

Algorithmic Trading

Presented to the S. Daniel Abraham Honors Program
in Partial Fulfillment of the
Requirements for Completion of the Program

Stern College for Women

Yeshiva University

May 19, 2022

Talia Leitner

Mentor: Professor Lawrence Teitelman, Computer Science

Abstract:

This paper gives an overview of algorithmic trading. It is intended to inform interested traders of the value of incorporating algorithms into their trading strategies. They must also be warned of the dangers of implementing faulty algorithms. This paper highlights the usefulness of categorizing algorithms. An overview of the most popular algorithms is also given. Methods for comparing the performance of different algorithms in trading specific stocks is discussed. The metrics used to evaluate their performance is also mentioned. Challenges within the field of algorithmic trading are important for traders to consider when choosing which algorithms to implement in specific situations. Finally, the integration of machine learning into the field of algorithmic trading is discussed. This revelation has significantly broadened the scope of algorithmic trading. Machine learning now serves as an essential tool for traders to compete in the fast-paced markets of modern day.

Introduction:

Technology is integrated into all areas of life. The mid 1970s marked the beginning of the computerization of trading with the introduction of the designated order turnaround (DOT) system. This revolutionized the world of finance from the floor-based trading that existed in the New York Stock Exchange (NYSE) to the electronic trading that is now widely available. High frequency trading dominates the market and is the practice of trading large volumes of stocks in infinitesimal periods of time. Electronic trading enables computer systems to control decision making. Many of these programs are automated and do not require human input.¹ Trading algorithms have been developed to trade stocks at the most profitable prices and at the fastest rates. Machine learning algorithms have further progressed the financial industry by incorporating the insights of more data than ever before. These highly complex models result in more accurate market predictions. With the potential to earn sizable profits comes the risk of great loss. It is extremely important that traders are well versed in the different algorithms, performance metrics, and challenges that exist in the field. Understanding all these components will permit traders to have confidence in their trading algorithms.

An algorithm consists of a series of instructions intended for problem solving. Algorithms are typically associated with computer programs because they are responsible for their inner workings. In the realm of trading, algorithms are used for data processing, market forecasting, decision making, and order execution. Non-executing algorithms are intended to present parsed data to the trader to inform his or her decision making. The focus of this paper will be on executing algorithms. These programs are connected to the stock exchange and execute trades without any human input. Traders oversee these algorithms to ensure that they

¹ McGowan (2010)

produce the intended results.² As industry advances and markets change, the code behind these algorithms is updated accordingly. Human intervention is still critical to ensure that the algorithms behave in the desired manner.

A couple of challenges were presented with the writing of this paper. Firstly, many works on algorithmic trading are kept private. This is because financial firms do not wish to share their trading strategies with others. In addition to limiting the resources available for research, this makes it difficult to evaluate the different trading algorithms from a computational perspective. In the world of computer science efficiency is key. Computers are expected to get tasks done in the fastest, and therefore simplest, way possible. It is difficult to evaluate the complexity of an algorithm without its code being readily available. In consistency with most of the published literature on this topic, this paper will mainly outline methods for evaluating the algorithms from a financial perspective. To briefly explain the ideal features of an algorithm from a computational perspective, the focus is on time and space efficiency. This interest in time efficiency is shared with traders because they compete to make split-second decisions as new data is processed. Programmers have different coding styles which may affect the efficiency of their code. Additionally, the computing power of the machine used to analyze data and execute trades can make a substantial difference. The data sources chosen by the trader can also account for the differences in performance between algorithms. This is without even looking at the code behind a program. When evaluating the code, it is important to count the number of operations. The idea is to implement the algorithm with as few operations as possible.³ It is important for programmers to remember this when designing trading algorithms because this is essential for minimizing time complexity.

² Arnoldi (2016)

³ Rinker (2002)

Categorization of Trading Algorithms:

There exist multiple ways to categorize trading algorithms. The categories highlight the differences between algorithms and make it easier to select the most appropriate algorithms for specific cases. One determining factor is whether it is a single-order or list-based algorithm. The former refers to an algorithm designed to take considerations of a single order into account before executing a trade. The latter refers to an algorithm designed to take the considerations of multiple orders into account before executing any of the trades. Single-order algorithms can be fed lists of orders to execute; however, only list-based algorithms monitor the considerations of all the orders in relation to one another. Another distinguishing factor is whether it is a schedule-based, dynamic, or opportunistic algorithm. All algorithms split up orders and execute them in parts over time. Schedule-based algorithms are unique in that they follow a predefined schedule to trade the parts of an order. Dynamic algorithms are programmed to track market conditions to find the most suitable rate at which to trade the parts of an order. Similarly, opportunistic algorithms track market conditions to find opportune times to trade. The difference between dynamic and opportunistic algorithms is that the former is guaranteed to execute all parts of an order, even if at a slow rate. The latter may only initiate a trade if an opportunity in the market arises.⁴

An additional way to categorize a trading algorithm is by whether it is an aggressive, working order, or passive algorithm. An aggressive algorithm aims to execute all parts of an order as fast as possible. This type of algorithm will trade all parts of an order at once if the price is desirable. Many schedule-based algorithms are designed as aggressive algorithms because they trade large volumes in short durations of time. A working order algorithm trades at a rate that

⁴ Bacidore (2020), pp.1-3

best minimizes the cost and risk. Many dynamic algorithms are designed as working order algorithms and vary their rates of trading to keep losses to a minimum. A passive algorithm only trades if its activity will not disturb the market. Many opportunistic algorithms are designed as passive algorithms that only trade at times when they will not affect market liquidity.⁵

Schedule-Based Trading Algorithms:

Schedule-based single-order algorithms constitute one category of trading algorithms and include time- and volume-weighted average price algorithms and the scheduled-based implementation shortfall algorithm. The time-weighted average price (TWAP) algorithm is simply a program that executes the parts of an order in proportion to time. The algorithm divides the order into parts and evenly spaces out the execution of the resulting trades. It is possible to set a minimum quantity for each trade to prevent the machine from trading millions of individual stocks at increments of less than a second. An issue with this method is that it executes all orders at market price without allowing the trader to set limits on trade prices. To incorporate limit orders into the TWAP algorithm, the program may be allowed to go slightly off schedule to trade more when the market price is favorable. A limit order is a trade that is executed on the condition of being filled at a set price or better.⁶ Because it is a schedule-based algorithm, there are upper and lower bounds that control the volume of trades that can be executed at each interval. The lower bound ensures that the algorithm does not fall too off schedule while waiting for the market price to match the price prescribed by a limit order. The upper bound ensures that the algorithm does not act too aggressively and execute too many trades when the price is favorable.⁷

⁵ Kissell (2014), p.16

⁶ Cesari, et al. (2012)

⁷ Bacidore (2020), pp.31-35

Another schedule-based single-order algorithm is the volume-weighted average price (VWAP) algorithm. This algorithm tracks the volume of a stock that has been historically traded at each time of day. It then divides up the order and trades different proportions of the total volume at each interval of the order life. The order life is split up into intervals of time called bins. A portion of the order is assigned to each bin, reflecting the quantity that has been historically traded at that time. At each of these bins, the program follows the TWAP algorithm in that it executes that piece of the order in proportion to time. Similar to the TWAP algorithm, the VWAP algorithm accommodates limit orders by allowing itself to veer slightly off schedule. It sets upper and lower bounds that limit the volume traded throughout the order life. It should be noted that the TWAP algorithm is usually implemented as a VWAP algorithm set to a constant rate of trade throughout the day.⁸

The schedule-based implementation shortfall (IS) algorithm is another scheduled-based single-order algorithm. This algorithm uses the IS performance measure which compares the arrival price, the midpoint between the buy and sell offers, with the realized price of the trade. The realized price of the trade takes all costs, including any changes in market price, into account.⁹ The goal of the algorithm is to buy the stock at a price below the arrival price or sell the stock at a price above the arrival price. The algorithm assesses the current market prices and chooses the behavior that minimizes the costs and risks of trading. It may choose to implement the schedules of the TWAP or VWAP algorithms if deemed most appropriate. The program also uses these algorithms as a lower bound to ensure that the order is completed on time. If the schedule generated by the algorithm will result in too few trades, the program is compelled to trade the volume prescribed by the lower bound.¹⁰

⁸ Ibid, pp.37-40

⁹ Hayes (2021)

¹⁰ Bacidore (2020), pp.43-47, 65

Dynamic Trading Algorithms:

Dynamic single-order algorithms constitute another category of trading algorithms and include the standard, forecasted, and optimal percent of volume (POV) algorithms. The POV algorithm tracks the trades of a specific stock in the market and ensures that its own participation rate is kept constant. Similar to the VWAP algorithm, the POV algorithm monitors the volume of trades in the market at each point during the order life. The difference is that the POV algorithm monitors real-time data and executes orders to maintain responsibility for a percentage of trades in the current market. There is no guarantee regarding the total volume that will be executed by the POV algorithm, as there is with the VWAP algorithm.¹¹ The algorithm usually trades at market price because it reacts to other orders in the market. Upper and lower bounds may be set to accommodate limit orders and ensure that the algorithm does not deviate too far from its prescribed participation rate.¹²

One variation of the aforementioned algorithm is the forecasted POV algorithm. The algorithm divides the order life into intervals. At the end of each interval, it uses a model to predict the volume of trades it expects to see in the next interval. It uses real-time data to forecast the expected volume and calculates the number of trades it needs to initiate to maintain its participation rate. It does this until the entire order has been filled or until the order life ends. The forecasted POV algorithm differs from the standard POV algorithm in that it predicts future volumes. In contrast, the standard POV algorithm tracks previous trades to keep up its participation rate. The forecasted POV algorithm can be implemented similarly to the VWAP algorithm if the historical data is replaced with real-time data and the end time is made dynamic.

¹¹ Cesari, et al. (2012)

¹² Bacidore (2020), pp.56-59

The key difference between the two is that the VWAP algorithm is concerned with the percentage of an order that it fills at each point during the order life. The forecasted POV algorithm cares about the percentage of trades in the market that it contributes at each interval.¹³

The optimal POV algorithm is a variation of both the POV algorithm and the IS algorithm. Its goal is to determine the optimal participation rate based on real-time market data. It can be fed definitions for aggressive, moderate, and passive participation rates to aid the algorithm in choosing the optimal rate. This algorithm is dynamic in that its participation rate is intended to fluctuate with time. The standard POV algorithm will not fill an entire order if the market volume and participation rate call for too few trades. The optimal POV algorithm is also an IS algorithm and can therefore be instructed to complete an entire order. It may use the TWAP or VWAP algorithm as a lower bound to keep it on track. The algorithm will be compelled to trade even when the optimal participation rate is very low to ensure that the trade is completed by the end of the order life.¹⁴

Opportunistic Trading Algorithms:

Single-order opportunistic algorithms constitute the final category of single-order trading algorithms. This category includes the “hide and take” algorithm and the adaptive arrival price algorithm. The “hide and take” algorithm remains hidden until an opportunity in the market presents itself. Unlike the algorithms of the other categories, it is not concerned with keeping to a schedule or to a participation rate. It waits for an opportunity to arise rather than settling for market prices. One factor that this algorithm may monitor is the price of the stock. It waits for the stock to move into its desired price range before executing any trades. Another factor that the

¹³ Ibid, pp.60-63

¹⁴ Ibid, pp.64-65

algorithm may consider is market liquidity. The algorithm considers it an opportune time to trade when market liquidity substantially increases. The algorithm may also create its own opportunities. If it is used to monitor an illiquid market, it may execute some desirable orders to draw traders to the other side of the market. The algorithm will then fill further orders before the market turns illiquid again. For large quantities, the algorithm may seek out opportunities to trade in the dark market rather than in the lit market. This way it can avoid impacting the market price or liquidity. Similar to the bounds applied to the other algorithms, the “hide and take” algorithm can uphold a minimum participation rate. This ensures that it will execute a reasonable number of trades even if there are few opportunities. The algorithm may also allow the user to indicate the level of urgency of the order. If the order is of high urgency, then as mentioned before, the algorithm can create its own opportunities instead of waiting for them to arise.¹⁵

Another single-order opportunistic algorithm is the adaptive arrival price algorithm. This algorithm is another variation of the IS algorithm. The algorithm monitors the market price and trades more when the price moves in the desired direction and less when it moves the other way. Unlike the “hide and take” algorithm, this algorithm seeks to constantly trade. The rate at which it trades varies based on the current market conditions. The algorithm only exploits opportunities related to the price of the stock, while the “hide and take” algorithm considers many factors. The adaptive arrival price algorithm seeks to take advantage of current opportunities rather than making predictions about future market conditions. Although this is the safer option, some opportunities will inevitably be missed since it does not look ahead.¹⁶

Multi-Order Trading Algorithms:

¹⁵ Ibid, pp.68-70

¹⁶ Ibid, pp.71-72

Moving away from single-order algorithms, the next category of trading algorithms is multi-order, or list-based. The two types of list-based algorithms are pairs trading and portfolio algorithms. Pairs trading algorithms monitor the price of two stocks in relation to one another. The goal of the algorithm is to buy one of the stocks and sell the other when their price ratio drastically changes.¹⁷ There are upper and lower trigger values set for the price ratio. If the ratio exceeds the upper trigger value, the algorithm will buy units of one of the stocks and sell units of the other. If the ratio falls short of the lower trigger value, the algorithm will buy and sell the stocks in reverse. The algorithm stops trading when the ratio returns to an exit trigger value, or the algorithm reaches the maximum position size. The exit trigger is a value that indicates that the price ratio is returning to normal, and that the algorithm should refrain from any further trading. The maximum position size is set by the user to prevent the algorithm from buying too much of a stock at once. A leader stock is chosen based on which of the assets is more difficult to trade. Units of the leader and follower stocks are traded in repeated intervals and at amounts proportionate to their price ratio. These units are kept small to minimize the risk that the price ratio changes before the follower stock is traded.¹⁸

Portfolio algorithms are the other type of multi-order algorithms. They monitor many stocks at a time, making it very complex to apply the same logic as in the pairs trading algorithm. Therefore, most portfolio algorithms implement an upgraded version of the schedule-based IS algorithm. The schedule-based IS algorithm evaluates the costs and risks of a stock when creating its trading schedule. In contrast, portfolio algorithms take the costs and risks of all the stocks into consideration when creating individual trading schedules. The trading cost used in generating each of the schedules is simply the sum of all the trading costs. The risk is more

¹⁷ Huang, et al. (2015)

¹⁸ Bacidore (2020), pp.85-89

complicated to calculate because it accounts for the effect that the trading of one stock in the portfolio has another. The total risk is multiplied by a covariance term to account for the risk each order creates in the markets of the others. Once the schedules have been produced, each of the orders can be executed by a single-order algorithm, such as the VWAP algorithm. The historical data in the VWAP algorithm is replaced with the schedule created by the portfolio algorithm. The orders can either be sent in their entirety or in increments to the single-order algorithms. The latter approach allows for the portfolio algorithm to act similarly to an opportunistic algorithm. It can search for profit-making opportunities to trade the remaining parts of the orders before the next increment begins. The algorithm must be careful not to disturb the level of risk in any of the markets while looking for such opportunities, or else it would need to recalculate all the trading schedules.¹⁹

Metrics for Measuring Performance:

It is not enough to understand the intricacies of the aforementioned algorithms. Traders must also learn to compare their performances to select the best algorithms for specific cases. There are different methods for comparing the performance of trading algorithms including paired observations and independent samples. The paired observations approach takes an order and divides it into two equal parts. The halves are fed into different algorithms and are given the same start and end times for trading. This approach should only be used for algorithms that have a static trading strategy. This includes the standard TWAP, VWAP, and POV algorithms. The volume of trades may vary over time; however, the trading strategies of these algorithms do not change. These algorithms will act nearly identically even when implemented by different traders. The TWAP algorithm will always trade at a constant rate for a given volume size and order life.

¹⁹ Ibid, pp.90-98, 104-106

The VWAP algorithm follows the schedule prescribed by historical data and does not assess current market conditions. The POV algorithm trades at the volume required to maintain its predefined quota in the market. The paired observations method serves as a strong means of comparison because the algorithms trade in the same market at the same time. The reason its implementation is limited to algorithms with static trading strategies is because the results may be skewed if the algorithms compete with one another. If the algorithms reacted to the trades executed by their counterparts, then the results would not be a fair assessment of their behaviors. Since the trading strategies of these algorithms do not vary with market conditions, it is safe to have the pair of algorithms trade at the same time.²⁰

Another approach for comparing trading algorithms is through independent samples. This method is ideal for algorithms that use dynamic trading strategies. This includes the IS algorithm, forecasted and optimal POV algorithms, “hide and take” algorithm, and adaptive arrival price algorithm. The strategies of these algorithms vary with the change of market conditions. If two of these algorithms were programmed to trade at the same time, they may end up competing with one another. If one of the algorithms disturbs the market liquidity, it may trigger the other to trade more or less aggressively than it would normally. These results would not appropriately reflect the behaviors of the algorithms. The algorithms used in independent samples should be given identical orders and time periods. The samples are independent because the algorithms trade at different times. They should also be given large enough orders to ensure that the algorithms trade at their intended capacities.²¹

Data from paired observations and independent samples is used to calculate metrics for comparing the algorithms. Financial analysts most commonly use the IS and VWAP benchmarks

²⁰ Kissell (2014), pp.115-116

²¹ Ibid

to evaluate the performance of trading algorithms. The IS benchmark compares the arrival price and average execution price of an order. The arrival price is the midpoint between the buy and sell offers at the time the decision to trade is made. The IS benchmark calculates the difference between the arrival price and realized price. This metric considers the loss endured due to the inability of the algorithm to trade large volumes all at once. The algorithm will inevitably buy above or sell below the arrival price due to transactional delays. For each part of an order, the IS benchmark measures the difference between the price at the time of the decision and the average execution price. The IS benchmark and volume size of all the parts of an order must be considered in evaluating the overall performance of a trading algorithm.²²

Another metric used for measuring the performance of trading algorithms is the VWAP benchmark. This metric takes the typical price of a stock over a set period and divides it by the volume traded. The typical value is the average of the highest, lowest, and final price at which the stock was traded. This is repeated for all periods during the order life and the values are added all together. The VWAP benchmark is calculated by dividing this value by the total volume traded during the order life.²³ It is typically preferred to use the VWAP benchmark when comparing trading algorithms through paired observations. This method is used for algorithms that have static trading strategies and therefore trade more aggressively. The VWAP benchmark is preferred for static algorithms because it measures the weighted average price. This is relevant for aggressive algorithms since they constantly execute trades with the intention of maintaining a desirable average execution price. The IS benchmark is preferred for comparing trading algorithms through independent samples. This method is used for algorithms that have dynamic trading strategies and therefore trade more passively.²⁴ The IS benchmark is preferred for

²² Markov (2019)

²³ Fernando (2022)

²⁴ Kissell (2014), pp.116

dynamic algorithms because it measures the execution and opportunity costs.²⁵ This is relevant for passive algorithms since they aim to trade when costs and risks are minimized.²⁶

Challenges of Algorithmic Trading:

Algorithmic trading presents many challenges that traders must consider in choosing which algorithms to implement in specific cases. One challenge is selecting the best approach for trading in illiquid markets. Illiquid markets have few participants which makes it difficult to trade. As mentioned earlier, opportunistic algorithms can be used. Algorithms such as the “hide and take” algorithm wait for opportunities to present themselves. They may also resort to creating their own opportunities by temporarily drawing more participants to the market. The issue with waiting for opportunities to arise is that there are more risks associated with large orders. If the algorithm waits for an opportunity it will try to exploit it by issuing trades at large volumes. A better solution may be to use algorithms with static trading strategies such as the TWAP, VWAP, and POV algorithms. These algorithms produce better results in illiquid markets than algorithms with dynamic trading strategies because their focus is on the average execution price. The intention of dynamic algorithms is to minimize the costs and risks of every trade. If an illiquid market were to attract more participants, a dynamic algorithm may trade aggressively to take advantage of the larger spread of buy and sell bids. This approach could drain the market of its resources, adding risk to the resulting orders. Instead, it may be better to use static algorithms that trade at a steady rate. These algorithms can still trade more aggressively when opportunities arise; however, their upper bounds prevent them from disturbing the liquidity of the market.²⁷

²⁵ Hayes (2021)

²⁶ Markov (2019)

²⁷ Shen (2021)

Another challenge of algorithmic trading is handling special days. Special days include half trading days, index rebalancing days, and federal announcement days. The occurrences on these days may impact the overall stock market or individual stocks. Historical data alone cannot accurately forecast market movements on these days due to the uncertainty of external factors. Algorithms that incorporate advanced statistical models or machine learning should be used to predict market changes on these days. These models make better predictions than standard trading algorithms because they take factors other than historical data into account. Some of these models can even be implemented to accommodate special periods of time. These periods can last up to a week or a month. This is relevant when a stock is first introduced to the public and its trading patterns are uncertain. Models for special periods are not regularly used because they are difficult to compute; however, it is in the best interest of traders to produce these models and take advantage of the opportunities presented on special days.²⁸

An additional challenge of algorithmic trading is the potential for human error in programming. Algorithms are improved over time to correct mistakes, update strategies, and implement new features. Turnover is common in industry and new people are often assigned to work on existing projects. Programmers do not all share the same coding styles. This makes it difficult for new employees to work on existing programs and often results in coding errors. This is especially an issue in algorithmic trading because even a small mistake can lead to a significant loss of money. Solutions to this problem include the use of object-oriented programming and unit and regression testing.²⁹

The first solution to the issue of human error is to use programming languages that support object-oriented programming such as Java and C++. Object-oriented programming is a

²⁸ Ibid

²⁹ Ibid

programming style that focuses on objects and their transactions. Objects represent real-world entities and are assigned unique identities, attributes, and behaviors. Classes of code are designed to initialize objects and regulate their transactions. Programmers can limit the accessibility of features within an object's class to ensure that they are only used in the intended manners. The reason that this form of programming is protective against human error is because it is formulated in layers of classes that are limited in their interactions with one another. The idea is to restrict changes made in one area of the code from affecting other areas.³⁰ This is not a full proof solution because objects are designed to interact with one another. These interactions may not work as originally intended if later changes are made by new programmers. It is therefore important to understand the intricacies of a program before making any changes to it.³¹

Regardless of whether a program has the protective features of object-oriented programming, it is important to thoroughly test all aspects of the code. Unit tests are used to test individual functions of a program. Regression testing is used to test the overall program after changes are made to it. The issue that arises with these tests is that they often do not check for all possible edge cases. To thoroughly test a trading algorithm, simulations of extreme market conditions must be automated to assess the reactions of the program. The reason that this is difficult to achieve is because most algorithms are comprised of series of conditional statements. It is not simple to ensure that all combinations of an algorithm's features are tested. A unit test could be used to check the functionality of an update to the program. Further regression testing would be necessary to examine how that change affects the remainder of the program. It is important to note that some updates to a program will produce more possible outcomes.

³⁰ Prabhu (2022)

³¹ Shen (2021)

Therefore, although testing is sometimes viewed as a tedious process, it is important to update the tests as the program is improved to ensure the effectiveness of the trading algorithm.³²

Integration of Machine Learning:

With the rise of big data came the integration of machine learning into the trading industry. Machine learning algorithms are not explicitly programmed; rather, they are designed to train themselves. This differentiates them from the trading algorithms previously discussed because those require the programmers to give explicit directions in the code. The three categories of machine learning include supervised, unsupervised, and reinforcement learning. Supervised learning is used to label sets of data. This type of machine learning model uses labeled data to train itself to predict labels for unseen data. It produces a function that takes unseen data as input and returns its predicted label. Unsupervised learning is similar, except that it is not trained with labeled data. Rather, it must learn to categorize data into groups on its own. It produces a mapping system that classifies the data that it takes as input and returns its labels. Reinforcement learning trains an agent to choose the actions that will maximize its reward. The actions performed by the agent move it between the predefined states of its environment. The agent learns which actions to choose based on the occasional rewards it receives. This form of machine learning is used for making predictions and is the category most used in trading.³³ When applying reinforcement learning to trading, the environment refers to the collection of data that the agent, or machine, uses to make decisions. The states refer to the data that describes the current market conditions and the agent's position. The agent uses a learned policy to inform it of

³² Ibid

³³ Koshiyama, et al. (2020)

the best action to take based on the current state. In cases of portfolio management, the agent may have to make multiple decisions to claim rewards and bring it to the next state.³⁴

Machine learning is applied in the trading industry to monitor market data and execute trades. Natural language processing is the area of machine learning used for analyzing language and speech. It can be used to examine data from social media and assess the public attitude towards a stock. It can also be used to analyze statements released by a publicly traded company and evaluate its economic well-being. Machine learning algorithms typically make more accurate predictions than standard trading algorithms due to the inclusion of this data. These market monitoring algorithms are paired with other machine learning algorithms that execute trades. They act as smart order routers that monitor the price of a stock across different venues and execute the desired trade on the platform with the best price.³⁵ Machine learning algorithms are also used to devise optimal trading schedules. Many of these algorithms are adaptations of the VWAP algorithm. The difference is that they use reinforcement learning to inform their trading strategies. This is advantageous because reinforcement learning algorithms are trained based on past experiences. The VWAP algorithm only uses historical data in its decision making and cannot learn from past successes or failures. Although using reinforcement learning is a strong strategy for order execution, it is very difficult to implement in markets that are constantly changing.³⁶

Another area of machine learning is deep learning. It is categorized by its hidden processing layers that transform inputted data. Each layer is designed to bring the data closer to the desired results. This structure is known as a neural network and is intended to replicate the functionality of the human brain. One algorithm that uses supervised deep learning is the

³⁴ Ritter (2017)

³⁵ Koshiyama, et al. (2020)

³⁶ Hansen (2020)

long-short term memory (LSTM) algorithm. It is mostly used for predicting stock prices. It is structured as a recurrent neural network that stores past data to inform future decision making. This type of neural network is useful for recognizing patterns in times series data because of its ability to reference past data. The LSTM algorithm categorizes data into long- and short-term memory cells. It must limit the amount of data that it stores to optimize its efficiency. The algorithm would work very slowly if it had to examine all the past data in making predictions. This would not be useful for the split-second decision making required of the fast-paced markets.³⁷

Moving away from the different applications of machine learning in trading, it is essential to understand how to choose which algorithms to implement. Traders must consider the tradeoff presented between accuracy and interpretability. Many machine learning algorithms are called black boxes due to the inability of humans to understand their inner workings. The reason for this tradeoff is because algorithms tend to make more accurate predictions as they examine more parameters. Algorithms also become more complex as they incorporate a larger number of parameters. The more complex an algorithm, the less interpretable it becomes to humans. Reducible errors occur when improper algorithms for specific cases are implemented. To minimize these types of errors, it is essential to understand the data and choose the algorithm that best approximates the relationship between inputs and outputs. Two types of reducible errors exist and are known as bias errors and variance errors. Bias errors occur when the programmer makes incorrect assumptions about the relationship between variables. The programmer chooses to implement an algorithm that is incapable of approximating the function between inputs and outputs. The results produced by such an algorithm are not accurate. Variance errors arise when the algorithm is overfitted to the training data. Overfitting occurs when the function produced by

³⁷ Koshiyama, et al. (2020)

the algorithm is too heavily influenced by trends specific to the training data. The algorithm will then make predictions that are not accurate for new data. Deep learning algorithms tend to have higher variance errors and lower bias errors. The reverse is true for simpler machine learning algorithms. These types of errors are inevitable; however, they can be reduced if the programmer is well versed in the different types of machine learning algorithms. He or she must study the data set before selecting which algorithm to implement.³⁸

Conclusion:

The intention of this paper is to emphasize the importance of education in the field of trading. This paper was purposely written in a manner that is understandable to someone with limited background in computer science or finance. With today's technology, it is easy for anyone to access stock prices and initiate trades. Trading can be expanded beyond this with the incorporation of algorithms. Computers process data at speeds that far exceed the capabilities of the human brain. It is therefore essential for traders to understand the potential of the powerful tool that is algorithmic trading. As research in machine learning advances, these algorithms are being implemented by more financial firms. Machine learning algorithms are capable of processing more data than standard algorithms and implementing trading strategies that are beyond human understanding. Standard algorithms are limited to the strategies programmed by humans, but machine learning algorithms surpass human intellect. Although machine learning is enticing, the field presents many of its own challenges. It is therefore important to acknowledge these setbacks before choosing to implement a machine learning strategy. No matter which approach is taken for trading, the trader must educate him or herself about the methods used to

³⁸ Lee and Shin (2020)

compare and evaluate algorithms. Only through this process can he or she choose the best algorithm to trade in specific scenarios.

Appendix:

	Schedule-Based	Dynamic	Opportunistic
Single-Order	TWAP, VWAP, Schedule-Based IS	POV, Forecasted POV, Optimal POV (IS)	“Hide and Take”, Adaptive Arrival Price (IS)

Figure 1: Classification of Single-Order Trading Algorithms

	2 Orders	3+ Orders
Multi-Order	Pairs Trading	Portfolio Algorithm

Figure 2: Classification of Multi-Order Trading Algorithms

Static Trading Strategies	Dynamic Trading Strategies
TWAP, VWAP, POV	Schedule-Based IS, Forecasted POV, Optimal POV, “Hide and Take”, Adaptive Arrival Price

Figure 3: Classification of Single-Order Trading Algorithm Strategies

References

1. Arnoldi, Jakob. "Computer Algorithms, Market Manipulation and the Institutionalization of High Frequency Trading." *Theory, Culture & Society*, vol. 33, no. 1, Oct. 2016, pp. 29–52., doi:10.1177/0263276414566642.
2. Bacidore, Jeffrey M. *Algorithmic Trading: A Practitioner's Guide*. TBG Press, 2020.
3. Cesari, Riccardo, et al. "Effective Trade Execution." *Portfolio Theory and Management*, 22 June 2012, pp. 411–437., doi:10.1093/acprof:oso/9780199829699.003.0018.
4. Fernando, Jason. "Volume-Weighted Average Price (VWAP)." *Investopedia*, Investopedia, 6 May 2022, www.investopedia.com/terms/v/vwap.asp.
5. Hansen, Kristian Bondo. "The Virtue of Simplicity: On Machine Learning Models in Algorithmic Trading." *Big Data & Society*, vol. 7, no. 1, 2020, doi:10.1177/2053951720926558.
6. Hayes, Adam. "Implementation Shortfall." *Investopedia*, Investopedia, 19 May 2021, www.investopedia.com/terms/i/implementation-shortfall.asp.
7. Huang, Chien-Feng, et al. "An Intelligent Model for Pairs Trading Using Genetic Algorithms." *Computational Intelligence and Neuroscience*, vol. 2015, 14 Mar. 2015, doi:10.1155/2015/939606.
8. Kissell, Robert. *The Science of Algorithmic Trading and Portfolio Management*. Elsevier/Academic Press, 2014.
9. Koshiyama, Adriano, et al. "Algorithms in Future Capital Markets." 11 Mar. 2020.
10. Lee, In, and Yong Jae Shin. "Machine Learning for Enterprises: Applications, Algorithm Selection, and Challenges." *Business Horizons*, vol. 63, no. 2, 2020, pp. 157–170., doi:10.1016/j.bushor.2019.10.005.

11. Markov, Vladimir. “Bayesian Trading Cost Analysis and Ranking of Broker Algorithms.”
22 Apr. 2019.
12. McGowan, Michael J. “The Rise of Computerized High Frequency Trading: Use and Controversy.” *Duke Law & Technology Review*, vol. 16, 2010.
13. Prabhu, Rishabh. “Object Oriented Programming (OOPS) Concept in Java.”
GeeksforGeeks, 13 May 2022,
www.geeksforgeeks.org/object-oriented-programming-oops-concept-in-java/.
14. Rinker, Robert. “Algorithm Efficiency.” 2002.
15. Ritter, Gordon. “Machine Learning for Trading.” *SSRN Electronic Journal*, 21 Aug. 2017, doi:10.2139/ssrn.3015609.
16. Shen, Jackie. “Nine Challenges in Modern Algorithmic Trading and Controls.”
Algorithmic Trading and Controls, vol. 1, no. 1, 21 Jan. 2021, pp. 1–9.