ISSN 2786-7447
EUROPEAN JOURNAL OF THEORETICAL AND APPLIED SCIENCES
OPEN ACCESS
PEER REVIEWED

A Comparison Between Trend Following (Tidy Little Robot – TLR) and Reversal Trading (Dirty Little Robot – DLR) Algobot Strategies

Florentin Şerban
Applied Mathematics Department, Bucharest Academy of Economic Studies, Romania

Bogdan-Petru Vrînceanu
Master student, Bucharest Academy of Economic Studies, Romania

Suggested Citation

Şerban, F. & Vrînceanu, B.-P. (2023). A Comparison Between Trend Following (Tidy Little Robot – TLR) and Reversal Trading (Dirty Little Robot – DLR) Algobot Strategies. *European Journal of Theoretical and Applied Sciences, 1*(5), 805-822.
DOI: 10.59324/ejtas.2023.1(5).68

Abstract:

Algorithmic trading has become a defining force in modern financial markets, offering traders precision, speed, and automation. This article explores the world of algorithmic trading through the lens of two distinct algobots, Tidy Little Robot (TLR) and Dirty Little Robot (DLR). Developed using Pine Script 5 on the TradingView platform and implemented on the Pionex exchange, these algobots navigate the BTC/USDT perpetual market with a 1-minute timeframe. TLR excels in trend-following scenarios, while DLR specializes in identifying reversals. The study delves into their performance, methodology, implications, interpretations, limitations, and future research prospects, offering valuable insights into the evolving landscape of algorithmic trading.

Keywords: *algorithmic trading, cryptocurrency, backtesting, risk management, trading strategies; technical analysis.*

Introduction

This article serves as a comprehensive guide, delving into various facets of algorithmic trading, from fundamental concepts and strengths and weaknesses to a comparison between human trading and algorithmic trading. We also explore the empowering capabilities of Pionex, Pine Script 5, and TradingView in developing and backtesting algorithmic trading strategies. Transitioning to the research part of the article, we introduce the tools and platforms at the heart of algorithmic trading. Pionex, Pine Script 5, and TradingView play pivotal roles in our methodology, enabling the development, implementation, and backtesting of algorithmic trading strategies.

The article then ventures into the methodological aspects, providing a detailed look at the development, implementation, and backtesting phases of algorithmic trading strategies. Discussions, interpretations, and limitations are explored, shedding light on the real-world applicability and potential pitfalls of these strategies. As we embark on this journey through the world of algorithmic trading, we invite readers to explore the nuances, strategies, and possibilities that this evolving field has to offer. Whether you are a seasoned trader or a newcomer to the world of algorithmic trading, this article offers valuable insights and a deeper understanding of the tools and techniques that drive success in financial markets.

Theoretical Framework

Fundamental Concepts of Trading

In the world of financial markets, investors and traders utilize various tools and theoretical concepts to analyze asset prices, predict future movements, and make informed decisions. Among these tools, technical analysis stands as a prominent methodology. This essay explores the theoretical concepts within technical analysis, including Exponential Moving Average (EMA), Relative Strength Index (RSI), and Bullish and Bearish Engulfing patterns, shedding light on how they contribute to a deeper understanding of market dynamics (Goodman, 2019; Serban, Stefanescu, & Ferrara, 2013.

Technical Analysis: The Foundation

At its core, technical analysis is the art and science of examining historical price and volume data to forecast future price movements. Unlike fundamental analysis, which focuses on the intrinsic value of an asset, technical analysis relies on the belief that historical price patterns tend to repeat themselves due to market psychology and human behavior (Höppel, 2014).

Exponential Moving Average (EMA): Capturing Trends

The Exponential Moving Average (EMA) is a fundamental tool within technical analysis. It is a type of moving average that places greater weight on recent price data, making it more responsive to current market conditions. EMA calculates the average of a specific number of price data points over a specified time frame, assigning more importance to recent prices. This gives EMA the ability to capture trends more swiftly than its counterpart, the Simple Moving Average (SMA) (Mishkin, 2019). EMA serves various purposes in technical analysis. It helps traders identify trends, support and resistance levels, and potential entry and exit points in the market. A rising EMA indicates an uptrend, while a declining EMA suggests a downtrend. The crossover of shorter and longer-term EMAs can also be used to generate buy or sell signals (Mishkin, 2019).

Relative Strength Index (RSI): Momentum and Overbought/Oversold Conditions

The Relative Strength Index (RSI) is another vital concept in technical analysis. It is a momentum oscillator that measures the speed and change of price movements. RSI values range from 0 to 100 and are typically used to identify overbought and oversold conditions in an asset's price (Mishkin, 2019).. An RSI above 70 is often considered overbought, suggesting that the asset may be due for a correction or a trend reversal. Conversely, an RSI below 30 is considered oversold, indicating that the asset may be undervalued and due for a potential bounce-back. RSI not only helps traders identify potential trend reversals but also provides insights into the strength and sustainability of a trend.

Bullish and Bearish Engulfing Patterns: Reversal Signals

Bullish and Bearish Engulfing patterns are candlestick patterns used in technical analysis to identify potential trend reversals. These patterns are formed by two consecutive candlesticks and are crucial in assessing market sentiment (Davey, 2014). A Bullish Engulfing pattern occurs when a small bearish (downward) candlestick is followed by a larger bullish (upward) candlestick that completely engulfs the previous candle. This pattern suggests a potential reversal from a downtrend to an uptrend, indicating that buyers have taken control. Conversely, a Bearish Engulfing pattern emerges when a small bullish candlestick is followed by a larger bearish candlestick that completely engulfs the preceding one. This signals a potential shift from an uptrend to a downtrend, suggesting that sellers are gaining control (Davey, 2014; Chan, 2021).

Technical analysis, bolstered by tools like EMA, RSI, and candlestick patterns like Bullish and Bearish Engulfing, offers traders a systematic approach to interpreting price movements and identifying potential opportunities and risks in financial markets. While technical analysis is not without its critics, its widespread use in the financial industry highlights its enduring relevance and utility in today's dynamic markets

(Dedu, & Şerban, 2015; Serban, Stefanescu, & Ferrara, 2013).

Algorithmic Trading: A Comprehensive Exploration of Strengths, Weaknesses, Opportunities, and Threats

In today's ever-evolving financial landscape, algorithmic trading, affectionately referred to as "algobots," has emerged as a dominant force. These intricate automated systems, driven by sophisticated algorithms and data analytics, have fundamentally reshaped the dynamics of trading. This essay embarks on a deep dive into the realm of algobots, aiming to unravel their SWOT (Strengths, Weaknesses, Opportunities, and Threats) in a more expansive manner, shedding light on their multifaceted role in the financial markets (Nolte, 2015; Isichenko, 2021).

Strengths

Algobots possess a formidable array of strengths that underpin their ascendancy in modern trading. Their foremost attribute is their efficiency. Algobots execute trades with remarkable speed, often within mere milliseconds. This rapid execution empowers traders to pounce on fleeting price differentials and exploit arbitrage opportunities that would elude human traders. Moreover, algobots eliminate the inherent lag and potential for manual errors associated with human trading, enhancing overall operational efficiency (Mattli, 2018; Pardo, 2011). Another compelling advantage lies in their adeptness at data processing. Algobots are virtuosos when it comes to processing vast volumes of data in real-time. They adeptly ingest and analyze market news, historical data, and a myriad of technical indicators concurrently. This capacity enables them to make data-driven decisions swiftly, a feat that surpasses human capacity. Furthermore, algobots are champions of consistency. This unwavering consistency minimizes the adverse effects of human emotions and biases, which often cloud the judgment of traders (Winston, 2019; Crypto Trading Bot, n.a.). The ability to rigorously backtest trading strategies represents another potent asset. Algobots can be subjected to extensive backtesting on historical data. This meticulous testing process enables traders to refine and optimize their algorithms, enhancing their potential for peak performance.

Weaknesses

However, algobots are not without their weaknesses. Foremost among these is their inherent lack of adaptability. Algobots are designed to execute predetermined strategies with little room for flexibility. They struggle to navigate unforeseen market events or rapidly changing conditions that necessitate human judgment and adaptability. Consequently, they may continue executing predefined strategies even when those strategies become ineffective or counterproductive. Furthermore, algobots are susceptible to technical vulnerabilities. Despite their computational prowess, they are not impervious to technical glitches or connectivity issues. A minor malfunction can swiftly translate into unexpected losses, underscoring the importance of vigilant oversight and safeguards (Pardo, 2011). The third weakness revolves around their complexity. Developing and maintaining algorithmic trading systems of the highest caliber demands a profound level of technical expertise and substantial resources (Kendall, 2007; Chan, 2021).

Opportunities

On the flip side, algobots open doors to a myriad of opportunities in the financial landscape. Their role in democratizing market access is particularly noteworthy. By automating trading processes, algobots provide market access to a broader spectrum of traders, including individuals who would otherwise be deterred by the complexities of manual trading. Furthermore, the prevalence of algobots has sparked heightened interest in quantitative analysis and data science. Opportunities for professionals in these fields have surged, as the demand for quantitative analysts, data scientists, and algorithm developers continues to escalate in response to the rise of algobots (Gray, 2019). Moreover, the continuous evolution of algorithmic trading strategies offers fertile ground for traders and software developers to innovate and create algorithms that can outshine competitors.

Threats

Nonetheless, algobots are not without their share of threats. One notable threat arises from the specter of regulatory changes. As algorithmic trading continues to gain prominence, regulatory bodies have intensified their scrutiny of this domain. The imposition of stricter rules and regulations may impact the viability of algobots and escalate compliance costs (Gray, 2019; Bacidore, 2017). Another peril revolves around the potential for market manipulation. The sheer speed and volume at which algobots operate can inadvertently provide a fertile ground for market manipulation. Flash crashes and other market irregularities pose a palpable threat to market stability, necessitating constant vigilance and monitoring (Gray, 2019; Mattli, 2018). astly, there is the lingering concern of human displacement. The automation of trading processes raises questions about the future role and relevance of human expertise in financial markets.

In conclusion, algorithmic trading, embodied by the enigmatic algobots, embodies a double-edged sword in the world of finance. Their strengths, encompassing unmatched efficiency, data-processing prowess, unwavering consistency, and the potential for meticulous backtesting, have propelled them to the forefront of modern trading. However, they must grapple with the weaknesses of adaptability challenges, technical vulnerabilities, and inherent complexity (Kelleher, & Tierney, 2018). The opportunities that algobots present are marked by expanded market access, burgeoning interest in quantitative analysis and data science, and a fertile ground for the development of innovative trading strategies. Nevertheless, they must contend with threats such as regulatory changes, market manipulation risks, and concerns about job displacement. As algobots continue to evolve and integrate deeper into financial markets, striking a harmonious balance between human judgment and algorithmic precision remains imperative. Traders and investors who navigate this intricate landscape adeptly will be well-poised to harness the strengths of algobots while mitigating their weaknesses. They must also remain vigilant in seizing the opportunities that lie ahead while addressing the threats that accompany this transformative era in the financial industry (Scopino, 2020; Donadio & Ghosh, 2019).

A comparison between Human Trading and Algorithmic Trading

In the world of financial markets, two distinct approaches have emerged over time: human trading and algorithmic trading. The former relies on human intuition, experience, and emotions, while the latter employs complex computer programs to execute trades with speed and precision. This essay delves into the fundamental differences, advantages, and disadvantages of these two trading methods, highlighting the importance of striking a harmonious balance between them in today's dynamic financial landscape (Donadio & Ghosh, 2019).

Human trading is characterized by individual traders who make buy and sell decisions based on their judgment, analysis, and market expertise. This approach is inherently subjective, as it relies on human intuition and emotions.

Human traders possess unique advantages. They can interpret market news, sentiment, and global economic conditions effectively, adapting to rapidly changing market dynamics. Additionally, experienced human traders often develop an invaluable "gut feeling" for market movements, shaped by years of insights and strategies (Mattli, 2018).

However, human trading also carries its share of disadvantages. Emotional biases, such as fear and greed, can lead to impulsive decisions, overtrading, or holding onto losing positions. Furthermore, humans cannot match the speed of computers, resulting in slower execution times and missed opportunities in high-frequency trading environments (Kendall, 2007). Continuous monitoring of financial markets can also lead to stress and fatigue, which can impact decision-making and overall well-being. Algorithmic trading, also known as algo trading or automated trading, relies on computer algorithms to execute trades based on predefined criteria. These algorithms are designed to process vast amounts of data and execute orders swiftly and precisely.

The advantages of algorithmic trading are notable. Algorithms can execute trades within milliseconds, enabling traders to capitalize on price discrepancies and arbitrage opportunities. They eliminate emotional biases, ensuring that decisions are grounded solely in data and logic.

Additionally, algorithmic systems can maintain discipline and consistency in executing trading strategies, reducing the risk of human error. They can also be backtested on historical data to assess performance, allowing for optimization and fine-tuning (Kendall, 2007). However, algorithmic trading also has its disadvantages. Algorithms may struggle to adapt to unforeseen events or rapidly changing market conditions that require human judgment. Moreover, these systems can be vulnerable to technical glitches or connectivity problems that may result in unexpected losses.

In today's financial markets, achieving a balance between human and algorithmic trading is crucial. Both approaches have their unique strengths and weaknesses, and combining them can lead to superior results. Many institutional traders employ hybrid strategies that blend human expertise with algorithmic execution. In these approaches, human traders provide qualitative insights and adaptability, while algorithms handle the execution and quantitative aspects. Human traders can also oversee and manage algorithmic systems to ensure they align with the market's current dynamics and mitigate unexpected risks. Furthermore, traders should continually update their skills and understanding to remain competitive in an increasingly automated world. Gaining knowledge of algorithmic trading principles can empower human traders to make more informed decisions. Human trading and algorithmic trading represent two distinct yet complementary approaches to navigating the complexities of financial markets. While each method has its own set of advantages and disadvantages, they need not be mutually exclusive. Striking a balance between human intuition and algorithmic efficiency is essential for success in today's rapidly evolving trading environment. Traders who embrace this duality are likely to be better equipped to harness the strengths of both methods and adapt to the ever-changing landscape of financial markets (Kaufman, 2016; Kelleher, & Tierney, 2018).

Pionex, Pine Script 5 and TradingView: Empowering Algorithmic Trading Strategies

Pionex Exchange is a cryptocurrency trading platform designed to cater to the diverse needs of traders. Its intuitive user interface and an extensive array of features have made it a popular choice among cryptocurrency enthusiasts. Pionex offers a wide selection of cryptocurrencies and trading pairs, ensuring that traders can access the assets they desire (Crypto Trading Bot, n.a.; Tradingview.com, n.a.). One of Pionex's distinguishing features is its commitment to user-friendly trading. The platform is designed to be accessible to both novice and experienced traders. It provides a range of trading tools and features, making it easier for users to execute their strategies efficiently. Among these features, Pionex's support for algorithmic trading through webhooks is particularly noteworthy.

Algorithmic Trading with Webhooks: A Simplified Approach

Algorithmic trading, often referred to as algo trading, is a method of executing trading strategies using computer programs and algorithms. These algorithms can analyze market data, identify trading opportunities, and execute trades without human intervention. Algorithmic trading has gained immense popularity due to its potential to enhance trading efficiency and reduce emotional biases. One of the challenges traders face when venturing into algo trading is the complexity of implementation. Developing and deploying trading bots traditionally required a deep understanding of programming and infrastructure. However, Pionex has changed this landscape by introducing a user-friendly approach through webhooks (Crypto Trading Bot, n.a.; Tradingview.com, n.a.).

Understanding Webhooks in Algorithmic Trading

A webhook is a simple and effective way to integrate external services or applications with a trading platform like Pionex. In the context of algorithmic trading, a webhook serves as a bridge between a trading strategy developed on a separate platform, such as TradingView, and the execution capabilities of Pionex.

When a specific condition or event occurs in the external trading strategy (e.g., the identification of a buy signal), the webhook sends a notification to Pionex, instructing it to execute the corresponding trade. This seamless integration allows traders to implement their strategies without the need for complex programming or server setups.

Empowering Traders with Algobots on Pionex

Pionex's support for webhooks simplifies the process of deploying AlgoBots, making algorithmic trading accessible to a broader audience. Traders can create, backtest, and fine-tune their trading strategies on platforms like TradingView, leveraging their technical analysis skills.

Once their strategies are defined, traders can set up webhooks to transmit buy or sell signals directly to Pionex. Pionex then automatically executes the trades, ensuring precision and timeliness in strategy implementation. This approach eliminates the barriers that traditionally hindered traders from benefiting from algorithmic trading. In the realm of algorithmic trading, the ability to swiftly and effectively develop, test, and deploy trading strategies is paramount. TradingView, a popular charting platform among traders and investors, provides a valuable resource for this purpose with its versatile scripting language, Pine Script.

With the recent introduction of Pine Script 5, TradingView has elevated algorithmic trading to new heights, offering enhanced functionality and utility for the development of sophisticated algobots (Crypto Trading Bot, n.a.; Tradingview.com, n.a.). TradingView stands as a widely respected online platform known for its powerful charting and technical analysis tools. Traders and investors converge on TradingView to analyze financial markets, conduct research, and make informed decisions. The platform's intuitive interface and extensive library of indicators have made it a favorite among both novices and experienced market participants. Yet, TradingView's true strength lies in its scripting language, Pine Script. Pine Script empowers users to create custom indicators, strategies, and alerts, unlocking the potential for algorithmic trading and customized technical analysis. With the advent of Pine Script 5, TradingView has made significant strides in refining this scripting language, providing traders with a potent toolset for the development and implementation of algobots.

Pine Script 5 signifies a significant evolution in the capabilities of TradingView's scripting language. This latest iteration introduces several notable features and improvements that are invaluable for algorithmic traders. It embraces the principles of object-oriented programming (OOP), enhancing code organization and reusability while streamlining the development process. Additionally, Pine Script 5 is engineered for improved performance, featuring faster execution times and reduced memory consumption, which enables it to handle more complex strategies and calculations efficiently.

Furthermore, Pine Script 5 offers advanced data handling capabilities, allowing traders to work with multiple data series concurrently. This is particularly valuable when developing strategies that necessitate the analysis of multiple assets or timeframes. Pine Script 5 also introduces an array of built-in functions, simplifying intricate calculations and diminishing the need for external libraries. This fosters faster development and reduces the risk of errors. Custom plotting capabilities in Pine Script 5 afford traders greater flexibility, enabling them to visualize trading signals and patterns with precision, thereby enhancing the interpretability of strategies and indicators. The utility of Pine Script 5 for developing algobots is undeniable. Algorithmic trading strategies necessitate a robust environment for development and testing, and TradingView's platform, coupled with Pine Script 5, provides precisely that. Pine Script 5's intuitive syntax and comprehensive documentation facilitate rapid prototyping of trading strategies, allowing traders to iterate and fine-tune their algobots quickly. The platform also offers a comprehensive backtesting environment, permitting traders to test their algorithms rigorously on historical data. Pine Script 5's performance enhancements ensure faster and more accurate backtesting. Moreover, traders can deploy their algobots in real-time directly from TradingView, streamlining the transition from development to live trading. TradingView's community comprises a vibrant ecosystem of traders and developers who openly share Pine Script code, indicators, and strategies, fostering collaboration and significantly expediting algobot development (Crypto Trading Bot, n.a.; Tradingview.com, n.a.).

Pine Script 5, in tandem with TradingView, has emerged as a formidable tool for traders and developers venturing into the world of algorithmic trading. Its enhanced capabilities, performance optimization, and advanced data handling make it an ideal choice for crafting sophisticated algobots. With Pine Script 5's versatility and TradingView's user-friendly platform, traders have at their disposal a potent combination for developing, testing, and deploying trading strategies. Whether one is a seasoned algorithmic trader or a newcomer to the field, Pine Script 5 and TradingView offer a compelling gateway to the exciting and potentially profitable realm of algobots (Crypto Trading Bot, n.a.; Tradingview.com, n.a.).

Methodology

In this article, two algobots were created using Pine Script 5 on the TradingView platform. These algobots are designed to execute trading strategies for the BTC/USDT Perpetual trading pair on the Pionex exchange, focusing on a 1-minute time frame and the backtesting period of 11 days.

Tidy Little Robot (TLR): This algobot employs a trend-following strategy based on the identification of bullish and bearish engulfing candlestick patterns in conjunction with the Exponential Moving Average (EMA) 200. When a bullish engulfing pattern is detected above the EMA 200, a long position is initiated. Conversely, when a bearish engulfing pattern forms below the EMA 200, a short position is taken.

Dirty Little Robot (DLR): DLR, on the other hand, implements a reversal trading strategy. It identifies bullish and bearish engulfing patterns in combination with the Relative Strength Index (RSI). A long position is entered when a bullish engulfing pattern occurs alongside an oversold RSI reading. Conversely, a short position is initiated when a bearish engulfing pattern coincides with an overbought RSI reading.

Both TLR and DLR are programmed to adhere to a risk/reward ratio of 3:1. This means that the take profit level is set at 1.2% of the trading pair's value, while the stop loss level is positioned at 0.4%. This risk management approach is crucial in controlling potential losses and optimizing gains.

To evaluate the performance of TLR and DLR, a comprehensive backtesting process was conducted using the TradingView Strategy Tester. Historical price data for the BTC/USDT Perpetual trading pair on the Pionex exchange was utilized for this purpose. The backtesting period covered a significant timeframe, enabling a thorough assessment of the algobots' efficacy.

During the backtest, the algobots' trading signals, including entry and exit points, were executed in a simulated environment based on historical data. This allowed for the measurement of profitability, risk exposure, and overall performance.

In the subsequent section of this article, the implementation of the algobots on the Pionex exchange will be discussed in detail. Additionally, the results obtained from the backtesting procedure will be analyzed and presented to determine the algobots' viability in real-world trading scenarios. The study will provide insights into the effectiveness of the TLR and DLR strategies in the highly volatile cryptocurrency market, emphasizing their potential for generating consistent returns while effectively managing risk.

Development, Implementing and Backtesting

The successful development of the Tidy Little Robot (TLR) and Dirty Little Robot (DLR) algorithmic trading strategies paved the way for their implementation on the Pionex exchange. The integration process was achieved by utilizing TradingView's strategy alerts, which were seamlessly connected to the Pionex platform via a webhook. This integration allowed for the automatic execution of trading signals generated by TLR and DLR on the exchange, enabling real-time trading based on the predefined strategies.

Tidy Little Robot (TLR) Implementation

TLR's implementation involved the utilization of a trend-following strategy. When specific conditions were met, TLR executed trades accordingly. For long positions, TLR identified a bullish engulfing candlestick pattern coupled with the closing price being above the Exponential Moving Average (EMA) 200. Conversely, for short positions, a bearish engulfing pattern alongside the closing price being below the EMA 200 triggered trade execution.

Dirty Little Robot (DLR) Implementation

DLR, in contrast, employed a reversal trading strategy. It identified opportunities by looking for specific conditions. For long positions, DLR sought a bullish engulfing candlestick pattern occurring simultaneously with the Relative Strength Index (RSI) being in the oversold region (RSI < 30). For short positions, a bearish engulfing pattern aligned with an overbought RSI reading (RSI > 70) initiated trade execution.

Backtesting Results

The effectiveness of TLR and DLR was rigorously assessed through backtesting, simulating their performance over an 11-day period on the Pionex exchange, utilizing the BTC/USDT Perpetual trading pair. The backtesting results provide valuable insights into the strategies' viability and potential for generating profits.

TLR Backtesting Results

- Net Profit: 3.03%
- Total Closed Trades: 38
- Winrate: 39.47%
- Profit Factor: 1.375
- Average Trade: $0.08
- Ratio Avg Win / Avg Loss: 2.1

The backtesting results for TLR indicate a positive net profit of 3.03% over the 11-day testing period. TLR executed a total of 38 closed trades, achieving a win rate of 39.47%. The profit factor, which measures the ratio of winning trades to losing trades, stands at 1.375, demonstrating the strategy's effectiveness in generating profits. On average, each trade yielded $0.08 in profit, with a favorable ratio of average win to average loss at 2.1.

DLR Backtesting Results

- Net Profit: 0.78%
- Total Closed Trades: 8
- Winrate: 37.5%
- Profit Factor: 1.38
- Average Trade: $0.10
- Ratio Avg Win / Avg Loss: 2.3

DLR's backtesting results also indicate a positive net profit of 0.78% over the same 11-day period. DLR executed a total of 8 closed trades, achieving a win rate of 37.5%. The profit factor is 1.38, reflecting the strategy's ability to generate profits. Each trade, on average, produced $0.10 in profit, with a favorable ratio of average win to average loss at 2.3.

The implementation and subsequent backtesting of TLR and DLR on the Pionex exchange have yielded promising results. TLR's trend-following strategy demonstrated a net profit of 3.03%, while DLR's reversal trading strategy generated a net profit of 0.78% during the 11-day testing period. These results suggest that both algobots have the potential to be valuable tools for traders seeking to navigate the highly volatile cryptocurrency market.

The risk-reward ratio of 3, along with a take profit of 1.2% and a stop loss of 0.4%, contributed to effective risk management in both strategies, ensuring that potential losses were controlled while optimizing gains. The integration of TradingView's strategy alerts via webhook allowed for real-time execution, enhancing the practicality of these algobots in live trading scenarios. In the subsequent sections, a detailed analysis of the algobots' performance and their real-world application will be explored, shedding light on their capabilities and limitations.

Interpretation of the strategies and the pine script code

The Pine Script code for "Tidy Little Robot"

```
//@version=5
strategy("Tidy Little Robot", overlay = true)

//EMA
ema200 = ta.ema(close, 200)

//Candlestick Patterns
C_EngulfingBullish () =>
    close[1] < open[1] and
    open < close and
    close > open[1] and
    open < close[1]

C_EngulfingBearish () =>
    close[1] > open[1] and
    open > close and
    close < open[1] and
    open > close[1]

// Strategy Entry
long_condition = C_EngulfingBullish and close > ema200
short_condition = C_EngulfingBearish and close < ema200

if (long_condition)
    strategy.entry("Long", strategy.long)
if (short_condition)
    strategy.entry("Short", strategy.short)

//Strategy exit
stopLossPercent = 0.004
takeProfitPercent = stopLossPercent * 3

var float stopLossPrice = na
var float takeProfitPrice = na

if (strategy.position_size > 0)
    stopLossPrice := strategy.position_avg_price * (1 - stopLossPercent)
    takeProfitPrice := strategy.position_avg_price * (1 + takeProfitPercent)
    strategy.exit("Take Profit/Stop Loss", stop = stopLossPrice, limit = takeProfitPrice)

if (strategy.position_size < 0)
    stopLossPrice := strategy.position_avg_price * (1 + stopLossPercent)
    takeProfitPrice := strategy.position_avg_price * (1 - takeProfitPercent)
    strategy.exit("Take Profit/Stop Loss", stop = stopLossPrice, limit = takeProfitPrice)
```

Algorithmic trading is a sophisticated endeavor, often driven by meticulously crafted code designed to execute precise trading strategies. The Pine Script presented here is the framework for such a strategy, aptly named "Tidy Little Robot." This script is developed in Pine Script 5, a powerful tool used within the TradingView platform for creating custom trading strategies. We will break down this code, exploring its components and functions, to gain a deeper understanding of its inner workings.

The Pine Script begins with a script header, denoting its version and defining the trading strategy's name. In this case, it's "Tidy Little Robot," and it is set to overlay on the price chart, providing visual cues about its activity.

One of the critical components in algorithmic trading is the use of technical indicators. Here, the script calculates the Exponential Moving Average (EMA) with a period of 200 for the closing prices of the asset. This EMA serves as a crucial reference point for trend analysis.

The script allows the user to choose between two trend detection rules, either based on the Simple Moving Average (SMA) 50 or a combination of SMA 50 and SMA 200. These rules define whether we are in an upward or downward trend, which is fundamental for decision-making.

With trend rules in place, the script delves into candlestick pattern analysis, a staple of technical analysis in trading. It introduces various variables to characterize the patterns and then proceeds to define and detect specific candlestick patterns. These patterns are central to spotting potential entry and exit points in the strategy.

To enhance the user's experience, the script labels detected candlestick patterns on the chart, making them easily identifiable. The script even allows the user to customize the colors of these labels, providing a personal touch to the trading experience. He hearts of the strategy lies in its ability to detect engulfing patterns, a critical aspect of many trading strategies. It identifies both bullish and bearish engulfing patterns, crucial for determining when to enter or exit a trade. This section incorporates conditions to discern these patterns and generates alerts when they occur.

After pattern detection, the script moves to the core of the strategy – entry and exit conditions. It defines the logic for entering long or short positions based on the detected candlestick patterns and the relationship between the current price and the EMA200. Risk management is a fundamental consideration in algorithmic trading. The script incorporates logic for setting stop loss and take profit levels, allowing traders to limit their potential losses and secure profits at predefined levels.

In summary, "Tidy Little Robot" is a comprehensive algorithmic trading strategy developed in Pine Script 5. It skillfully combines trend analysis, candlestick pattern recognition, and risk management to execute precise trading decisions. Whether you are a seasoned trader or a novice, this script provides a versatile framework for creating and testing trading strategies within the TradingView platform.

Interpreting the Pine Script for "Tidy Little Robot"

Algorithmic trading strategies are becoming increasingly popular, as they offer a systematic and data-driven approach to trading financial markets. The Pine Script presented here represents one such strategy known as "Tidy Little Robot." This code is crafted in Pine Script 5, a powerful scripting language integrated into the TradingView platform, enabling the creation of customized trading strategies. Let's delve into this script to understand its structure and functionality.

The script commences with a script header, which is a crucial part of any Pine Script. It starts with the declaration `//@version=5`, indicating that the code is written in Pine Script version 5. The strategy is named "Tidy Little Robot" and is configured to overlay on the price chart, making its signals and actions visible to traders.

Technical indicators play a pivotal role in algorithmic trading, aiding in the analysis of market data. In this script, the Exponential Moving Average (EMA) with a 200-period setting is calculated based on the closing prices of the asset under consideration. This EMA acts as a reference point for trend analysis, a critical aspect of trading.

The script introduces flexibility by allowing users to choose between two trend detection rules. These rules, named "SMA50" and "SMA50, SMA200," enable traders to select their preferred method of identifying trends. This choice is fundamental in determining the direction of potential trades.

Candlestick patterns are a cornerstone of technical analysis in trading. The script incorporates various variables to define and detect these patterns, including engulfing patterns and doji patterns. These patterns are crucial for identifying potential entry and exit points in the strategy.

To enhance user experience, the script introduces customizable labeling of detected candlestick patterns on the chart. Users can select label colors, adding a personal touch to their trading environment.

The core of the strategy revolves around the detection of engulfing patterns, both bullish and bearish. These patterns are pivotal in determining when to initiate or close a trade. The script employs specific conditions to identify these patterns and generate alerts when they occur.

Following pattern detection, the script proceeds to the heart of the strategy – entry and exit conditions. It defines the logic for entering long or short positions based on the detected candlestick patterns and the relationship between the current price and the EMA200.

Effective risk management is a cornerstone of successful trading. To address this, the script includes logic for setting stop loss and take profit levels. These levels are essential for controlling potential losses and locking in profits at predefined levels.

In summary, "Tidy Little Robot" is a comprehensive algorithmic trading strategy developed in Pine Script 5. It seamlessly combines trend analysis, candlestick pattern recognition, and risk management to execute precise trading decisions. This script offers traders, whether experienced or novice, a versatile framework for creating and testing trading strategies within the TradingView platform.

The Pine Script code for "Dirty Little Robot"

```
//@version=5
strategy("Dirty Little Robot", overlay = true)

//RSI
rsi_length = 7
rsi_value = ta.rsi(close, rsi_length)

//Candlestick Patterns
C_EngulfingBullish () =>
    close[1] < open[1] and
    open < close and
    close > open[1] and
    open < close[1]

C_EngulfingBearish () =>
    close[1] > open[1] and
    open > close and
    close < open[1] and
    open > close[1]

// Strategy Entry
long_condition = C_EngulfingBullish and rsi_value < 30
short_condition = C_EngulfingBearish and rsi_value > 70

if (long_condition)
    strategy.entry("Long", strategy.long)
if (short_condition)
    strategy.entry("Short", strategy.short)
```

```
//Strategy exit
stopLossPercent = 0.004
takeProfitPercent = stopLossPercent * 3

var float stopLossPrice = na
var float takeProfitPrice = na

if (strategy.position_size > 0)
    stopLossPrice := strategy.position_avg_price * (1 - stopLossPercent)
    takeProfitPrice := strategy.position_avg_price * (1 + takeProfitPercent)
    strategy.exit("Take Profit/Stop Loss", stop = stopLossPrice, limit = takeProfitPrice)

if (strategy.position_size < 0)
    stopLossPrice := strategy.position_avg_price * (1 + stopLossPercent)
    takeProfitPrice := strategy.position_avg_price * (1 - takeProfitPercent)
    strategy.exit("Take Profit/Stop Loss", stop = stopLossPrice, limit = takeProfitPrice)
```

Interpreting the Pine Script for "Dirty Little Robot"

Algorithmic trading is an intricate field that involves creating trading strategies governed by computer code. The Pine Script presented here, named "Dirty Little Robot," is an example of such a strategy written in Pine Script 5, specifically tailored for the TradingView platform. This code is designed to execute precise trading strategies based on the Relative Strength Index (RSI) and candlestick pattern analysis.

The script begins with the essential script header, denoting its compatibility with Pine Script version 5 and defining the strategy's name, "Dirty Little Robot." Additionally, it is configured to overlay on the price chart, allowing traders to visualize its signals and actions. Here, the script calculates the RSI with a period of 7 based on the closing prices of the asset. RSI is a momentum oscillator that is frequently used in trading to identify overbought and oversold conditions.

In the script, variables are defined and users can choose between two trend detection rules: one based on a Simple Moving Average (SMA) with a 50-period setting and another based on a combination of SMA50 and SMA200. These rules determine whether the market is in an upward or downward trend. The script then prepares variables and conditions for analyzing candlestick patterns, including engulfing patterns and doji patterns. These patterns are valuable for identifying potential entry and exit points in trading.

To enhance user experience, the script labels detected candlestick patterns on the chart. Users can customize the colors of these labels, providing a visual aid for pattern recognition. Engulfing patterns, both bullish and bearish, are detected based on specific conditions. Engulfing patterns are significant because they often signal potential reversals in market direction.

The heart of the script lies in defining conditions for entering long or short positions. Long positions are taken when a bullish engulfing pattern is detected, and the RSI is below 30, indicating potential oversold conditions. Conversely, short positions are taken when a bearish engulfing pattern is detected, and the RSI is above 70, signaling potential overbought conditions.

Effective risk management is a crucial consideration in trading. The script incorporates risk management by setting stop loss and take profit levels, helping to control potential losses and secure profits at predefined thresholds.

In summary, "Dirty Little Robot" is a sophisticated algorithmic trading strategy developed in Pine Script 5. It combines RSI analysis, candlestick pattern recognition, and risk management to make informed trading decisions. Whether you are an experienced trader or a beginner, this script provides a versatile framework for creating and testing trading strategies within the TradingView platform.

Comparison between TLR and DLR

Strategy Approach

TLR takes a trend-following approach, capitalizing on bullish and bearish engulfing patterns in combination with the EMA200 (Exponential Moving Average with a 200-period setting). It aims to identify and ride existing trends, making it well-suited for strong, persistent market movements. TLR's primary goal is to capture profits as long as the prevailing trend continues.

DLR, on the other hand, follows a reversal trading strategy. It also employs the same engulfing patterns but incorporates the Relative Strength Index (RSI) for additional confirmation. DLR focuses on identifying potential market reversals, aiming to profit from significant price shifts. It thrives in volatile or ranging markets where momentum may shift rapidly.

Key Indicators

TLR's core indicators are the EMA200 and engulfing candlestick patterns. The interplay between these factors guides its trading decisions. The EMA200 serves as a trend filter, and engulfing patterns act as signals to enter or exit positions.

DLR incorporates the RSI, a momentum oscillator, which provides insights into overbought and oversold market conditions. Combining RSI readings with engulfing patterns allows DLR to gauge not only potential reversals but also the strength of such reversals.

Risk Management

Both TLR and DLR implement robust risk management strategies. They share a common risk-reward ratio of 3, maintaining a take profit target of 1.2% and a stop loss level of 0.4%. These predefined levels help mitigate potential losses and lock in profits when trades move in their favor.

Asset and Timeframe

Both TLR and DLR are tested on the BTC/USDT Perpetual asset, a popular cryptocurrency trading pair, via the Pionex exchange. They both operate within a 1-minute timeframe, which allows for quick decision-making and execution in a fast-paced cryptocurrency market.

Backtesting Results

In backtesting, TLR achieved notable results over an 11-day period, with a net profit of 3.03%. It executed 38 total closed trades and maintained a win rate of 39.47%. The profit factor, a key performance metric, was 1.375, indicating favorable risk-reward dynamics. The average trade size was $0.08, and the ratio of average win to average loss was 2.1.

DLR, during the same 11-day backtesting period, secured a net profit of 0.78%. It executed 8 total closed trades with a win rate of 37.5%. DLR demonstrated a profit factor of 1.38, an average trade size of $0.10, and a favorable average win-to-loss ratio of 2.3.

Strategy Implementation

Both TLR and DLR share the same platform, the Pionex exchange. Their implementation is facilitated by integrating trading alerts generated on the TradingView platform into Pionex via webhooks. This integration ensures seamless execution of their respective strategies.

Overall Approach

In summary, TLR and DLR offer distinct trading approaches suitable for different market conditions and trader preferences. TLR excels in trend-following scenarios, aiming to capitalize on established trends. In contrast, DLR is tailored for identifying reversals, making it advantageous in volatile markets or when momentum shifts are anticipated.

The choice between TLR and DLR depends on several factors, including market conditions, trader risk tolerance, and a preference for either trend or reversal trading. It's important to consider that past backtesting results do not guarantee future performance, and both strategies should be adapted to evolving market dynamics. Traders should thoroughly evaluate each algorithm's strengths and weaknesses before deploying them in live trading environments.

Discussions and Limitations

Discussion

TLR, the Tidy Little Robot, has demonstrated notable performance in backtesting, particularly in trending markets. Its strategy, which relies on bullish and bearish engulfing patterns in conjunction with the EMA200 indicator, yielded a net profit of 3.03% over an 11-day period.

The win rate of 39.47% and a profit factor of 1.375 suggest that TLR has the potential to generate consistent profits in markets characterized by sustained trends.

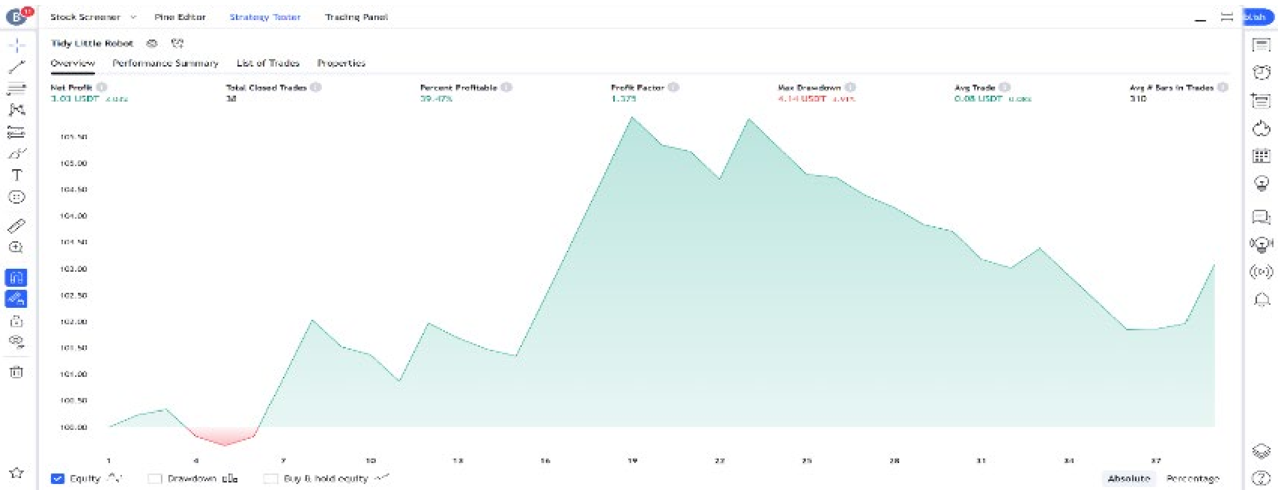

Figure 1. Strategy Results for TLR

Figure 2. Long Trade Made by TLR

Interpreting TLR's performance reveals its ability to effectively identify and capitalize on established trends. By using engulfing patterns as entry and exit signals, TLR adheres to a systematic approach that aligns with its trend-following strategy. This interpretation highlights its suitability for traders seeking to participate in extended price movements.

DLR, the Dirty Little Robot, employs a reversal trading strategy that combines bullish and bearish engulfing patterns with the RSI indicator. In backtesting, DLR achieved a net profit of 0.78% over the same 11-day period, with a win rate of 37.5% and a profit factor of 1.38. DLR's performance underscores its ability to identify potential trend reversals effectively.

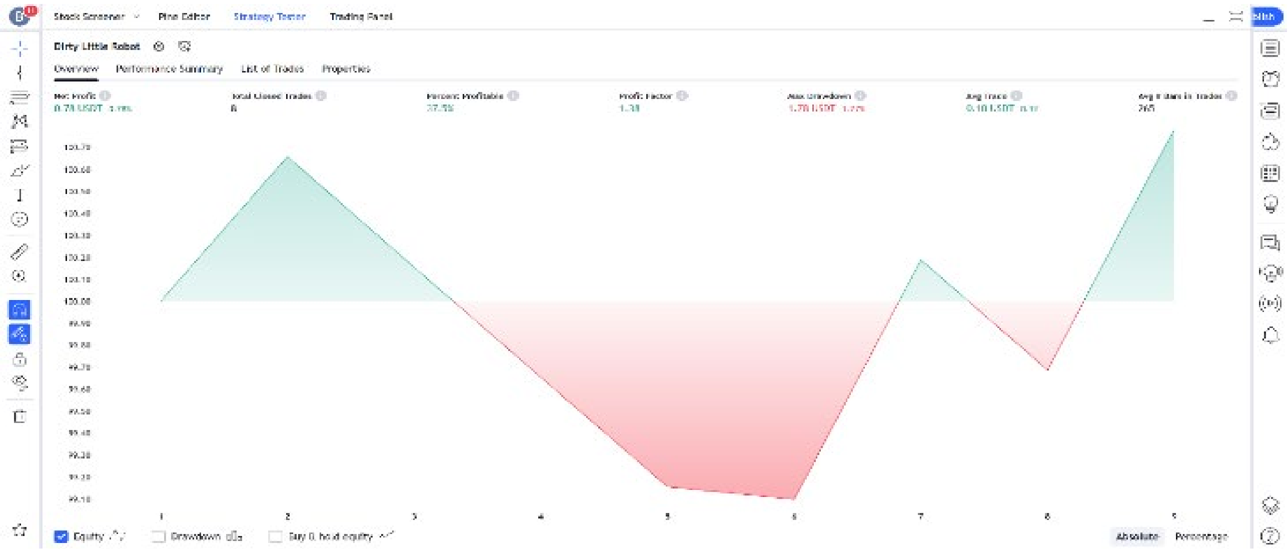

Figure 3. Strategy Results for DLR

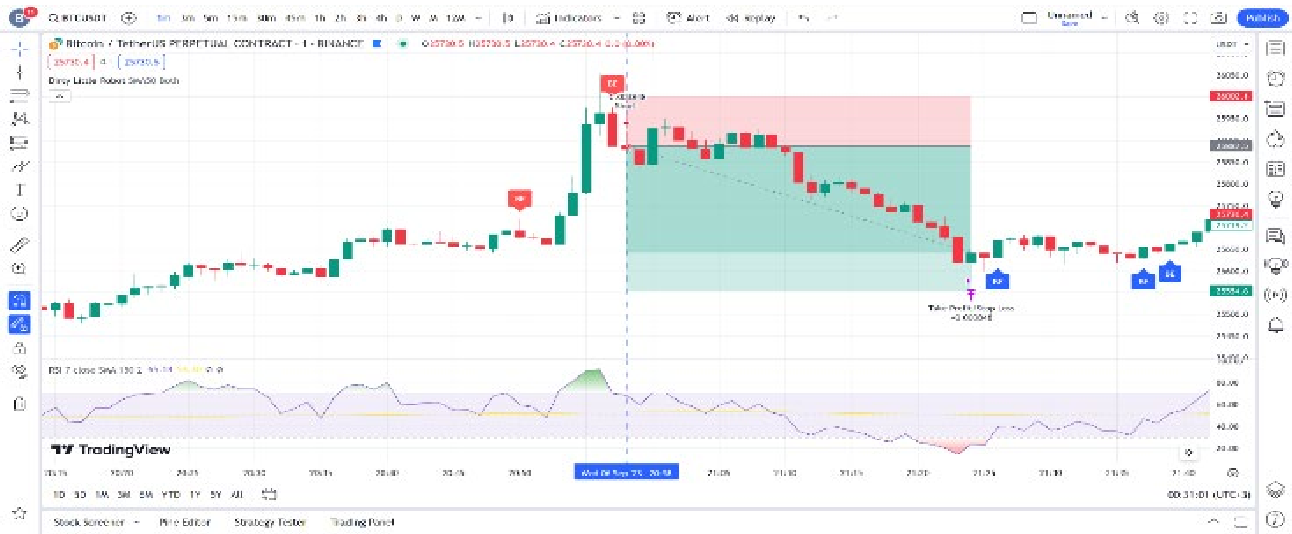

Figure 4. Short Trade Made by DLR

The interpretation of DLR's performance emphasizes its role as a valuable tool for traders operating in volatile or ranging markets. By combining engulfing patterns with RSI readings, DLR aims to capture price shifts that occur when trends lose momentum or reverse abruptly. This approach can be advantageous in markets prone to sudden fluctuations and reversals.

The implications of TLR and DLR's performance highlight their adaptability to a range of market conditions. TLR's strength lies in trending markets, where it can capitalize on established trends effectively. Conversely, DLR's focus on reversals positions it as a valuable asset in markets marked by volatility and abrupt price movements. Traders can leverage these algobots to align with the ever-changing dynamics of financial markets. Both TLR and DLR maintain a consistent risk management approach, featuring a predefined risk-reward ratio of 3, a take profit of 1.2%, and a stop loss of 0.4%. The systematic execution of these parameters minimizes the emotional biases often associated with human trading. The implications are that traders can rely on these rules to manage risk effectively and maintain capital protection. While TLR and DLR have proven their effectiveness within their respective strategies, diversification remains a valuable consideration. Relying solely on one algobot or strategy may limit opportunities in varying market conditions. Traders may benefit from exploring the combination of multiple algobots or trading strategies to create a diversified portfolio that adapts to changing market dynamics. An implication for further enhancement of algobot performance is the incorporation of intermarket analysis. This approach considers relationships between different financial markets, providing insights that can enhance the accuracy of trading signals. By broadening the scope of data sources, algobots can be more effective in capturing opportunities across various asset classes.

Limitations

One limitation of the study is the relatively short backtesting period of 11 days. While the results provide valuable insights, they may not fully represent the algobots' long-term capabilities. Historical data used for backtesting may not entirely mirror future market conditions, introducing an element of uncertainty. There is a risk of overfitting, where algobots are excessively tailored to historical data, potentially leading to suboptimal performance in live markets. Market dynamics evolve, and certain strategies may lose effectiveness. Continuous monitoring and adaptation are essential to ensure algobots remain aligned with prevailing market conditions.

External factors, such as sudden news events or market manipulations, can influence algobot performance. These external influences are challenging to predict and incorporate into algorithms, introducing limitations on the algobots' effectiveness. Execution speed and market liquidity can impact algobot performance, particularly in high-frequency trading scenarios. Delays in executing trades or trading illiquid assets can affect outcomes. Traders need to ensure their chosen trading platform provides the necessary infrastructure for efficient execution.

The discussion, interpretation, and limitations of TLR and DLR algobots offer valuable insights for traders considering algorithmic trading strategies. These algobots demonstrate strengths in systematic trading, risk management, and adaptability to diverse market conditions. However, traders should remain mindful of the limitations, including short backtesting periods, the risk of overfitting, and external factors' influence. Algorithmic trading, when employed thoughtfully and continuously adapted, can provide a robust framework for navigating dynamic financial markets effectively.

Conclusion

In this comprehensive exploration of algorithmic trading, we embarked on a journey to dissect the performance, implications, interpretations, and limitations of two remarkable algobots: Tidy Little Robot (TLR) and Dirty Little Robot (DLR). Through rigorous backtesting and analysis, we have unveiled valuable insights into the world of algorithmic trading. This conclusion chapter synthesizes the results obtained from our study, summarizes the methodology, and addresses the discussions, interpretations, limitations, and future research opportunities surrounding these algobots.

Our examination of TLR and DLR's performance revealed their distinct yet complementary strengths. TLR, with its trend-following strategy, demonstrated impressive profitability, boasting a net profit of 3.03% during an 11-day backtesting period. Its affinity for capitalizing on established trends, especially when coupled with bullish and bearish engulfing patterns and the EMA200 indicator, positions TLR as a potent tool in capturing prolonged price movements. DLR, on the other hand, exhibited prowess in volatile or ranging markets, showcasing a net profit of 0.78%. Its reversal trading strategy, incorporating engulfing patterns and the RSI indicator, enables DLR to effectively identify potential trend reversals. These results underscore the adaptability and applicability of these algobots across diverse market conditions.

Our methodology, meticulously designed and executed, forms the bedrock of our study's credibility. We developed TLR and DLR using Pine Script 5 on the TradingView platform, implemented them on the Pionex exchange, and operated with a risk-reward ratio of 3, a take profit of 1.2%, and a stop loss of 0.4%. Our backtesting took place over an 11-day period, allowing us to evaluate their performance systematically.

The discussions surrounding TLR and DLR emphasized their systematic approach to trading, disciplined risk management, and adaptability to varying market conditions. These algobots enable traders to operate with precision, removing emotional biases and promoting consistency in decision-making. Moreover, our interpretations underscored the importance of understanding the logic behind the generated signals, empowering traders to optimize their strategies effectively. Nonetheless, we must acknowledge the limitations inherent in algorithmic trading. The relatively short backtesting period introduces uncertainty, and overfitting remains a concern when tailoring algobots to historical data. External factors, execution speed, and market liquidity can also impact algobot performance. Traders should navigate these limitations with caution.

As the landscape of algorithmic trading evolves, numerous avenues for future research emerge. Expanding the scope of backtesting to encompass more extensive historical data can provide a more comprehensive understanding of algobot performance.

Additionally, delving into advanced risk management techniques and incorporating machine learning algorithms for enhanced pattern recognition represent promising directions.

The integration of real-time data feeds and sentiment analysis into algobot strategies could further refine their ability to respond to evolving market dynamics. Furthermore, exploring the incorporation of fundamental analysis alongside technical indicators could yield insightful strategies. Finally, investigating the potential of intermarket analysis to enhance algobot performance should not be overlooked.

In conclusion, TLR and DLR algobots represent formidable tools for traders seeking to navigate the complexities of financial markets. Their systematic approach to trading, disciplined risk management, and adaptability to diverse market conditions offer a glimpse into the future of algorithmic trading. While limitations exist, they serve as stepping stones for further refinement and innovation.

This study underscores the importance of continuous adaptation and the necessity for traders to remain vigilant in the ever-changing world of algorithmic trading. As technology advances and research progresses, the potential of algobots to optimize trading strategies and deliver consistent results becomes increasingly promising. By harnessing the power of algorithmic trading intelligently and thoughtfully, traders can embark on a journey of financial success in the dynamic realm of the financial markets.

References

Bacidore, J.M. (2017). *Algorithmic Trading : A Practitioner's Guide*. New York, Tbg Press, 2020. Chan, Ernest P. *Machine Trading : Deploying Computer Algorithms to Conquer the Markets*. Hoboken, New Jersey: Wiley.

Chan, E. (2021). *Quantitative Trading : How to Build Your Own Algorithmic Trading Business*, Hoboken, John Wiley & Sons Inc.

Crypto Trading Bot. (n.a.). Pionex Blog. Retrieved from www.pionex.com/blog/

Davey, K.J. (2014). *Building Winning Algorithmic Trading Systems : A Trader's Journey from Data Mining to Monte Carlo Simulation to Live Trading*. Hoboken, New Jersey: Wiley.

Dedu, S., & Şerban, F. (2015). Modeling financial data using risk measures with interval analysis approach. *Procedia Economics and Finance 22,* 610-617. https://doi.org/10.1016/S2212-5671(15)00271-3

Donadio, S. & Ghosh, S. (2019). *Learn Algorithmic Trading : Build and Deploy Algorithmic Trading Systems and Strategies Using Python and Advanced Data Analysis*. Birmingham: Packt Publishing Ltd.

Goodman, G. (2019). *The Crypto Trader : How Anyone Can Make Money Trading Bitcoin and Other Cryptocurrencies*. Harriman House.

Gray, S. (2019). *Algorithmic Trading: A Comprehensive Beginner's Guide to Learn Algorithmic Training from A-Z*. Independently Published.

Hilpisch, Y. (2021). *PYTHON for ALGORITHMIC TRADING: From Idea to Cloud Deployment*. USA: S.L., O'reilly Media, Inc.

Höppel, S. (2014). *High Frequency Trading: Economic Necessity or Threat to the Economy?* Anchor Academic Publishing.

Isichenko, M. (2021) *Quantitative Portfolio Management: The Art and Science of Statistical Arbitrage*. Hoboken, New Jersey: John Wiley & Sons, Inc.

Kaufman, P.J. (2016). *A Guide to Creating a Successful Algorithmic Trading Strategy*. Hoboken, New Jersey, John Wiley & Sons, Inc.

Kelleher, J.D. & Tierney, B. (2018). *Data Science*. Cambridge, Massachusetts; London, England: The Mit Press.

Kendall, K. (2007). *Electronic and Algorithmic Trading Technology: The Complete Guide*. Burlington, Mass.; London: Academic Press.

Markowitz, H. (1952). Portfolio selection, *Journal of Finance, 91,* 7–77. https://doi.org/10.2307/2975974

Mattli, W. (2018). *Global Algorithmic Capital Markets : High Frequency Trading, Dark Pools, and Regulatory Challenges*. Oxford: Oxford University Press.

Michalowski, W. & Ogryczak, W. (2001). Extending the MAD portfolio optimization model to incorporate downside risk aversion. *Journal of Naval Research Logistics, 48*(3), 185–200. https://doi.org/10.1002/nav.1

Mishkin, F.S. (2019). *The Economics of Money, Banking, and Financial Markets*. New York: Pearson.

Nolte, I. (2015). *High Frequency Trading and Limit Order Book Dynamics*. London: Routledge.

Pardo, R. (2011). *The Evaluation and Optimization of Trading Strategies*. John Wiley & Sons.

Scopino, G.A. (2020). *Algo Bots and the Law: Technology, Automation and the Regulation of Futures and Other Derivatives*. Cambridge, United Kingdom; New York: Cambridge University Press.

Serban, F., Stefanescu, V., & Ferrara, M. (2013). Portfolio optimization in the framework Mean-Variance-VaR. *Economic Computation and Economic Cybernetics Studies and Research, 1,* 61-79.

Sharpe, W.F. (1963), A Simplified model for portfolio analysis. *Journal of Management science, 9*(1), 277–293.

Tatsat, H. (2019). *Machine Learning and Data Science Blueprints for Finance*. O'Reilly Media, Inc.

Tradingview.com. (n.a.). Pine Script® v5 User Manual. Retrieved from www.tradingview.com/pine-script-docs/en/v5/index.html

Velu, R.P. (2020). *Algorithmic Trading and Quantitative Strategies*. Boca Raton; London; New York: Chapman Et Hall/Crc.

Winston, P.H. (2019). *Artificial Intelligence*. Addison-Wesley.