model to uncover and test for recurrent patterns across different periods. Training on data from all but one year thus allows the SOM to learn a wide range of potential patterns, while the excluded year remains entirely unseen for validation.

Another factor is the desire for multiple out-of-sample checks. By iterating through each calendar year as the validation set, the model is confronted with distinct market contexts, including bull, bear, and sideways conditions, thereby providing a more comprehensive view of its generalizability. This practice helps guard against overfitting to a particular era or type of market regime.

Our leave-one-out approach deliberately lets the Perceptually Important Points (PIPs) algorithm scan the entire price series, including data that fall after the training window, before we train the Self-Organizing Map (SOM). This is not a data-leak problem, because PIPs only label *where* structural landmarks (peaks, troughs, inflections) occur; they don't reveal directional information the SOM can exploit. Since financial markets are believed to cycle through similar shapes, using future prices to identify cleaner, more consistent landmarks in the past is entirely consistent with the goal of learning time-agnostic patterns.

After the PIP pass, we train the SOM using MiniSOM library in Python (Vettigli, Giuseppe, 2013/2018) on all years except one and validate on that unseen year. Rotating the held-out year gives multiple out-of-sample tests that expose the model to bull, bear, and sideways regimes, helping ensure the patterns it learns generalize across eras rather than overfitting to a single market climate.

Although this arrangement departs from standard forward-looking assumptions, each validation year remains strictly unseen, preserving the integrity of the out-of-sample evaluation.

The leave-one-out cross-validation approach proposed can be visualized in Figure 4, Figure 5 and Figure 6, illustrating how the dataset was divided. Each figure has three colored sections:

- Yellow: Data that is not directly used for training or testing (often included so we can compute features like PIPs over a prior window without contaminating the training/test sets).
- Green: Data used for training the model.
- Blue: Data used for validating (unseen data) to evaluate model performance.

Below is the explanation of each section of Figure 4. The other two types can relate to this same explanation but less complex.

90 Trading Days (1) – A buffer window used to calculate PIPs for the upcoming Train Data (1).

Train Data (1) – Used to train the model. Its first 90 entries depend on PIPs calculated from “90 Trading Days (1).” Its last 30 entries compare predictions against the subsequent 30-day period.

30 Trading Days (1) – Used by the following Test Data section to compute its PIPs and also used by Train Data (1) to assess actual outcomes.

Validation Data – The unseen period used to evaluate model performance. Its first 90 entries use PIPs from the previous sections (i.e., they rely on data from “Train Data (1)” and “30 Trading Days (1)” but do not leak future information back into the training).

90 Trading Days (3) – Another buffer window for calculating PIPs before the next training segment.

Train Data (2) – A second training segment. Its first 90 entries rely on “90 Trading Days (3)” for PIPs; its last 30 entries evaluate how the market performed in the subsequent 30-day period.

30 Trading Days (2) – Used by “Train Data (2)” to evaluate the identified PIPs.

The other figures (3 and 4) follow the same principle but place the test window at different parts of the timeline. In all cases, chronological ordering is preserved so no data leakage occurs, ensuring a more realistic assessment of how the model performs on truly unseen future data.

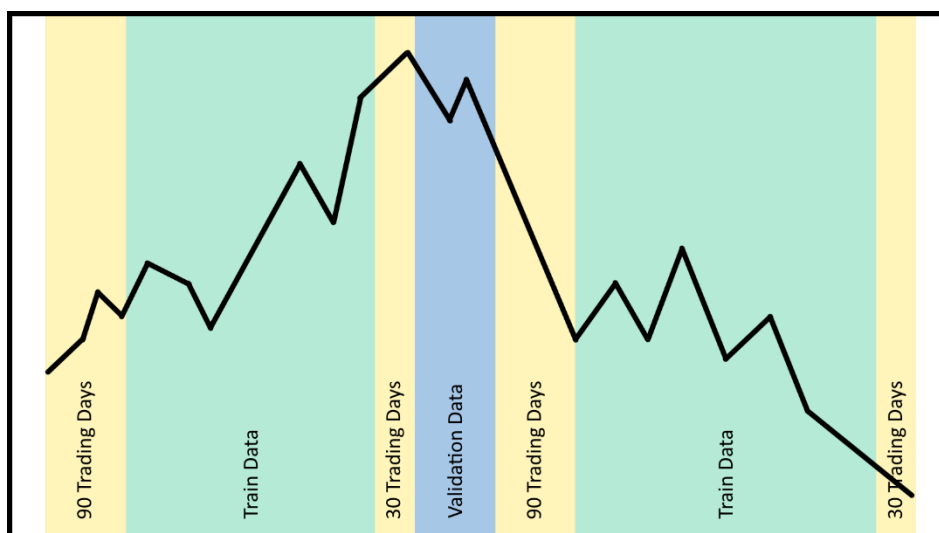


Figure 4 – Cross-validation where Validation Data is between two other Train Sets

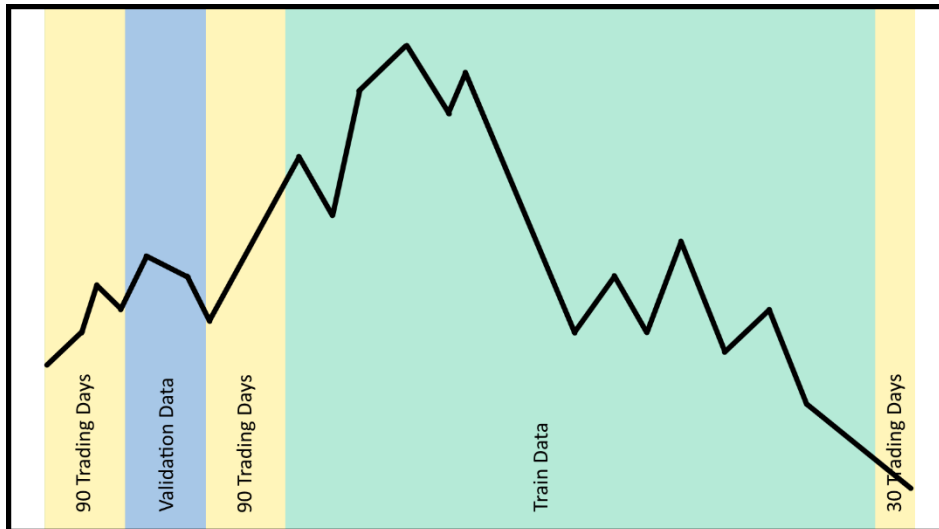


Figure 5 – Cross-validation where Validation data is at the beginning

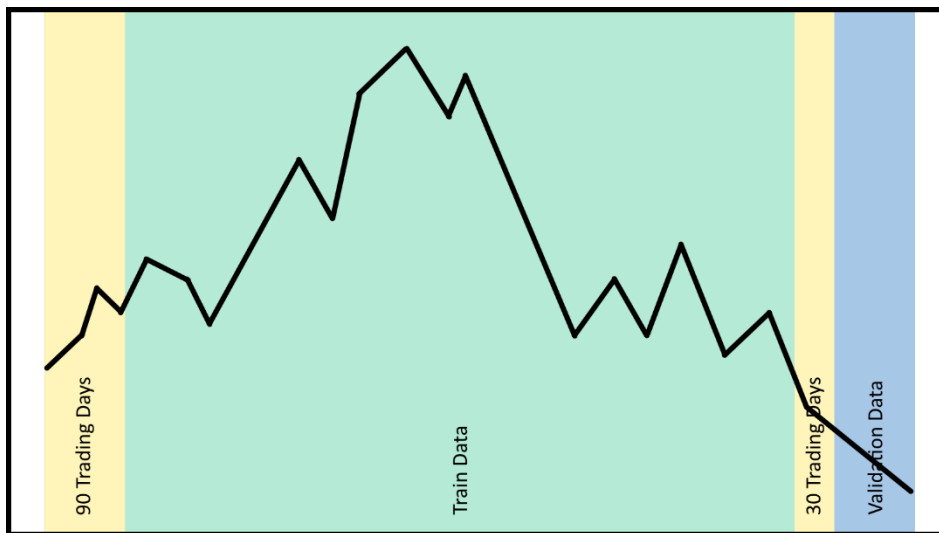


Figure 6 – Cross-validation method where Validation data is at the end

### 3.6. BACKTESTING AND SOM EVALUATION

As the key goal of this research is to build an SOM-based trading model that maximizes profitability, rather than finding a model that minimizing traditional clustering, and that can beat other traditional trading strategies, in special Buy-and-hold. Consequently, conventional SOM quality measures, such as topographic error or quantization error, were not evaluated. Instead, we evaluated the SOM by backtesting its trading signals via a time-series cross-validation approach, ensuring that each candidate model was tested on unseen data.

For each data point in the evaluation dataset the future 30 days return was calculated. As every point belongs to a SOM cluster, this data was then grouped by cluster and a list of metrics was calculated, as shown in Table 3 below.

Table 3 – Metrics Calculated for each Cluster

<b>Metric</b>	<b>Description</b>
<b>Mean</b>	The average future return for each cluster.
<b>Standard Deviation</b>	The standard deviation of future returns for each cluster.
<b>Median</b>	The median future return for each cluster.
<b>Cluster Occurrence</b>	The number of occurrences of each cluster.
<b>Sharpe Ratio</b>	The Sharpe ratio for each cluster, calculated as mean / std, scaled to a percentage.
<b>Negatives</b>	The total count of negative future returns for each cluster.
<b>Positives</b>	The total count of positive future returns for each cluster.
<b>Positive – negative</b>	The difference between the count of positive and negative future returns for each cluster.
<b>Positive Occurrence</b>	The percentage of positive future returns for each cluster, calculated as positives / (positives + negatives) * 100.
<b>Negative Occurrence</b>	The percentage of negative future returns for each cluster, calculated as negatives / (positives + negatives) * 100.
<b>Mean Up</b>	The mean of positive future returns for each cluster, scaled to a percentage.
<b>Mean Down</b>	The mean of negative future returns for each cluster, scaled to a percentage.
<b>Mean Difference</b>	The difference between mean up and the absolute value of mean down, representing the net difference between positive and negative returns.

Two of which were used to signal buy and sell calls for the backtesting, the Mean and the Positive Occurrence, generating two sets of grid search, each using a different approach for drawing the trade signals. The one using the Mean would place a buy signal on clusters that had a positive average future return. The other one, using the Positive Occurrence ratio would signal a cluster as bullish if the ratio was above or equal to 60%.

Since the trading strategies tested were exclusively long trades—where an asset is purchased with the expectation of selling at a higher price—any cluster that did not qualify for a buy signal was automatically labeled with a sell signal.

By focusing on actual trading results rather than abstract clustering errors, we ensured that the grid search favored SOM configurations truly capable of generating superior returns. This approach directly aligns the selection process with the practical objective of maximizing profitability in real-world trading scenarios.

### **3.7. MODEL COMPARISON**

Four other long-trading models were built for backtesting to compare the performance of the proposed model against other traditional trading strategies, these are presented in Table 4.

Table 4 – Comparing Trading Strategies

<b>Strategy</b>	<b>Description</b>
<b>Buy-and-hold</b>	Simply purchase the asset at the beginning of the backtesting period and hold it until the end, without performing any trades in between.
<b>Moving Average Crossover (MAC) Strategy</b>	Uses two moving averages (a short-term and a long-term one). A buy signal is generated when the short-term moving average crosses above the long-term moving average (golden cross), and a sell signal is generated when the short-term average crosses below (death cross). Two versions are used: one with SMA and another with EMA.

<b>Relative Strength Index (RSI) Mean-Reversion Strategy</b>	Uses RSI to identify overbought and oversold conditions. Buys occur when the RSI dips below a certain oversold threshold (e.g., RSI < 30), and sells when the RSI rises above a certain overbought threshold (e.g., RSI > 70).
<b>MACD Histogram</b>	Signals to buy when the histogram crosses from negative to positive, and to sell when the histogram crosses from positive to negative.

### 3.8. EMPIRICAL STUDY

This section details the configuration of the dataset, the time-series splits used for training and validation, the technical indicators and Perceptually Important Points (PIPs) extraction procedure, the Self-Organizing Map (SOM) training process, and finally the backtesting framework used to assess the profitability of the proposed trading strategy.

Starting with the PIPs calculation which were done after obtaining the data, cleansing and calculating the chosen technical indicators. Each rolling 90-day segment was transformed into PIPs with 9 points each. Drawing on Fu et al. (2001), the algorithm identifies nine critical “turning points” within each window, capturing peaks, troughs, and other significant movements. For each window, the same procedure is applied separately to the close price and each of the nine technical indicators, resulting in a new feature for each, these features are what is fed to train the SOM model.

A grid search approach was taken following parameters displayed in Table 5 – Changing settings on grid search, and Table 6 – Immutable setting on grid search. These parameters were chosen after a few preliminary tests and to ensure that the process would run in a manageable timeframe due to limited computational resources. In Table 5, the Range column specifies the values tested for each parameter during the grid search, and the Step column indicates the increment between successive values. For example, the year parameter was varied from 2000 to 2023 in increments of 1 (2000, 2001, 2002, ...), and the sigma parameter was varied from 0.5 to 4.0 in increments of 0.5 (0.5, 1.0, 1.5, ...).

Table 5 – Changing settings on grid search

Variable	Description	Range	Step
Year	The years used to train and validate the model.	2000 to 2023	1
som_x and som_y	The dimensions of the SOM grid. Only square grids were used.	6 to 18	2
sigma	Controls how many nearby neurons are updated during training.	0.5 to 4.0	0.5
learning_rate	Determines how much a neuron’s weight vector changes in response to the input.	0.1 to 0.5	0.1
lambda_threshold	Distance threshold used to decide whether two SOM clusters are similar enough to be merged.	0.1 to 0.3	0.1
Technical Indicators	Either technical indicators should be used or not.	Boolean	NA
Backtesting Evaluation	How to determine how good the model is.	Mean Diff or Positive %	NA

Table 6 – Immutable setting on grid search

Setting	Description	Value
neighborhood_function	Shape of the neighborhood around the winning neuron.	gaussian
activation_distance	Distance metric used to compare input vectors to weight vectors.	euclidean
topology	Determines how the cells in the grid connects to one another.	hexagonal
num_iteration	Defines how many times the SOM will be updated using samples from the dataset.	5000
window_size	The number of trading days on each window where the PIPs would be identified.	90
num_pips	The number of PIPs in each window.	9
future_window	Number of trading days in the future used to evaluate the market behavior after a cluster.	30

After training, we examine the pairwise distances of the resulting cluster. If two clusters were separated by a distance smaller than a predefined threshold ( $\lambda_{\text{threshold}}$ ), we merged them into a single cluster. This helps reduce over-fragmentation and increases interpretability, as shown in Fu et al. (2001), which is also a step taken by Chen and He (2003). The final output is a set of consolidated clusters, each representing distinct types of PIP-based patterns in the data.

During backtesting, which was a part of the grid search, two different approaches were taken to label clusters as bullish or bearish as previously described in section 3.6. The backtesting results of every single iteration of the grid search were stored in CSV files, each one corresponding to a validation year containing all the metrics previously described in Table 3 in section 3.6.

To address either the model could really yield better performance than other traditional trading strategies we backtested not only the best configuration identified for the SOM model, but also 4 others traditional trading strategies for comparison. All were executed in data from 2000 – 2023, same used in the construction of our model using the same logic of long trades only and the same library for calculating the necessary technical indicators.

## 4. RESULTS AND DISCUSSION

### 4.1. BEST OVERALL CONFIGURATION IN BACKTESTING

A total of 80,640 tests were performed, 3,360 per each validation year.

The grid search result of each validation year was put into separate CSV files, a transposed preview of a single line of one of these CSVs can be seen in Table 7, with description added for clarity.

Table 7 – Grid search result table sample

Columns	Values	Description
Index	3200	Auto increment index that identifies the setting used
Ticker	BVSP	The ticker of the asset
Final Diff	3.44	The % gain/lost at the end of the year
Avg Diff	1.78	The daily average difference against buy-and-hold in %
Median Diff	1.83	The daily median difference against buy-and-hold in %
Positive Days	194	Number of days above buy-and-hold
Negative Days	50	Number of days below buy-and-hold
Positive %	79.51	% of days above buy-and-hold
Negative %	20.49	% of days below buy-and-hold
Window Size	90	Length of trading days for PIPs
Num PIPs	9	Number of PIPs per window
SOM X	18	SOM grid X dimension
SOM Y	18	SOM grid Y dimension
Sigma	3.0	Controls the size of the neighborhood around the winning node in SOM
Learning Rate	0.2	Determines how much the weights of the BMU and its neighbors are updated toward the input vector in SOM
Iterations	5000	Total number of iterations for SOM
Lambda Threshold	0.3	Threshold to merge similar clusters
Prediction Year	2021	Validation year
Signal Base	mean	Signal method to identify trades (Depends on backtesting approach, Mean or Positive Occurrence)
Initial Balance	100	Initial balance for backtesting
Technical Indicators	TRUE	Boolean for the use of indicators

Given the extensive number of tests, there are many ways to define the “best” model. For example, one might choose the model that outperforms the market the most at the end of the period. However, this approach might favor a model that performs weakly for most of the period but has a strong finish, while overlooking models that consistently outperform the market throughout. Alternatively, one could define the best model as the one with the highest average daily outperformance, but this might ignore models that achieve exceptional gains on fewer, but more significant, days. Other metrics, like the consistency of positive returns or

even risk-adjusted measures, further complicate the definition of “best.” These examples illustrate that the choice of metric can dramatically alter the ranking of models.

To address this challenge, we evaluated the models using the following ranking system, based on the columns shown in Table 5 (which serves as a sample to illustrate the available data):

1. **Average Difference (Avg Diff):**  
We calculated the mean of the daily differences between the model return and the buy-and-hold strategy (using the Avg Diff column), then ranked the models in descending order.
2. **Final Difference (Final Diff):**  
We calculated the mean of the final differences between the model return and the buy-and-hold strategy (using the Final Diff column), then ranked the models in descending order.
3. **Positive Days Percentage (Positive %):**  
We calculated the mean percentage of days on which the model outperformed the buy-and-hold strategy (using the Positive % column), then ranked the models in descending order.
4. **Combined Ranking:**  
All the ranks from the metrics above were summed, creating a final aggregated rank. The model with the best overall performance (i.e., the lowest total rank) was considered the top performer.

In other words, the data was grouped by each configuration set where we computed the average values across the years for Final Diff, Avg Diff, and Positive %, then ranked them individually, with higher performance receiving lower rankings. Finally, we summed these individual rankings to produce an overall ranking, with lower sums indicating superior configurations.

Based on this final ranking, the top 20 results are presented in Table 8, and the parameters used in the best 20 configurations are presented in Table 9.

Table 8 – Top 20 configurations

Index	Final Diff	Avg Diff	Positive %	Final Diff Rank	Avg Diff Rank	Positive % Rank	Sum Rank
3200	1.75	1.90	66.99	6.0	6.0	1.0	13.0
2272	1.57	2.27	65.14	9.0	3.0	5.0	17.0
148	1.29	1.87	65.55	18.5	7.5	3.5	29.5
144	1.29	1.87	65.55	18.5	7.5	3.5	29.5
796	2.57	1.64	61.38	2.0	21.0	20.0	43.0
164	1.80	1.50	62.53	5.0	36.0	13.0	54.0
0	1.20	1.75	61.94	24.0	14.0	18.0	56.0
4	1.21	1.75	61.94	24.0	14.0	18.0	56.0
8	1.21	1.75	61.94	24.0	14.0	18.0	56.0

<b>1864</b>	1.57	1.46	63.69	10.0	39.0	8.0	57.0
<b>2072</b>	1.09	1.79	61.36	32.0	11.0	21.0	64.0
<b>2348</b>	0.90	1.71	62.86	45.0	17.0	11.0	73.0
<b>1860</b>	1.16	1.35	62.37	29.0	47.0	14.0	90.0
<b>2248</b>	0.60	1.76	62.63	67.0	12.0	12.0	91.0
<b>748</b>	1.21	1.55	58.97	22.0	27.0	53.0	102.0
<b>1620</b>	0.66	2.32	59.49	61.0	2.0	46.0	109.0
<b>656</b>	0.92	1.42	61.27	44.0	43.0	22.0	109.0
<b>1220</b>	1.06	1.54	58.91	36.0	28.0	54.0	118.0
<b>264</b>	3.15	1.72	57.14	1.0	16.0	109.0	126.0
<b>196</b>	1.13	1.09	61.02	31.0	70.0	25.0	126.0

Table 9 – Parameters of the top 20 best configurations

<b>Index</b>	<b>SOM X</b>	<b>SOM Y</b>	<b>Sigma</b>	<b>Learning Rate</b>	<b>Lambda Threshold</b>	<b>Signal Base</b>	<b>Technical Indicators</b>
<b>3200</b>	18	18	3.0	0.2	0.3	mean	TRUE
<b>2272</b>	14	14	3.0	0.5	0.2	mean	TRUE
<b>148</b>	6	6	1.5	0.3	0.2	mean	TRUE
<b>144</b>	6	6	1.5	0.3	0.1	mean	TRUE
<b>796</b>	8	8	3.0	0.2	0.2	mean	TRUE
<b>164</b>	6	6	1.5	0.4	0.3	mean	TRUE
<b>0</b>	6	6	0.5	0.1	0.1	mean	TRUE
<b>4</b>	6	6	0.5	0.1	0.2	mean	TRUE
<b>8</b>	6	6	0.5	0.1	0.3	mean	TRUE
<b>1864</b>	12	12	4.0	0.1	0.2	mean	TRUE
<b>2072</b>	14	14	1.5	0.3	0.3	mean	TRUE
<b>2348</b>	14	14	4.0	0.1	0.3	mean	TRUE
<b>1860</b>	12	12	4.0	0.1	0.1	mean	TRUE
<b>2248</b>	14	14	3.0	0.3	0.2	mean	TRUE
<b>748</b>	8	8	2.5	0.3	0.2	mean	TRUE
<b>1620</b>	12	12	2.0	0.1	0.1	mean	TRUE
<b>656</b>	8	8	1.5	0.5	0.3	mean	TRUE
<b>1220</b>	10	10	2.5	0.2	0.3	mean	TRUE
<b>264</b>	6	6	2.5	0.3	0.1	mean	TRUE
<b>196</b>	6	6	2.0	0.2	0.2	mean	TRUE

By analyzing these last two tables we see that if the ranking system have not been implemented and we for a chance would have selected solely the Avg Diff as the metric to measure the best performing configuration, it was not even present in the top 20 listed in Table 8.

From Table 9 we see that all top 20 configurations used the mean approach for backtesting and all of them used technical indicators, and it seems to be somewhat of a correlation (0.58) between the size of the SOM grid and the Sigma.

Table 10 provides a year-by-year breakdown of our strategy’s performance under the optimal parameter settings. The Prediction Year column denotes the calendar year in which the backtest was conducted. Final Diff represents the difference, measured at year-end, between the portfolio value generated by the SOM strategy and that of the buy-and-hold benchmark; positive values indicate that the strategy outperformed the benchmark. Avg Diff is the arithmetic mean of the daily differences between strategy and benchmark returns over the course of the year, while Median Diff is the median of those daily return differences, highlighting the typical daily magnitude of out- or under-performance. The Signal Sharpe Ratio column displays the risk-adjusted return of the strategy’s trading signals, calculated as the strategy’s excess return divided by its return volatility. By contrast, the Market Sharpe Ratio” is the analogous metric for the benchmark itself, allowing for direct comparison of risk-adjusted performance. Finally, Positive % indicates the proportion of trading days during which the strategy outperformed the benchmark, days on which the daily difference was positive.

Table 10 – Results by year using best overall configuration

Prediction Year	Final Diff	Avg Diff	Median Diff	Signal Sharpe Ratio	Market Sharpe Ratio	Positive %
2000	-4.03	-0.11	-0.81	-0.62	-0.34	33.90
2001	-6.67	-1.91	-2.18	-0.49	-0.22	30.83
2002	23.34	11.54	14.38	0.29	-0.47	92.65
2003	-6.33	1.27	1.34	3.11	2.84	73.17
2004	2.74	2.88	3.50	0.80	0.69	96.37
2005	4.92	2.62	2.25	1.58	1.19	86.69
2006	19.31	11.70	17.44	2.00	1.30	96.71
2007	3.59	3.12	2.57	1.53	1.40	99.59
2008	6.14	5.00	5.80	-0.64	-0.72	92.71
2009	10.53	8.84	10.68	2.37	1.91	99.17
2010	-1.42	0.76	1.49	-0.03	0.05	71.54
2011	9.15	5.06	3.91	-0.32	-0.73	100.00
2012	0.79	2.10	3.33	0.28	0.24	77.50
2013	8.98	4.93	5.24	-0.43	-0.87	100.00
2014	-10.99	-5.84	-4.07	-0.41	0.09	15.10
2015	12.68	1.58	0.28	0.20	-0.38	58.65
2016	-6.42	-3.14	-2.37	1.48	1.49	29.44
2017	-3.67	-0.80	-0.79	1.31	1.42	26.82
2018	0.98	3.00	1.66	0.77	0.66	96.61
2019	-12.02	-5.58	-5.90	1.04	1.47	4.88
2020	5.63	1.86	0.75	0.36	0.24	99.19
2021	3.44	1.78	1.83	-0.36	-0.49	79.51
2022	-3.16	0.36	-0.10	0.24	0.37	45.16
2023	-15.53	-5.43	-6.10	0.77	1.47	1.62

The overall best configuration beats the market on 14 of the 24 years, in the years that it performed better than buy-and-hold it had an average final result of 8.01% higher, in the years that it performed worse it had an average of -7.02%. For the Avg Diff on these same conditions the rate was 4.02% and -3.26% respectively.

The Sharpe rate of the model was slightly higher than buy-and-hold on average, depicting 0.61 against 0.52, respectively.

There does not appear to have significant correlation between the market behavior in a year and the model performance, that is, the model does not seem to work better on years with higher or lower oscillations, or on year that are predominantly bull or bear markets.

## **4.2. COMPARISON WITH OTHER TRADING STRATEGIES**

Figure 7 presents a comparative analysis of cumulative returns on the January 2024 to March 2025 test data, showcasing the best-performing model configuration identified in Table 6 against the buy-and-hold strategy, as well as strategies employing RSI, MACD, Bollinger Bands, Exponential Moving Averages and Simple Moving Averages.

Specifically, in 2024, the selected model achieved superior results, outperforming the market (buy-and-hold) by 11.82% at year's end and maintaining an average daily outperformance of 6.60%. Additionally, the model demonstrated superior returns compared to most alternative strategies, with RSI being the only strategy yielding higher returns at the end of the period. But it is important to note that the proposed model achieved higher returns than all other strategies from beginning of August until mid-December, and that it had comparable results to those of RSI alone since April. Moreover, the proposed model, along with most traditional strategies, had very good timing at the longer uptrend of the year spanning from mid-June until the end of August, maintaining the bullish position throughout the entire period.

For the first 3 months of 2025, the model did not perform as good as the market, missing a few short pumps at the beginning of the year, but the good performance in 2024 compensated, and even drove it to surpass the RSI strategy. At the end of March, the model was 9.05% above buy-and-hold.

Throughout the entire period we see that MACD performance is quite similar to buy-and-hold, sometimes overperforming and at other underperforming the market, a possible combination of MACD and RSI strategies could possibly result in a better model alone which has already been noticed by Chio (2022), although using the default setting of the RSI strategy and on the data used in this study, this strategy does not provide many trades.



Figure 7 – Strategies Comparison

Table 11 displays a few metrics of the very same strategies returns displayed in Figure 7. Final result is the final value at the last day of March 2025 for each strategy, Avg Return gives the overall mean return of each strategy throughout the entire period, and max loss and max gain give the max point of loss and gain in relation to the starting point.

The table makes it clearer that the proposed model is the best for the period tested as it gives the best average return and highest return at a given point while keeping the loss at a minimum.

Table 11 – Strategies Comparison Metrics

	Final Result	Avg Return	Max Loss	Max Gain
<b>SOM Model</b>	108.45	103.31	2.80	12.26
<b>Buy and Hold</b>	99.40	96.22	10.67	3.50
<b>EMA</b>	90.52	92.72	12.58	0
<b>SMA</b>	90.30	91.89	13.38	0
<b>Bollinger Bands</b>	95.61	94.56	10.04	0
<b>RSI</b>	105.07	102.97	2.42	5.07
<b>MACD</b>	99.08	97.13	7.78	1.78

As seen, the SOM-based model delivers the highest cumulative return and strongest average performance over January 2024–March 2025, Figure 8 drills down into how it achieves those gains. Here, the same equity curves are plotted for the SOM model, the RSI, and buy-and-hold, but now overlaid with green and red markers that denote each strategy’s entry and exit points.

In the first half of 2024, the SOM strategy moves through a series of small fluctuations around the 100 starting level, capturing small up-moves while avoiding the deeper drawdowns suffered by buy-and-hold, which steadily erodes nearly ten points by mid-year. Over this same interval the RSI filter remains flat at 100, having not triggered any trades. Beginning in June, a

green triangle marks the SOM's well-timed entries on the strong mid-year rally, and red triangles flag exit just ahead of late August's plateau. The RSI approach by contrast executes a single upward jump to roughly 105, which initiated at about the same moment as the SOM model (green circle) and liquidated the position (red circles) at late July and sat out the rest of the tested period.

As the market becomes less stable and starts to fall starting from October, the SOM model threads a succession of shorter trades, not all well-timed, but that still outperforms the market by simply not taking on the entirety of the down trend. The RSI strategy neither participates in nor suffers this drawdown.

Again in February 2025 the SOM model signals a long trade just before the market suffers a pump, which is not followed by the RSI strategy and is also not liquidated until the end of the test period, leading it to its final result of 108.45 points against 105.07 and 99.40 points of the RSI and Buy-and-hold strategies respectively.

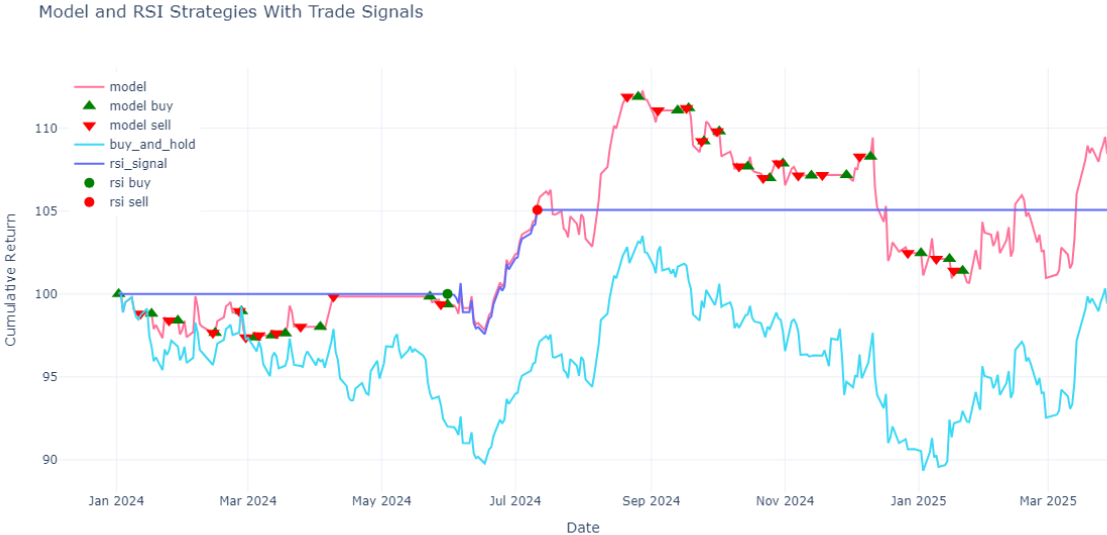


Figure 8 – SOM Model vs RSI and Buy and Hold with trade signals

## 5. CONCLUSIONS AND FUTURE RESEARCH

This study explored the application of Self-Organizing Maps (SOMs) to develop an algorithmic trading strategy for the Brazilian stock market, specifically targeting the Ibovespa index. By integrating technical indicators, Perceptually Important Points (PIPs), and unsupervised clustering, the proposed model aimed to outperform traditional strategies such as buy-and-hold, moving average crossovers, RSI-based and other technical indicators based approaches. The results demonstrate that SOMs, when combined with PIPs and a robust grid search framework, can effectively identify recurring market patterns and generate actionable trading signals, offering a compelling alternative to conventional methods.

### 5.1. KEY FINDINGS

- **Moderate Performance:** The SOM-based strategy outperformed buy-and-hold in 14 out of 24 validation years and demonstrated competitive results against most technical strategies. For instance, in the test year 2024, it achieved a 96% positive day rate against the benchmark, though it underperformed the RSI strategy at the end of the year. This highlights the model's potential to capitalize on market inefficiencies while suggesting room for refinement.
- **Effectiveness of PIPs:** The use of Perceptually Important Points reduced computational complexity while preserving critical market dynamics, enabling the SOM to cluster structural patterns (e.g., peaks, troughs) more efficiently.
- **Adaptability:** The leave-one-out cross-validation and grid search approach ensured the model's robustness across diverse market regimes, including bull, bear, and sideways conditions.

### 5.2. THEORETICAL AND PRACTICAL IMPLICATIONS

This research challenges the strong form of the Efficient Market Hypothesis (EMH) by demonstrating that SOMs can exploit recurring patterns in historical data to generate excess returns. It also contributes to the growing body of literature on machine learning in finance, particularly in emerging markets like Brazil, which are often underrepresented in algorithmic trading studies. Practically, the model provides traders with a data-driven framework to automate entries and exits, reducing emotional biases and enhancing decision-making consistency.

### 5.3. FUTURE STUDIES

Despite the strengths demonstrated by the SOM-PIP-based approach, there remain several areas that warrant further investigation to enhance its practical utility and resilience in real-world trading scenarios. By systematically exploring and addressing these aspects, researchers and practitioners can refine the methodology and potentially achieve improved performance and adaptability in diverse market conditions. The following points outline key directions for future development:

1. **Parameter Optimization:** The optimization of SOM parameters and architectures to enhance trading performance.
2. **Dynamic Risk Management:** Incorporate stop-loss mechanisms, position sizing, and short-selling capabilities to improve risk-adjusted returns.

In conclusion, this study demonstrates that SOMs, when combined with PIPs and technical indicators, can serve as a viable tool for algorithmic trading in the Brazilian stock market. While still leaving room for improvement, the approach shows measurable promise and aligns with existing research on SOMs in finance. Future work should focus on refining robustness, expanding scope, and addressing real-world trading constraints to unlock its full potential.

## BIBLIOGRAPHICAL REFERENCES

- Afolabi, M. O., & Olude, O. (2007). *Predicting Stock Prices Using a Hybrid Kohonen Self Organizing Map (SOM)*. 48–48. <https://doi.org/10.1109/HICSS.2007.441>
- Agarwal, V., & Naik, N. Y. (2000). Performance Evaluation of Hedge Funds with Option-Based and Buy-and-Hold Strategies. *SSRN Electronic Journal*. <https://doi.org/10.2139/ssrn.238708>
- Alsubaie, Y., Hindi, K. E., & Alsalman, H. (2019). *Cost-Sensitive Prediction of Stock Price Direction: Selection of Technical Indicators*. 7, 146876–146892. <https://doi.org/10.1109/ACCESS.2019.2945907>
- Atsalakis, G. S., & Valavanis, K. P. (2009). Surveying Stock Market Forecasting Techniques – Part II: Soft Computing Methods. *Expert Systems with Applications*, 36(3, Part 2), 5932–5941. <https://doi.org/10.1016/j.eswa.2008.07.006>
- Atsalakis, G. S., & Valavanis, K. P. (2010). Surveying Stock Market Forecasting Techniques-Part I: Conventional Methods. *Journal of Computational Optimization in Economics and Finance*, 2(1), 45–92.
- Back, B., Sere, K., & Vanharanta, H. (1998). Analyzing Financial Performance with Self-Organizing Maps. *1998 IEEE International Joint Conference on Neural Networks Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98CH36227)*, 1, 266–270 vol.1. <https://doi.org/10.1109/IJCNN.1998.682275>
- Bhanja, S., & Das, A. (2019). *Impact of Data Normalization on Deep Neural Network for Time Series Forecasting* (No. arXiv:1812.05519). arXiv. <http://arxiv.org/abs/1812.05519>
- Bollinger, J. (1992). Using Bollinger Bands. *Stocks & Commodities*, 10(2), 47–51.

- Brock, W., Lakonishok, J., & LeBaron, B. (1992). Simple Technical Trading Rules and the Stochastic Properties of Stock Returns. *The Journal of Finance*, 47(5), 1731–1764. <https://doi.org/10.1111/j.1540-6261.1992.tb04681.x>
- Chen, S.-H., & He, H. (2003). Searching Financial Patterns with Self-Organizing Maps. In Chen, Shu-Heng & Wang, Paul P. (Eds.), *Computational Intelligence in Economics and Finance* (pp. 203–216). Springer Berlin Heidelberg. <https://www.aiecon.org/staff/shc/pdf/cief2000-1.pdf>
- Chi, S.-C., Kuo, R.-J., & Teng, P.-W. (2000). A Fuzzy Self-Organizing Map Neural Network for Market Segmentation of Credit Card. 5, 3617–3622 vol.5. <https://doi.org/10.1109/ICSMC.2000.886571>
- Chiang, Y.-C., Ke, M.-C., Liao, T. L., & Wang, C.-D. (2012). Are Technical Trading Strategies Still Profitable? Evidence from the Taiwan Stock Index Futures Market. *Applied Financial Economics*, 22(12), 955–965. <https://doi.org/10.1080/09603107.2011.631893>
- Chio, P. T. (2022). A Comparative Study of the MACD-Base Trading Strategies: Evidence from the US Stock Market (No. arXiv:2206.12282). arXiv. <http://arxiv.org/abs/2206.12282>
- Cohen, G., & Cabiri, E. (2015). Can technical oscillators outperform the buy and hold strategy? *Applied Economics*, 1–9. <https://doi.org/10.1080/00036846.2015.1013609>
- Deboeck, G. (1998). Financial Applications of Self-Organizing Maps. *Neural Network World*, 8(2), 213–241.
- Deboeck, G., & Kohonen, T. (1998). *Visual Explorations in Finance: With Self-Organizing Maps*. Springer Science & Business Media.
- Fu, T., Chung, F., Ng, V., & Luk, R. (2001). Pattern Discovery from Stock Time Series Using Self-Organizing Maps. *Workshop Notes of KDD2001 Workshop on Temporal Data Mining*, 1.

<https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=331b863eb7e11b5d70fb37411c8d18378b052221>

He, F. (2023). *The Best Combination of Indicators for the U.S. Stock Market* (Handle: eprints.ubishops.ca/12345) [Master's thesis, Bishop's University]. Bishop's University Repository. <https://www.ubishops.ca/wp-content/uploads/he20240105.pdf>

Hoover, T. (2015). *A Case Against the Efficient Market Hypothesis* (Handle: 116099117/10349) [Honors thesis, Texas Christian University]. TCU Digital Repository. <https://repository.tcu.edu/server/api/core/bitstreams/efb3d26d-c669-4492-9308-4eaf1134252c/content>

Hsu, C.-M. (2011). A Hybrid Procedure for Stock Price Prediction by Integrating Self-Organizing Map and Genetic Programming. *Expert Systems with Applications*, 38(11), 14026–14036. <https://doi.org/10.1016/j.eswa.2011.04.210>

Kohonen, T. (1982). Self-Organized Formation of Topologically Correct Feature Maps. *Biological Cybernetics*, 43(1), 59–69. <https://doi.org/10.1007/BF00337288>

Kohonen, T. (2013). Essentials of the Self-Organizing Map. *Neural Networks*, 37, 52–65. <https://doi.org/10.1016/j.neunet.2012.09.018>

Kossakowski, P., & Bilski, P. (2017). Analysis of the Self-Organizing Map-Based Investment Strategy. *International Journal of Computing*, 10–17. <https://doi.org/10.47839/ijc.16.1.866>

Leković, M. (2018). Evidence for and Against the Validity of Efficient Market Hypothesis. *Economic Themes*, 56(3), 369–387. <https://doi.org/10.2478/ethemes-2018-0022>

Li, A. W., & Bastos, G. S. (2020). Stock Market Forecasting Using Deep Learning and Technical Analysis: A Systematic Review. *IEEE Access*, 8, 185232–185242. <https://doi.org/10.1109/ACCESS.2020.3030226>

- Lo, A. W., & MacKinlay, A. C. (1988). Stock Market Prices Do Not Follow Random Walks: Evidence from a Simple Specification Test. *The Review of Financial Studies*, 1(1), 41–66. <https://doi.org/10.1093/rfs/1.1.41>
- Luukka, P., Pätäri, E., Fedorova, E., & Garanina, T. (2016). Performance of Moving Average Trading Rules in a Volatile Stock Market: The Russian Evidence. *Emerging Markets Finance and Trade*, 52(10), 2434–2450. <https://doi.org/10.1080/1540496X.2015.1087785>
- Malkiel, B. G. (2003). The Efficient Market Hypothesis and Its Critics. *Journal of Economic Perspectives*, 17(1), 59–82. <https://doi.org/10.1257/089533003321164958>
- Mohamed, A. (2019). *Artificial Intelligence in Investing: Stock Clustering with Self-Organizing Map and Return Prediction with Model Comparison* (URN:NBN:fi-fe2019073023242) [Master's thesis, Lappeenranta University of Technology]. LUTPub. <https://urn.fi/URN:NBN:fi-fe2019073023242>
- Murphy, J. J. (1999). *Technical Analysis of the Financial Markets: A Comprehensive Guide to Trading Methods and Applications*. Penguin.
- Nayak, S. C., Misra, B. B., & Behera, H. S. (2014). Impact of Data Normalization on Stock Index Forecasting. *International Journal of Computer Information Systems and Industrial Management Applications*, 6, 13–23.
- Pedial, Darío López. (2025). *Technical Analysis Library* (Version 0.11.0) [Python]. <https://github.com/bukosabino/ta> (Original work published 2018)
- Pei, D., Luo, C., & Liu, X. (2023). Financial Trading Decisions Based on Deep Fuzzy Self-Organizing Map. *Applied Soft Computing*, 134, 109972. <https://doi.org/10.1016/j.asoc.2022.109972>

- Petrusheva, N., & Jordanoski, I. (2016). Comparative Analysis Between the Fundamental and Technical Analysis of Stocks. *Journal of Process Management. New Technologies*, 4(2), 26–31. <https://doi.org/10.5937/JPMNT1602026P>
- Ran Aroussi. (2024). *Yfinance: Download Market Data from Yahoo! Finance's Api* (Version 0.2.37) [Computer software]. <https://github.com/ranaroussi/yfinance> (Original work published 2017)
- Resta, M. (2007). Intraday Trading Rules Based on Self Organizing Maps. *International Workshop on Self-Organizing Maps: Proceedings*. <https://doi.org/10.2390/biecoll-wsom2007-108>
- Sharpe, W. F. (1998). The Sharpe Ratio. In P. L. Bernstein & F. J. Fabozzi (Eds.), *Streetwise* (pp. 169–178). Princeton University Press. <https://doi.org/10.1515/9781400829408-022>
- Shirani Faradonbeh, R., Shaffiee Haghshenas, S., Taheri, A., & Mikaeil, R. (2020). Application of Self-Organizing Map and Fuzzy C-Mean Techniques for Rockburst Clustering in Deep Underground Projects. *Neural Computing and Applications*, 32(12), 8545–8559. <https://doi.org/10.1007/s00521-019-04353-z>
- Uriarte, E. A., & Martín, F. D. (2008). Topology Preservation in SOM. *World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering*.
- Vettigli, Giuseppe. (2018). *Minisom: A Minimalistic Implementation of the Self Organizing Maps* (Version 2.3.0) [Computer software]. <https://github.com/JustGlowing/minisom> (Original work published 2013)
- Wilder, J. W. (1978). *New Concepts in Technical Trading Systems*. <https://dspace.lib.uom.gr/handle/2159/29408>



**NOVA Information Management School**  
**Instituto Superior de Estatística e Gestão de Informação**

Universidade Nova de Lisboa