Deep Reinforcement Learning for Stock Trading

A Non-Technical Case Study

Kordel K. France
Goldilox Autonomy, LLC
Dallas, TX, USA
kfrance8@jh.edu

Abstract**—***The trading of securities on the stock market has been touched by just about every algorithm feasible. Machine learning and artificial intelligence (AI) have largely accelerated both the use and efficacy of algorithmic trading of these securities. This paper is not an attempt to postulate yet another strategy for "beating the market", but rather a highlight of some common pitfalls among many AI-based approaches to algorithmic trading. We introduce why these methods and their assumptions fail and offer ways to reconcile them. In addition, we illustrate why leveraging a branch of AI called reinforcement learning can address these failures and may have a higher probability of generating consistent returns. Finally, we show results from our own fund utilizing such a method over a 3-month period with these points of failure rectified. The points made in this paper are based on research performed by and data acquired from the Goldilox Fund by Goldilox Autonomy, LLC and may not necessarily be representative of all practitioners of algorithmic trading.*

I. Introduction

It has been analyzed that one in five day traders in the stock market actually turn consistent profits [8]. To help raise these odds, some traders turn to algorithms to provide a higher level of automation over their strategy. Algorithmic trading is—no pun intended—a trade that has been sharply refined by artificial intelligence (AI). The stock market, although a deterministic system with clear cause-and-effect structure, is often discounted as randomness due to its immense complexity. Many modern hedge funds and day-traders now make use of neural networks and advanced machine learning forecasting methods as an attempt to gain an edge in making profitable trades, and some of them even achieve it [6]. However, if AI truly is the advertised end-all-be-all for finding order within chaos, why have there not been more headlines about millionaire AI-engineer-traders who have seemingly beat the market? We suspect some reasons of this to be the manner in which these algorithms are deployed, the assumptions used to define these algorithms, and the domain in which they operate.

In this paper, we explore why some approaches with machine learning still fail to achieve consistency within the market and why some algorithms are more fundamentally fit to provide the intended outcome over others. We also define some helpful constraints that can improve these approaches. Additionally, we elaborate on why reinforcement learning has untapped potential within the market and briefly define what such an algorithm looks like. Finally, we evaluate a case study performed by Goldilox Autonomy that employed these constraints over a three-month period for a real fund using deep reinforcement learning. We analyze these results and show how the fund achieved superior performance against the NASDAQ, DJI, and S&P 500 comparator indices.

The spirit of this paper is non-technical by design. Its intention is to communicate new research at Goldilox Autonomy and to increase awareness about the current state-of-the-art of algorithmic trading to a large general audience. We refrain from extreme technicalities in order to accommodate such an audience, but we assume some familiarity with AI upon the reader. Although some formulas are referenced, we do so only as a gesture of completeness.

The hard truth is, nobody knows how to predict the market and this paper is not another attempt claiming "how" to—we are simply communicating how AI may provide a slightly better edge with a few alterations to how it is used in algorithmic trading.

II. Background

Technical Analysis (TA) is the most common methodology used in algorithmic trading. TA is a set of statistical (not machine learning) tools that effectively assess the stock's current price relative to its past performance. Some examples of these tools are *Bollinger Bands, Moving Average Convergence and Divergence (MACD), Relative Strength Index (RSI),* and *Exponential Moving Average (EMA)*. Algorithms that employ these tools are largely rule-based and do not leverage AI in order to implement them. In both long-term and near-term trading, a meticulous application of using TA can actually yield reasonable returns[1] [4]. Leveraging time series methods in cohesion with TA begins to incorporate modern machine learning methods and starts crossing over into AI. However, changing market conditions can quickly invalidate any successful strategy and overfit data. In other words, strict TA algorithms are not dynamic enough to generalize over all market conditions.

Enter AI. AI is fundamentally dynamic and can exploit scenarios where TA fails. Neural networks are one of the biggest exemplars of AI being used in algorithmic trading. A neural network is, as the name suggests, a large network of small approximation functions that altogether can compute the solution to a complex problem by parallel processing small subsets of the problem. Their design was inspired by the way in which neurons are arranged and compute information within the human brain, and they are best used in classification or prediction tasks. Many traders use a flavor of these networks called *recurrent neural networks (RNN)* in order to provide forecasting insight on how a stock is expected to behave based on past performance data. *Convolutional neural networks (CNN)* have also been used to classify a "winning" or "losing" pattern within a given set of market data, among other tasks[2] [2].

Deep reinforcement learning is another domain of AI that has experienced use by traders within the market. While perhaps more accurately viewed as a form of adaptive control, it is an algorithm that shows merit in a variety of problems. It has, based on our research, experienced the least amount of attention of machine learning-based approaches within algorithmic trading, but is a very prudent tactic and obviously the focal point of this paper. While neural networks are trained using a *supervised* strategy in which batches of labeled data points are shown to an algorithm repeatedly in order to teach it the true meaning of those data points, a

[1] Goldilox Autonomy has also shown that using TA tools only can return reasonable profits when coordinated through algorithmic trading. It has also shown that the same TA algorithm with the same parameters that impressively beats the market in certain conditions, horribly fails in others. Results from this fund may be provided upon reasonable request to the author.

[2] Goldilox Autonomy has also shown the feasibility of a CNN in detecting such patterns by leveraging an algorithm on one of its own funds. Results were mixed but may be provided upon reasonable request to the author.

reinforcement learning agent is trained with a slightly different methodology. A reinforcement learning agent learns to dynamically adjust to observed data by facilitating continuous feedback in order to maximize some reward. In other words, the agent selects its actions in such a manner that positions itself for the best possible future return. Yang, et al. [7] demonstrate a potentially very effective approach to algorithmic trading through the use of multiple reinforcement learning algorithms together in an ensemble strategy.

III. Fooled by Randomness

There is an extravagant flaw in treating stock market movements as random. Albeit extremely complex, the market is fundamentally non-chaotic, driven by laws of supply and demand within the economy. Too many approaches define the distribution of the market events as Gaussian which undoubtedly leads to underperforming models because the model is based on false assumptions from the start [9]. Some market events that occur with astonishing frequency are deemed as "one-in-a-billion" events by the Gaussian model. One requirement of a Gaussian (or Normal) distribution is known as the *iid* principle—the data must be independently and identically distributed. The prices of some stocks are contingent upon the prices of others. Additionally, stock prices are partially influenced by shareholder signals within the company. When stakeholders with a significant interest in a company sell their interest within that company, or, conversely, a stakeholder acquires a significant share of a company, the stock price of that company changes as a consequence of the drastic change in supply. This can indicate large buying or selling signals to other shareholders [5] triggering algorithmic funds to react, and the effects compound. In no way are these events independent or random. *Some* aspects of the market may be treated as random to make statistical modeling easier, but the general movement of the stock should never be. Just because one cannot identify the drivers behind a company's current price does not mean that that price is a consequence of "random" events.

IV. Overcoming Overfitting

A machine learning algorithm is only as good as the data it is trained on. While the above algorithms each show merit in certain market scenarios, strong evidence exists to show that any one method alone cannot achieve superior performance over *all* market scenarios with current practices [3]. This is partially due to the fundamentals behind how the algorithm works, but, we believe, more largely due to the disparity between the data the algorithm is trained on and the data it is evaluated on.

For example, it seems common for a lot of traders to train one machine learning model and refine it to achieve the desired performance over past data *for one stock*. Then, they may leverage this model in the field over this same stock, but also apply it to a stock completely adjacent to the training stock only to realize completely different results. This highlights a major issue in machine learning known as *overfitting*. The model fails to achieve expected performance on data it was not trained on. One cannot take a model trained on Tesla (TSLA) data and expect it to perform identically on Delta Airlines (DAL). The two companies are in completely adjacent industries and may not exude the same underlying patterns or volatility. Quite frankly, models for companies that experience extremely high volatility such as Tesla, AMC (AMC), and Gamestop (GME), may not generalize well enough on any other company due to their unique price fluctuations [2].

Conversely, securities that reside within the same sector will follow relatively similar trends both for long-term and short-term intervals. By extension, the same model across those stocks should provide relatively similar returns, or returns that are not generally statistically different. Stocks within the same sector are probable to experience the same underlying price movements, and investors of one security are likely to invest in securities within a similar domain. These companies will also undergo similar supply chain effects and resource constraints and launch similar products in general. There are exceptions to this, such as consequences of executive misconduct, product recalls, and other company-specific events, but the fundamental relationship generally holds.

Furthermore, let us assume the above conditions are reconciled in the design of the trading algorithm. Let us assume that the designer wants to execute a trading strategy over a three-month timeline. A stock's current three-month history is not reflective of any other three-month interval within the stock's history—a given interval is simply a statistical sample from the overall population that is not necessarily a good representation of that population in its entirety. Goldilox explores this effect in its study by assessing its reinforcement learning algorithm over thirteen different companies lying exclusively within the semiconductor industry and over a changing time interval.

V. Case Study

In exemplification of the above, Goldilox Autonomy conducted a case study to show there is strong evidence to recognize the effects of poor assumptions when utilizing machine learning models in the market. We elaborate on the experimental design and results below.

A. Data

We selected thirteen stocks within the semiconductor industry of the technology sector that were tradable on United States markets. These stocks are shown below in Table 1. The period of evaluation began on 1 September 2021 and ended 1 December 2021. These companies were evaluated to ensure they maintained similar product lines, but not so identical that we could not evaluate the boundaries of our model. It seems more common in literature to communicate results of a stock trading algorithm that can make money in a bull market. However, of particular research interest to Goldilox Autonomy is the ability of an algorithm to actually make money in a bear market. As a result, in order to further establish the boundaries of our model, it was important to select an industry that was subject to a high probability of downside potential. We found evidence to support this downside potential within the semiconductor industry due to the global supply chain pressure experienced in 2021.

Table 1 - Case Study Portfolio

Company	Ticker
Advanced Micro Devices, Inc.	AMD
Nvidia Corporation	NVDA
Micron Technology, Inc.	MU
Intel Corporation	INTC
Taiwan Semiconductor Mfg. Co. Ltd.	TSM
Xilinx, Inc.	XLNX
STMicroelectronics N.V.	STM
Microchip Technology, Inc.	MCHP
Texas Instruments, Inc.	TXN
QUALCOMM, Inc.	QCOM
Analog Devices, Inc.	ADI
Semtech Corporation	SMTC

B. Comparators

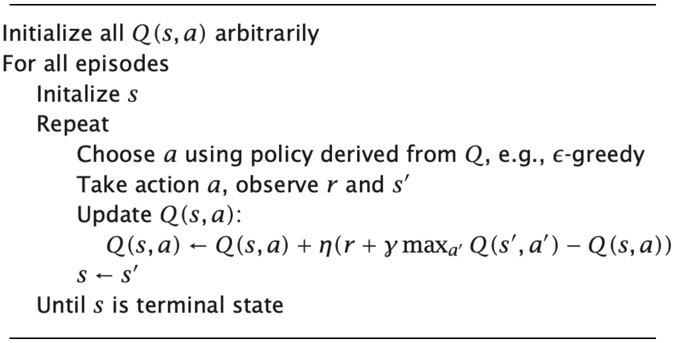
In order to evaluate the efficacy of our algorithm design, we evaluate its performance against three comparator indices, namely the Dow Jones Industrial Average (DJI), the S&P 500 (SPX), and the NASDAQ (IXIC)[3]. These indices were

[3] The NASDAQ is more heavily weighted by technology companies than the other two indices, so comparison of our semiconductor fund to its performance is somewhat biased.

evaluated over the same period of 1 September through 1 December 2021. We select these three indices as comparators because they give good indications of how the market is trending in general.

C. Reinforcement Learning Algorithm

For this case study, we centered our algorithm around reinforcement learning. Specifically, we constructed a deep reinforcement learning algorithm called Q-learning that leverages a seven-layer neural network to evaluate which action to take next. The algorithm for Q-learning is shown in Figure 1 [1].



```
Initialize all $Q(s,a)$ arbitrarily
For all episodes
    Initalize $s$
    Repeat
        Choose $a$ using policy derived from $Q$, e.g., $\epsilon$-greedy
        Take action $a$, observe $r$ and $s'$
        Update $Q(s,a)$:
            $Q(s,a) \leftarrow Q(s,a) + \eta(r + \gamma \max_{a'} Q(s',a') - Q(s,a))$
        $s \leftarrow s'$
    Until $s$ is terminal state
```

Figure 1 *- Pseudocode for Q-learning reinforcement learning algorithm.*

We initially trained our algorithm with hourly data acquired from the stock AMD and trained it over its last one month's worth of data. We retrained this algorithm daily such that today's algorithm leveraged data from yesterday with slightly new weights that determine when to buy, hold, and sell. In other words, the *nth* day of the fund used a model trained over hourly data accumulated between *n - 30* and *n - 1* days, a moving time interval.

Q-learning allows the algorithm some freedom to explore outside of its training data in order to find the best policy for buying, selling, and holding a stock. This is partially where reinforcement learning's advantages shine in the interpretation of market data. Any given interval of market data is unique and therefore its training data is unique, so there is a low probability that the training set will be entirely representative of the testing data [2]. Exploration helps with this in that it allows the algorithm to venture outside the training data space in attempt to find better relationships among data and transactions.

D. Broker

Our algorithm was linked to a Webull stock account and given authority to automatically buy and sell when it determined apporpriate. We disabled day-trading to allow for additional experimental control. In other words, the algorithm was only allowed to make a single transaction with any given stock per day—buying and selling the same stock within the same trading period was not allowed. As a follow-up research effort, Goldilox Autonomy is evaluating this same model while allowing for day-trading.

VI. Results

We evaluated the results of our algorithm over the last 3-month, 1-month, and 1-week periods against the three comparator indices. Full screenshots from the Webull trading account for each interval are located in the *Supplementary Material* section.

A. 3-Month Period

Our algorithm returned +10.20% over the entire 3-month period from 1 September—1 December 2021. In comparison, the S&P 500, NASDAQ, and Dow Jones Industrial Average indices each returned -2.38%, +1.79%, and -4.41% respectively.

B. 1-Month Period

Our algorithm returned +8.03% over the last 1-month period from 1 November—1 December 2021. In comparison, the S&P 500, NASDAQ, and Dow Jones Industrial Average indices each returned -1.01%, -0.67%, and -3.98% respectively.

C. 1-Week Period

Our algorithm returned +1.24% over the last 1-week period from 23 November—1 December 2021. Note that that this week included a federal holiday in which the market was closed, so we extend the interval to acquire an additional weekday and maintain 5 market days. In comparison, the S&P 500, NASDAQ, and Dow Jones Industrial Average indices each returned -2.64%, -1.80%, and -3.71% respectively.

The Sharpe ratio is a common metric used to show approximate performance of a fund. For this algorithm, the Sharpe ratios were computed utilizing each of DJI, S&P 500, and NASDAQ as the "risk-free rate". We elected not to use the Sharpe ratio as an indicator of performance for this study because it assumes that investment returns are normally distributed, which we do not yet have enough evidence to confirm. However, we included them as a means of *approximation* when comparing to other algorithmic trading funds.

Table 2 - Sharpe Ratios for Goldilox

Interval	S&P 500	NASDAQ	DJI
1 week	1.0862	1.1681	1.2128
1 month	2.5306	3.3430	2.9426
3 months	3.5216	3.2315	3.5797

VII. Discussion

The selection to use reinforcement learning is worth focusing on for a moment. In reinforcement learning, the algorithm learns to react to its environment and plan sequential steps toward a goal. In other machine learning approaches, say supervised learning in a recurrent neural network, the algorithm is essentially looking for sequences of various lengths that match a sequential pattern it has seen before. However, future performance may not resemble past price patterns, which can nullify the use of such pattern classifiers. The use of deep reinforcement learning allows us to treat the problem as a robotic control problem versus a pattern-matching problem. However, the success of our solution in this study may have been benefitted by good initial assumptions just as equally as our selection to use reinforcement learning. Additionally, the algorithm leveraged over this specific timeframe should be deployed against other timeframes to ensure the duplication of results within different markets is achievable.

As machine learning experts first and foremost, we at Goldilox do not have the expertise nor intuition of seasoned traders, and we do not necessarily care to acquire such. Our goal is to build the algorithm that becomes the expert, finding underlying relationships between price movements within the market and economic events that humans may overlook. In combination with the aforementioned constraints, deep reinforcement learning proves itself as a great candidate algorithm for paving an initial path toward discovering those relationships.

Training the algorithm on one stock over a moving interval proved to be an effective proxy for learning the other twelve similar stocks denoted in Table 1. This allowed us to establish control over the accuracy of our algorithm, but our goal is to expand the scope of this algorithm to generalize over stocks outside of the industry we selected. We will start with companies within the same sector (technology) and later expand to companies outside this sector.

VIII. Conclusion

An algorithmic-based trading strategy can largely increase the odds of earning consistent profits as a trader, but only with proper assumptions, statistical modeling, and algorithm selection. In this paper, we addressed some common mistakes made with machine learning in algorithmic trading and how to reconcile them with proper modeling. We also focused on why reinforcement learning may provide a slight edge in seeking profits and evaluated the setup of our own fund based on such an algorithm. Finally, we discussed the results of our fund and explain how proper initial assumptions may have helped achieve impressive performance.

References

1. Alpaydin, Etham. *Introduction to Machine Learning*. Fourth Edition. (2020). Massachusetts Institute of Technology.

2. Goodfellow, Ian; Bengio, Yoshua; Courville, Aaron. *Deep Learning.* (2016). Massachusetts Institute of Technology. p 147 -149, 525 – 527.

3. Frost, A. J., & Prechter, R. R. (2017). *Elliott Wave Principle: Key to Market Behavior.* New Classics Library.

4. Bacidore, J. M. (2020). *Algorithmic Trading: A Practitioner's Guide.* TBG Press.

5. Graham, B., & Dodd, D. L. (2009). *Security Analysis*. McGraw-Hill.

6. Mandelbrot, B. B., & Hudson, R. L. (2006). *The (Mis)behavior of Markets: A Fractal View of Financial Turbulence.* Published by Basic Books.

7. Yang, Hongyang; et al. *Deep Reinforcement Learning for Automated Stock Trading: An Ensemble Strategy* (September 11, 2020). Available at SSRN: https://ssrn.com/abstract=3690996 or http://dx.doi.org/10.2139/ssrn.3690996

8. Smigel, L. (2019, October 17). *Algorithmic trading: Is it worth it?* Analyzing Alpha. Retrieved December 24, 2021, from https://analyzingalpha.com/algorithmic-trading-is-it-worth-it.

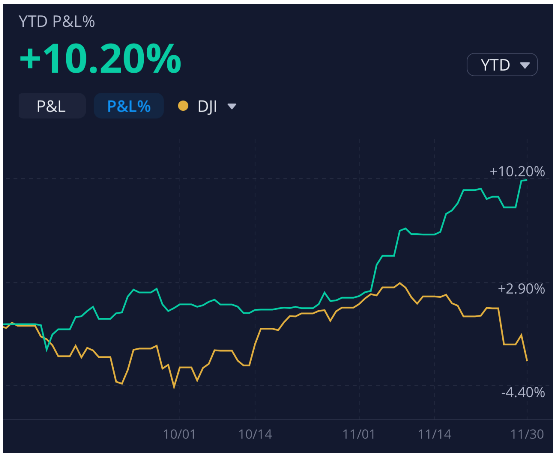
9. Taleb, N. N. (2016). *The Black Swan: The Impact of the Highly Improbable.* Random House.

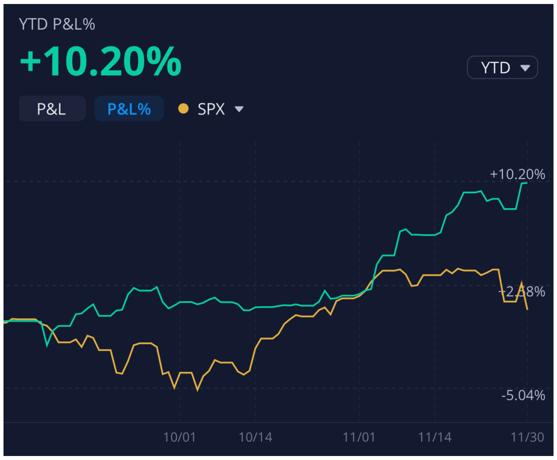
Supplementary Material

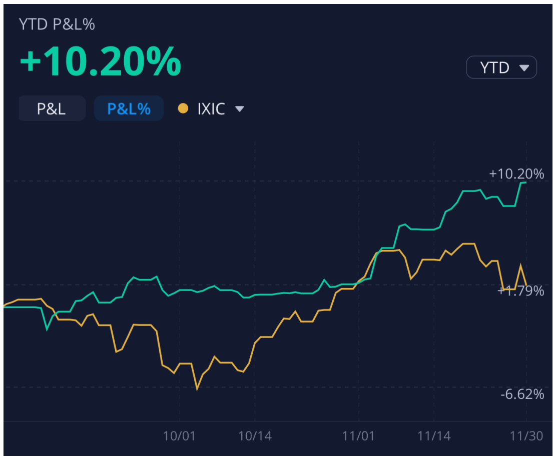
We provide screenshots from the Webull brokerage application that show the performance of our algorithm over the specified time interval.

A. 3-Month Period

Goldilox returned +10.20% over the entire 3-month period from 1 September—1 December 2021.

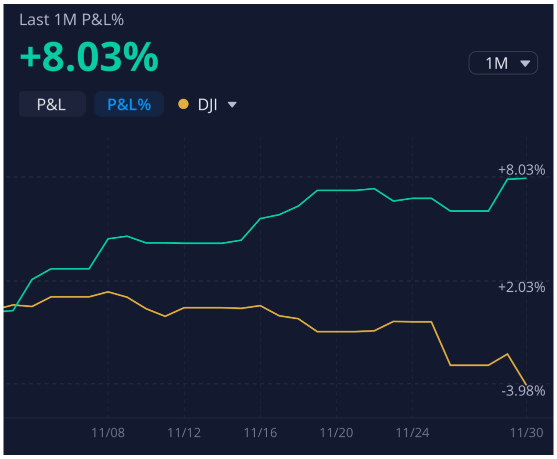


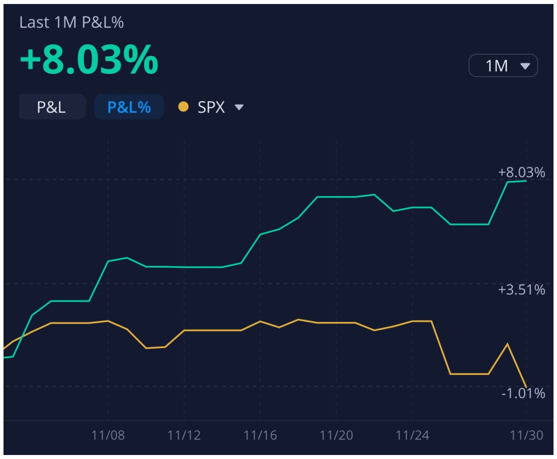


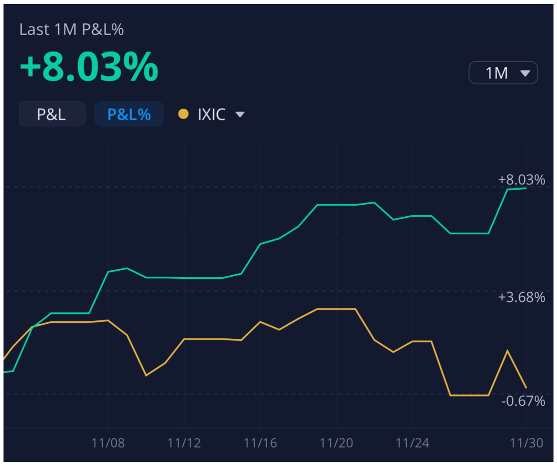


B. 1-Month Period

Goldilox returned +8.03% over the last 1-month period from 1 November—1 December 2021.

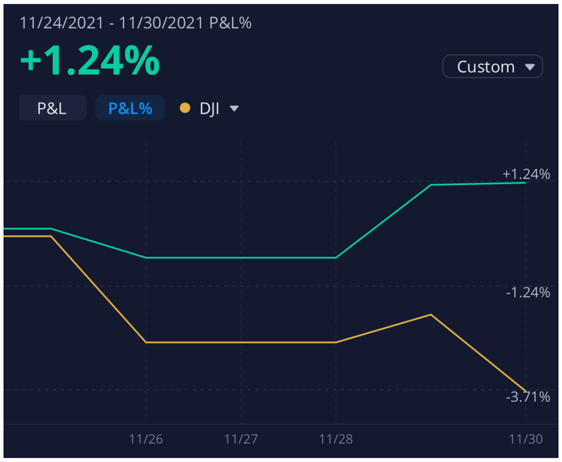


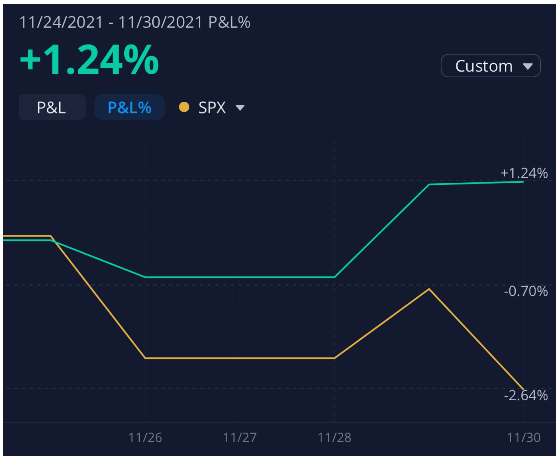


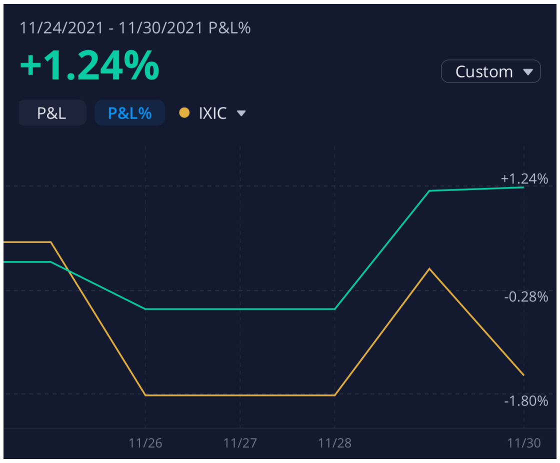


C. 1-Week Period

Goldilox returned +1.24% over the last 1-week period from 24 November—1 December 2021.


11/24/2021 - 11/30/2021 P&L%
Custom
P&L
P&L%
● DJI
11/26
11/27
11/28
11/30


11/24/2021 - 11/30/2021 P&L%
Custom
P&L
P&L%
● SPX
11/26
11/27
11/28
11/30


11/24/2021 - 11/30/2021 P&L%
Custom
P&L
P&L%
● IXIC
11/26
11/27
11/28
11/30