Can a Machine Learning Model Consistently Learn Profitable Trading Strategies in the Forex Market?

Ethan Pedersen
Computer Science Department
Brigham Young University
Provo, United States
ethanp55@byu.edu

Jacob W. Crandall
Computer Science Department
Brigham Young University
Provo, United States
crandall@cs.byu.edu

***Abstract*—Financial day trading is a popular endeavor, whether as a part-time hobby or full-time career. Despite this popularity, some argue that day trading is a zero-sum (or perhaps even negative-sum) game and more focus should be placed in longer-term investments. Nevertheless, in recent years, many have attempted to employ machine learning (ML) algorithms in day trading due to their ability to quickly process large amounts of information to arrive at informed trading decisions. Despite ongoing research, it remains unclear how to achieve consistent profit with such ML algorithms, especially given that existing work often does not test trading algorithms under realistic market simulations. Given these considerations, should we expect machine learning models to at least be capable of achieving the minimax outcome (break even)? The answer to this question could guide future research by improving promising algorithms to potentially yield consistent, positive returns. In this paper, we address this question in the context of the foreign exchange (forex) market. Specifically, we analyze the ability of four different genres of machine learning algorithms to make profitable investments in the forex market for small investors using realistic market simulations. Our results also highlight the difficulty of day trading and shed insight into possible qualities of successful algorithms.**

***Index Terms*—Machine Learning, Forex, Day Trading**

I. Introduction

The perceived ability of machine learning (ML) algorithms to trade in and exploit financial markets has been and continues to be high. Tantalizing results, including those obtained by the authors as they made their own foray into trading in the foreign exchange (forex) market, give credence to this perception.

Using ML algorithms in day trading[1] has several potential advantages. First, using automated algorithms allows traders to trade for more hours of the day (including night hours) and on more securities at one time. Second, some view the failures of human-based trading to be due to emotion and/or lack of speed in trade execution, which are problems that can potentially be solved with automated, algorithmic strategies [1], [2]. Finally, machine learning models seem to have the ability to discover useful patterns that are difficult for people to identify [3]. These perceived advantages seem to indicate why ML algorithms have been considered extensively for financial trading over the last thirty (plus) years (e.g., [4]–[6]).

[1] In basic terms, *day trading* (or, simply, *trading*) is the practice of buying and selling securities rapidly, often in the span of less than a single day. Trading operates on much smaller time frames than *investing*.

While these potential advantages exist, there is a strong ongoing debate about whether day trading (with or without ML) is a profitable endeavor in the first place. The concept of trading is sometimes viewed through the lens of a multi-agent system, where numerous agents buy and sell commodities with the goal of maximizing profit [7], [8]. As these agents execute trades, a feedback loop is triggered, leading to price adjustments that require strategy reassessment and adaptation. Such systems are extremely difficult to model. Furthermore, it has been argued that day trading is a zero-sum game [9], [10] where one player's gain is another's loss. John Bogle, the founder of The Vanguard Group, is famously known for saying that beating the market is a zero-sum game, and a negative-sum game after accounting for miscellaneous fees [11]. In fact, Ramsey Solutions likens trading to gambling [12]. Finally, while there is evidence that skilled long-term investors can consistently *beat the market* [13], data suggests that relatively few day traders are successful. For example, data suggests that less than 1% of traders are profitable in the Brazilian equity futures market [14]. Other sources (e.g., [15]) estimate that at least 90% of traders lose money in the long run.

Despite these negative views about day trading, there are also plenty of sources that claim that, with enough time, dedication, focus, and effort, one can become a profitable trader. The existence of numerous trading courses and academies indicates that many hold optimistic beliefs about day trading. At the time of writing, Investopedia maintains a list of the top-ranked trading schools and courses [16]. Existing research also often touts the predictive accuracy of ML trading models, though they often do not always test them in simulations that replicate realistic market conditions (including spread and day fees) [17].

Considering these various challenges, can ML algorithms at least consistently learn the *minimax* outcome? A common value to strive for when dealing with zero-sum problems is the minimax value [18], which is the best outcome under the worst possible conditions. We would argue that, in day trading, the minimax value is a profit of $0, as one can simply avoid trading altogether. We hope that, by discovering which algorithms can consistently break even (or better), we can shed insight into which algorithms should be explored further in future research, as well as what qualities a profitable algorithm might need.

Specifically, in this paper, we study the ability of four genres

TABLE I

ALGORITHMS USED IN OUR EXPERIMENTS. COUNTS INDICATE THE NUMBER OF ALGORITHMS SELECTED FOR EACH GENRE.

Genre	**Count**	**Algorithms**
Trained Rules	14	Bar Movement (BM), Beep Boop (BBo), Bollinger Bands (BB), Choc, Keltner Channels (KC), MA Crosover (MAC), MACD, MACD Key Levels (MACDK), MACD Stochastic (MACDS), PSAR, RSI, Squeeze Pro (SP), Stochastic (Stoch), and Supertrend (Sup)
Ensemble	1	Ensemble [19] (E)
Bandits	4	UCB [20], EXP4 [21], EEE [22], and AlegAATr (Aleg) [23]
Price Forecasters	7	LSTM-Mixture (LSTMM) [24], CNN w/ Grammian Angular Fields (CNN) [25], Transformer w/ Time Embeddings (TranT) [26], KNN [27], LSTM [28], MLP [29], and Random Forest (RF) [30]

of ML algorithms to consistently learn profitable strategies in the forex market by testing them in realistic market simulations. Forex is the global financial market where individuals and organizations engage in currency trading. Widely regarded as the world's largest market, estimates suggest that the forex market has an average daily trading volume of approximately $6.6 trillion [31]. Compare this with the NYSE's daily volume of roughly $18.9 billion [32], and one can gain appreciation for the forex market's size.

II. THE FOREX MARKET: BACKGROUND

In the foreign exchange (forex) market, *currencies* are traded in *pairs*. Each currency pair represents the exchange rate between two different currencies. For example, the EUR/USD trading at 1.2548 means that one would pay 1.2548 dollars for a single euro. Prices in forex are usually broken into *bid* and *ask* prices. The bid price is the price to sell and the ask price is the price to buy. The difference between these prices, known as the *spread*, is how brokers typically profit from trades regardless of outcome (one can think of it as a small fee paid to the broker on every trade). Aside from spread, brokers might charge *day fees* if a trade is left open overnight. These fees depend on trade size, currency pair, and other factors. The *middle* price is the price halfway between the bid and ask prices. A *pip*, usually represented to four decimal places, is the unit of measurement used in forex to express the change in value between two currencies. For example, the difference between 1.0255 and 1.0250 is 0.0005, or 5 pips.

Most trading charts display *price candles*, composed of open, close, high, and low prices for a given time interval. The opening and closing price in the interval form the candle body, while the high and low price during the time interval form the candle wicks. Bullish candles indicate that price increased over the time interval (the price at the end of the interval was larger than the price at the beginning). Bearish candles show decreases (the close price was smaller than the open price). Trading charts can be adjusted to different *time frames*. For instance, each candle in a thirty-minute time frame represents thirty minutes of price data.

Traders often use key levels, such as a *stop-loss* and/or a *stop-gain*, to manage risk. If the price hits a stop-loss, the trade automatically closes, or stops, for a loss. Similarly, if the price hits a stop-gain, the trade closes for a gain. Traders may adjust these levels after placing a trade and are allowed to use one, both, or none.

III. MACHINE LEARNING FOR FOREX DAY TRADING

We selected algorithms from four genres of ML algorithms for use in forex trading: (1) trained rules, (2) ensembles of trained rules, (3) bandits with expert advice, and (4) price forecasters. For each genre of algorithm, we selected a representative set of one or more ML algorithms (Table I). In this section, we overview these algorithms and justify why we selected them. Algorithmic details are given in the Appendix.

A. Trained Rules

The first set of algorithms we consider are *trained rules*. These algorithms typically represent either popular trading strategies or variations on popular trading strategies that are well-known in the trading community based on known indicators. For example, the prominent MACD (Moving Average Convergence Divergence) indicator is commonly thought to help traders identify changes in market sentiment and momentum, and thus can in turn be used to define a behavior rule. We selected fourteen of these behavior strategies, twelve of which are based on well-known indicators and two of which (Bar Movement and MACD Key Levels) are customized rules defined by one of the authors.Each trained rule has parameters that are, for each currency pair, tuned by a genetic algorithm using historical trading data.

B. Ensembles of Trained Rules

Ensemble algorithms combine the decisions of multiple behavior rules in attempt to boost performance. We created a single ensemble algorithm similar to the one defined by Fisichella [19]. This ensemble agent combines the predictions of a subset of the fourteen trained rules listed in the previous subsection. Let Γ denote this set of trained rules. A genetic algorithm is used to identify a subset $\Gamma_s \subseteq \Gamma$ of these trained rules. Each $G_i \in \Gamma_s$ *votes* for both the type of trade and the trade parameters that should be used at any given time. For discrete parameters, the ensemble agent uses the statistical mode of the votes to make its decision. For continuous parameters, the ensemble agent uses the mean. For example, if the majority of the trained rules vote that a *buy* should be placed (discrete parameter), *buy* is the selected action. If there are three trained rules that vote that 10, 20, and 15 pips should be used for the stop-loss size (continuous parameter), respectively, then the stop-loss is set to $\frac{10+20+15}{3} = 15$ pips from the open price.

C. Bandit Algorithms

Similar to ensemble algorithms, bandit algorithms combine multiple trained rules (or experts) in attempt to make superior decisions. However, rather than combining together advice from all training rules, bandits instead seek to learn to follow the *best* expert, often based on the principle of regret, at any particular moment. While many flavors of bandit algorithms exist [33], we selected four such algorithms. The first three (Exp4 [21], EEE [22], and UCB [20]) are well-known and commonly used bandit algorithms with known regret bounds. The fourth algorithm, AlegAATr [23], is a more recently developed contextual bandit algorithm based on the concept of assumption-alignment tracking [34]. AlegAATr has been shown to be effective in a variety of multi-agent domains, largely due to its ability to adapt quickly to sudden shifts in the environment (which is important for forex).

Each bandit algorithm was supplied with the set Γ of fourteen trained rules for use as experts. At each time step, the bandit follows the expert advice of the bandit $G_i \in \Gamma$ selected by the algorithm for that time step.

D. Price Forecasters

Price forecasters predict future market prices. They then use these price forecasts to determine which trades to make. Most efforts in this area focus on the price-forecasting aspect of these algorithms. We include three different price-forecasting algorithms designed specifically for predicting prices for forex day trading. First, we include LSTM-Mixture (LSTMM) [24], a neural network that uses LSTMs with different activation functions. Second, we include CNN with Grammian Angular Fields (GAF) [25], an approach borrowed from the computer vision literature that uses a snapshot of the previous N price candles, enhanced with a GAF, to predict future market movement. Third, we include Transformer with Time Embeddings (TranT) [26], a transformer neural network that uses time embeddings in its input. We also include a handful of popular ML algorithms (KNN, LSTM, MLP, and Random Forests) as additional price forecasters for points of comparison.

The agent uses the future predicted price to determine what trades (if any) to place. While there are numerous ways to make this decision, we found the following mechanism to be effective based on trial and error. If the predicted price for the next price candle indicates sufficient movement in a given direction, the agent places a trade in this direction.

E. Tuning Hyperparameters

The behavior of each algorithm in Table I is dependent on a variety of hyperparameters that can be tuned in an effort to maximize net profits. These include both strategy-specific hyperparameters as well as hyperparameters that are common across all strategies. A genetic algorithm (GA) is used to tune the algorithms' hyperparameters, similar to Fisichella [19]. For each currency pair, time frame, and algorithm, the GA runs multiple generations of genetic evolution to find parameters that maximize net profit. Implementation details and descriptions of hyperparameters are given in the Appendix.

IV. Experiments

We created a simulation environment that iterates through historical market data (from November 2013 to November 2023), retrieved using Oanda's (a forex broker) API [35], to recreate, as realistically as possible, original market conditions. In these simulations, day fee estimates (for trades that open before and close after 5 PM ET) and loss due to spread are incorporated into final trade amounts. The simulations assume the agent has an initial account balance of \$10,000, with each trade risking 2% of the account value at the time of the trade's placement. For example, if the current account balance has grown to \$15,000 during the simulation, a new trade would risk $0.02 \times \$15000 = \$300$.

Experiments were conducted using three currency pairs: EUR/USD, USD/JPY, and GBP/CHF. We believe these three pairs provide a simple yet representative sample of different market characteristics: two contain the US dollar (EUR/USD and USD/JPY), one is a cross pair (GBP/CHF), and one contains the Japanese yen (USD/JPY). Thirty-minute (M30), one-hour (H1), and four-hour (H4) data were used in our evaluations. In short, we used nine distinct pair-time combinations (three pairs crossed with three time frames).

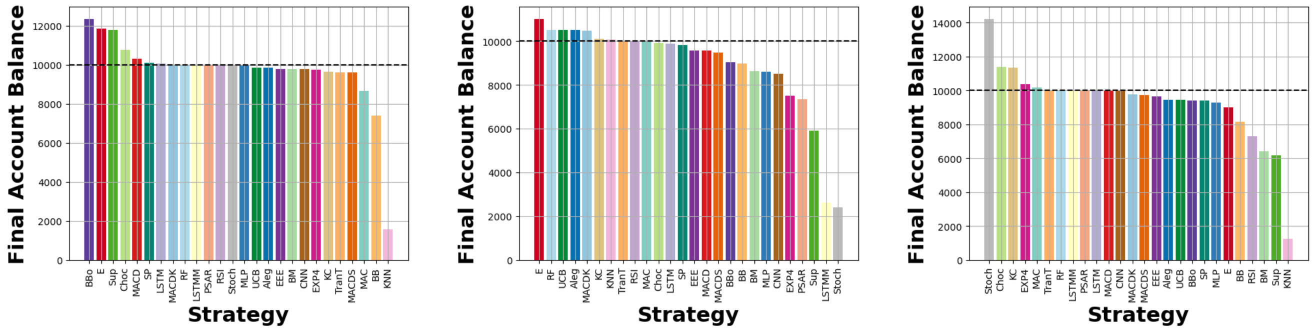
Two main experiments were executed. In the first experiment (Phase 1), agents were trained on data from November 2013 through October 2020 and tested on data from November 2021 to November 2023. In the second experiment (Phase 2), we alternated between training and testing years. For example, we trained the agents on data from 2013 to 2014, and then used those tuned parameters to test from 2014 to 2015. We then re-trained on data from 2014 to 2015, and then re-tested on 2015 to 2016 data, etc. In short, this second phase can be viewed as a type of "sliding window" of training and testing periods over the previous ten years of market data. The motivation for Phase 2 was to see how consistent the agents were over a longer period of time. At the beginning of every test simulation for both phases, we reset the initial account balance to \$10,000.

These evaluations assume that the agents would not have impacted forex prices due to the relatively small amounts they trade with. This assumption is consistent with the fact that most individual traders exert minimal influence compared to larger institutions (such as global corporations and banks) that can potentially sway market prices in their favor [36].

V. Results

Figure 1 shows the final account balances for each agent on each currency pair in the Phase 1 (H1) experiment. For each currency pair, most of the algorithms either broke even or lost money. However, there are a handful of cases that are promising, such as Beep Boop (BBo) on EUR/USD H1 and, perhaps most notably, Stochastic (Stoch) on USD/JPY H1 (which gains over \$4,000 of profit). These cases show annual returns of 20% or more, which is significant considering that the S&P 500 has yielded approximately 10% annual returns over the course of its existence [37].

While these selected results seem to indicate that some ML algorithms can make outstanding profits, they are offset by



(a) EUR/USD H1 (b) GBP/CHF H1 (c) USD/JPY H1

Fig. 1. Final account balances in the last year of trading in the Phase 1 experiment for all three currency pairs (H1 time frame). A black dashed line shows the initial account balance of $10,000 at the beginning of the year – anything above this line is a profit.

TABLE II
AVERAGE ANNUAL PROFITS ($) OVER ALL PHASE 2 YEARS FOR EACH AGENT, TIME FRAME, AND CURRENCY PAIR, ROUNDED TO THE NEAREST DOLLAR. ± VALUES GIVE THE STANDARD ERROR OF THE MEAN.

Strategy	M30	H1	H4	EUR/ USD	GBP/ CHF	USD/ JPY	Overall
Aleg	125	-33	-58	**31**	-230	**232**	**11** (± 106)
Choc	-246	28	7	-72	-338	199	-70 (± 95)
RSI	**170**	-539	**31**	-196	-106	-37	-113 (± 96)
UCB	-184	-135	-247	-244	-262	-60	-189 (± 127)
EXP4	-636	188	-185	13	-357	-290	-211 (± 185)
MACD	-394	-370	-40	2	-445	-361	-268 (± 163)
SP	-398	-347	-64	-351	-395	-63	-270 (± 80)
MACDK	-305	**40**	-690	-302	-594	-59	-318 (± 108)
EEE	-316	-688	-139	-732	**118**	-529	-381 (± 240)
MAC	-559	-477	-368	-227	-736	-440	-468 (± 128)
MACDS	-849	-385	-257	-307	-176	-1008	-497 (± 157)
Sup	-664	-738	-492	-991	-452	-451	-631 (± 251)
E	-767	-611	-720	-761	-342	-996	-699 (± 201)
BB	-1117	-408	-689	-1135	-346	-734	-738 (± 212)
LSTMM	-837	-1110	-278	-534	-707	-985	-742 (± 177)
LSTM	-499	-517	-1229	-934	-853	-458	-748 (± 213)
KC	-1053	-1017	-208	-1028	-1120	-130	-759 (± 183)
TranT	-1065	-939	-316	-888	-776	-655	-773 (± 261)
BM	-1365	-617	-373	-694	-693	-968	-785 (± 259)
Stoch	-1369	-293	-747	-514	-635	-1260	-803 (± 235)
KNN	-819	-829	-860	-864	-780	-863	-836 (± 260)
CNN	-1185	-760	-1071	-1069	-1100	-846	-1005 (± 305)
MLP	-1363	-682	-1294	-1440	-571	-1327	-1113 (± 198)
RF	-1787	-993	-837	-1163	-1237	-1218	-1206 (± 257)
PSAR	-1582	-1577	-533	-1905	-722	-1064	-1230 (± 287)
BBo	-1274	-1578	-1270	-945	-711	-2466	-1374 (± 268)
Average	-771	-578	-504	-654	-552	-647	-618

the dismal performance of the same algorithms in other cases. For example, while Stochastic was outstanding for USD/JPY H1, it was exceptionally poor for GBP/CHF H1 in 2022. This invokes questions about algorithm viability.

Phase 2 experiments answer this question. Table II gives a summary of the profits obtained by the agents over all years, broken down by agent, time frame, and currency pair. While there are a few positive cases for some time frames and currency pairs, profits are mostly negative. Indeed, none of the agents made more than negligible profits overall in an average year.

We observe interesting trends regarding genres of algorithms.

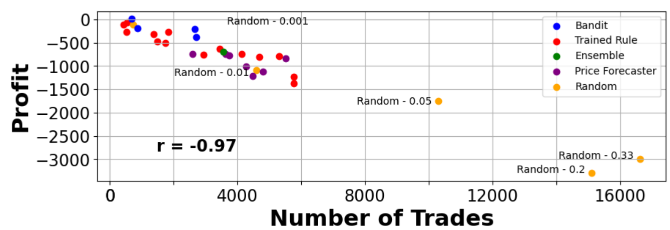


Fig. 2. Average annual profits as a function of the average number of trades placed (Phase 2). Numbers next to Random state the probability of Random placing a buy or sell (e.g., *Random - 0.2* has a 0.2 probability of entering a trade when it does not have a trade open). Random sets a static stop-gain and stop-loss of 50 pips above and below a trade's open price.

Table II shows that Bandit Algorithms tended to be among the highest performing algorithms in the Phase 2 experiments. Among these bandits, AlegAATr obtained the highest average profit (roughly achieved the minimax outcome of breaking even), though a multiple comparisons test between the bandits reveals no statistical significance in profits (see the Appendix). On the other hand, Price Forecasters were all among the worst performing agents. LSTMM (fifteenth highest average profit) was the best performing Price Forecaster, losing $742 per year, on average. The Appendix discusses (unsuccessful) attempts to improve this algorithm's performance as well as an analysis of how price predictions impacted its performance. These results highlight the importance of testing algorithms under realistic market conditions, as metrics such as predictive accuracy might be too optimistic. In short, we would argue that bandit algorithms, and AlegAATr in particular, were the most capable of achieving the minimax value.

Why do the algorithms perform so poorly overall, and what makes some algorithms perform better than others? To help answer these questions, we plot the average annual profits of the algorithms against the number of trades they placed in Figure 2. The figure shows a strong negative correlation between the number of trades placed and profits (Pearson correlation of $r = -0.97$). This result indicates that algorithms that performed better learned to trade less. The figure also shows comparisons to Random algorithms, which placed random trades at different rates. When the Random algorithms traded less, they typically performed better, on par with the ML algorithms that traded with similar frequency. Given that Random's expected profit on any trade is zero minus fees for each trade it enters, it is clear that many of the ML algorithms perform no better than Random. Losses are primarily due to trading fees.

VI. Discussion

In this section, we address three important reactions to the results presented in the previous section.

A. Reaction 1: ML algorithms really are no better than random

Figure 2 shows a strong negative correlation between profit and the number of trades an algorithm placed. The figure also shows that most, perhaps even all, of the ML algorithms performed no better than random. Is there anything positive or worthwhile to learn from these results? We offer two answers to this question. First, we believe that one could remain optimistic and take the view that, in order for ML algorithms to consistently profit, they should be extremely selective in the trades they choose to pursue (i.e., few, but very confident, trades). Second, given that these results are consistent with estimates that the majority of individual traders lose money in the long run, we argue for the importance of testing ML algorithms in realistic simulations as part of ongoing and future research, as even small fees can eat into profits.

B. Reaction 2: The chips might be stacked Against ML algorithms in forex

In addition to the number of trades and numerous fees (Figure 2) that provide an explanation as to why it can be difficult for ML algorithms to succeed in day trading, we offer another possibility. Results reported in the previous section highlight the zero-sum (even, negative-sum) nature of forex day trading. Only one of the 25 ML algorithms (4%) in our study was profitable on average (and only by a small amount), which corresponds to data measuring the performance of human traders [14], [15]. A reasonable question to ask then is, who actually makes money in forex trading? One party that profits, regardless of a trade's outcome, are brokers. For any trade that is placed, a small fee (the spread) is charged that goes directly to the broker, allowing them to make money even if a trade loses. Other groups that might make money are large institutions, such as international banks, companies, and even governments. These agents can operate on a scale that most individuals cannot fathom, possessing the power to, at least temporarily, shift market prices in their favor.

To illustrate this, we conducted a simple simulation where a powerful agent, representing a large bank, traded with fifty smaller agents. Results illustrate that an astute bank will quickly change behavior when the market does not yield profits. However, when individual traders are less sophisticated than the bank, the bank takes profits from individual traders. Details about these simulations can be found in the Appendix.

C. Reaction 3: Perhaps we did not use the right ML model, input features, or implementation

The third possible reaction to our results is that we (the authors) simply did not choose the correct ML model, input features, or implementation. Perhaps a bigger and/or better model exists or could be created that would perform better. This is certainly a possibility. For example, perhaps an effective deep reinforcement learning (RL) algorithm could be found, as none of the algorithms used in our study used RL. We chose not to include such an algorithm for three main reasons. First, multi-agent RL agents are known to be somewhat unstable. Combined with noisy and non-stationary markets, we are doubtful RL models would consistently perform well. Second, training times for a sufficiently large deep RL algorithm in this domain are very long and require large amounts of data to converge to intelligent behavior. Third, known dedicated efforts to create RL algorithms for forex day trading have not yet yielded consistently effective results. For example, [38] and [39] show negative or minimal (statistically insignificant) profits, while [40] show results that are positive, but only in a single year. As was shown in our studies, positive results in a single year do not indicate general success. Additionally, as far as we can tell, experiments reported in these papers did not include broker fees, such as spread, day fees, or both.

While a different model we did not evaluate could certainly be constructed, we would like to offer insights into what properties such an algorithm would need. First, a successful day-trading ML model should be sample efficient (i.e., not *data hungry*). A common issue in financial domains is a lack of abundant data. Most brokers offer roughly ten to fifteen years of historical data which, depending on trade entry rules, is often not enough to train large models. Additionally, because the market can change significantly from year to year and over time, data from many years ago is unlikely to accurately reflect current market conditions.

Second, a successful day-trading ML model needs to be able to generalize to scenarios not represented by the data. Because today's market tends to be different than past markets, a model must do more than just generalize from historical data – it must have a minute understanding of how current conditions drive the behaviors of other traders, brokers, institutions, etc. For example, if some new RL model became temporarily profitable, it is likely that other day traders would adopt it or a similar model. The RL algorithm would then need to adapt to this new market condition. Simulated data, predicting future behavior and conditions, could potentially help ML algorithms overcome such issues. However, obtaining reliable simulators with such accurate predictions seems fraught with challenges.

Third, related to Reaction 1, our results suggest that a profitable algorithm should focus on the quality of trades over the quantity. Placing fewer trades, and only risking money when highly confident in the outcome, might be effective.

VII. Delimitations

The experiments and results presented in this paper are only for day trading in the foreign exchange (forex) market and are applicable only to individual traders that cannot exert market power (i.e., not large organizations). Our results do not cover other markets, nor do they address medium/long-term investing strategies. We also emphasize our focus on trading performance rather than metrics like predictive accuracy, etc. Additionally, we note that our results are empirical and do not offer concrete proof regarding the difficulties of machine learning in forex day trading.

VIII. Conclusion

In this paper, we implemented algorithms from four genres of machine learning algorithms to day trade in the foreign exchange (forex) market. We created a simulator to represent market conditions as realistically as possible. Empirical results indicate that, over ten years of simulations, most ML algorithms perform as well as random. We believe there are a few reasons for these results: (1) it is extremely difficult to make accurate predictions of the future in a noisy zero-sum market that changes over time, (2) trades incur various fees (regardless if they win or lose), and (3) individual traders lack resources and market power that larger agents possess. The results do show that at least one algorithm, AlegAATr, performed on par with the minimax outcome. Accordingly, it could shed insights on potentially successful machine learning approaches.

IX. Supplementary Material

The technical appendix and the code used in the experiments can be found at https://github.com/ethanp55/alegaatr_forex/tree/ICMLA2025-SM.

References

[1] R. Rosati, L. Romeo, C. A. Goday, T. Menga, and E. Frontoni, "Machine learning in capital markets: decision support system for outcome analysis," *IEEE Access*, vol. 8, pp. 109 080–109 091, 2020.

[2] M. Kunwar, "Artificial intelligence in finance: Understanding how automation and machine learning is transforming the financial industry," 2019.

[3] S. Mankad, G. Michailidis, and A. Kirilenko, "Discovering the ecosystem of an electronic financial market with a dynamic machine-learning method," *Algorithmic Finance*, vol. 2, no. 2, pp. 151–165, 2013.

[4] R. R. Trippi and D. DeSieno, "Trading equity index futures with a neural network," *Journal of Portfolio management*, vol. 19, pp. 27–27, 1992.

[5] M. Harries and K. Horn, "Detecting concept drift in financial time series prediction using symbolic machine learning," in *AI-CONFERENCE-*. World Scientific Publishing, 1995, pp. 91–98.

[6] X. Zhang, "Non-linear predictive models for intra-day foreign exchange trading," *Intelligent Systems in Accounting, Finance and Management*, vol. 3, no. 4, pp. 293–302, 1994.

[7] M. Shubik, "A game theoretic approach to the theory of money and financial institutions," *Handbook of monetary economics*, vol. 1, pp. 171–219, 1990.

[8] C. Y. Huang, "Financial trading as a game: A deep reinforcement learning approach," *arXiv preprint arXiv:1807.02787*, 2018.

[9] J. A. Jankovsky, "Time compression trading: Exploiting multiple time frames in zero sum markets," 2010.

[10] O. Groette, "Is the stock market a zero sum game? (what about trading?)," *Quantified Strategies* – https://www.quantifiedstrategies.com/is-the-stock-market-a-zero-sum-game/, 2024, accessed: 2023-12-02.

[11] J. C. Bogle, *Enough: True measures of money, business, and life*. John Wiley & Sons, 2010.

[12] Ramsey, "What is day trading and how does it work?" *Ramsey* – https://www.ramseysolutions.com/retirement/what-is-day-trading, 2023, accessed: 2024-02-23.

[13] J. D. Coval, D. Hirshleifer, and T. Shumway, "Can individual investors beat the market?" *The Review of Asset Pricing Studies*, vol. 11, no. 3, pp. 552–559, 2021.

[14] F. Chague, R. De-Losso, and B. Giovannetti, "Day trading for a living?" *Available at SSRN 3423101*, 2020.

[15] J. Russell, "Reasons why forex traders lose money," *the balance* – https://www.thebalancemoney.com/why-do-forex-traders-lose-money-1344936, 2022, accessed: 2023-11-13.

[16] E. Norris, "Day trading schools and courses," *Investopedia* – https://www.investopedia.com/articles/active-trading/061214/best-daytrading-schools.asp, 2024, accessed: 2024-02-28.

[17] P. Gradzki and P. Wojcik, "Is attention all you need for intraday forex trading?" *Expert Systems*, vol. 41, no. 2, p. e13317, 2024.

[18] J. P. Hespanha, *Noncooperative game theory: An introduction for engineers and computer scientists*. Princeton University Press, 2017.

[19] M. Fisichella and F. Garolla, "Can deep learning improve technical analysis of forex data to predict future price movements?" *IEEE Access*, vol. 9, pp. 153 083–153 101, 2021.

[20] P. Auer, "Using confidence bounds for exploitation-exploration trade-offs," *Journal of Machine Learning Research*, vol. 3, no. Nov, pp. 397–422, 2002.

[21] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire, "The nonstochastic multiarmed bandit problem," *SIAM journal on computing*, vol. 32, no. 1, pp. 48–77, 2002.

[22] D. P. De Farias and N. Megiddo, "Combining expert advice in reactive environments," *Journal of the ACM (JACM)*, vol. 53, no. 5, pp. 762–799, 2006.

[23] E. Pedersen and J. W. Crandall, "AlegAATr the bandit," in *Proceedings of the* 26^{th} *European Conference on Artificial Intelligence*, 2023, pp. 1867–1874.

[24] L. Munkhdalai, T. Munkhdalai, K. H. Park, H. G. Lee, M. Li, and K. H. Ryu, "Mixture of activation functions with extended min-max normalization for forex market prediction," *IEEE Access*, vol. 7, pp. 183 680–183 691, 2019.

[25] J.-F. Chen, W.-L. Chen, C.-P. Huang, S.-H. Huang, and A.-P. Chen, "Financial time-series data analysis using deep convolutional neural networks," in *2016 7th International conference on cloud computing and big data (CCBD)*. IEEE, 2016, pp. 87–92.

[26] T. Fischer, M. Sterling, and S. Lessmann, "Fx-spot predictions with state-of-the-art transformer and time embeddings," *Expert Systems with Applications*, vol. 249, p. 123538, 2024.

[27] E. Fix and J. L. Hodges, "Discriminatory analysis. nonparametric discrimination: Consistency properties," *International Statistical Review/Revue Internationale de Statistique*, vol. 57, no. 3, pp. 238–247, 1989.

[28] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[29] L. Noriega, "Multilayer perceptron tutorial," *School of Computing. Staffordshire University*, vol. 4, no. 5, p. 444, 2005.

[30] L. Breiman, "Random forests," *Machine learning*, vol. 45, pp. 5–32, 2001.

[31] L. Babypips.com, "Forexpedia forex (fx)," *babypips* – https://www.babypips.com/forexpedia/forex, 2023, accessed: 2023-11-13.

[32] T. N. Y. S. Exchange, "Trading & data," *NYSE* – https://www.nyse.com/trading-data/, 2023, accessed: 2023-12-02.

[33] T. Lattimore and C. Szepesvári, *Bandit algorithms*. Cambridge University Press, 2020.

[34] X. Cao, A. Gautam, T. Whiting, S. Smith, M. A. Goodrich, and J. W. Crandall, "Robot proficiency self-assessment using assumption-alignment tracking," *IEEE Transactions on Robotics*, vol. 39, no. 4, pp. 3279–3298, 2023.

[35] O. Corp., "Rest-v20," *Oanda* – https://developer.oanda.com/rest-live-v20/introduction/, n.d., accessed: 2023-11-16.

[36] R. Snow, "How central banks impact the forex market," *DAILYFX* – https://www.dailyfx.com/education/forex-fundamental-analysis/how-central-banks-impact-forex.html, 2022, accessed: 2022-10-04.

[37] B. Graham, "The intelligent investor," 2017.

[38] J. Carapuço, R. Neves, and N. Horta, "Reinforcement learning applied to forex trading," *Applied Soft Computing*, vol. 73, pp. 783–794, 2018.

[39] S. Sornmayura, "Robust forex trading with deep q network (dqn)," *ABAC Journal*, vol. 39, no. 1, 2019.

[40] A. Riva, L. Bisi, P. Liotet, L. Sabbioni, E. Vittori, M. Pinciroli, M. Trapletti, and M. Restelli, "Addressing non-stationarity in fx trading with online model selection of offline rl experts," in *Proceedings of the Third ACM International Conf. on AI in Finance*, 2022, pp. 394–402.