

The Ephemeral Threat: Assessing the Security of Algorithmic Trading Systems powered by Deep Learning

Advije Rizvani
advije.rizvani@uni.li
Liechtenstein Business School
University of Liechtenstein
Vaduz, Liechtenstein

Giovanni Apruzzese
giovanni.apruzzese@uni.li
Liechtenstein Business School
University of Liechtenstein
Vaduz, Liechtenstein

Pavel Laskov
pavel.laskov@uni.li
Liechtenstein Business School
University of Liechtenstein
Vaduz, Liechtenstein

Abstract

We study the security of stock price forecasting using Deep Learning (DL) in *computational finance*. Despite abundant prior research on vulnerability of DL to adversarial perturbations, such work has hitherto hardly addressed practical adversarial threat models in the context of DL-powered *algorithmic trading systems* (ATS).

Specifically, we investigate the vulnerability of ATS to adversarial perturbations launched by a realistically constrained attacker. We first show that existing literature has paid limited attention to DL security in the financial domain—which is naturally attractive for adversaries. Then, we formalize the concept of *ephemeral perturbations* (EP), which can be used to stage a novel type of attack tailored for DL-based ATS. Finally, we carry out an end-to-end evaluation of our EP against a profitable ATS. Our results reveal that the introduction of small changes to the input stock-prices not only (i) induces the DL model to behave incorrectly but also (ii) leads to the whole ATS to make suboptimal buy/sell decisions, resulting in a worse financial performance of the targeted ATS.

CCS Concepts

• Computing methodologies → Machine learning; • Security and privacy → Systems security.

Keywords

Deep Learning, Computational Finance, Trading System, Attack

ACM Reference Format:

Advije Rizvani, Giovanni Apruzzese, and Pavel Laskov. 2025. The Ephemeral Threat: Assessing the Security of Algorithmic Trading Systems powered by Deep Learning. In *Proceedings of the Fifteenth ACM Conference on Data and Application Security and Privacy (CODASPY '25)*, June 4–6, 2025, Pittsburgh, PA, USA. ACM, New York, NY, USA, 12 pages. <https://doi.org/10.1145/3714393.3726490>

1 Introduction

In recent years, Deep Learning (DL) has become a popular tool in the financial sector, transforming many aspects of data analysis and decision-making processes [40, 43, 63]. One of the standout

strengths of DL models is their ability to excel in time-series forecasting, which is essential for predicting market trends and guiding investment strategies [69]. In practice, these models are now integral to comprehensive systems that automate the entire “predict–buy/sell–repeat” process, known as Algorithmic Trading Systems (ATS) [30]. ATS have profoundly reshaped the financial landscape by determining the timing, type, and volume of trades [79]. With the integration of DL, these systems (see Fig. 1) execute transactions at scales and speeds beyond human capabilities, thereby minimizing emotional biases in trading decisions [39].

However, the reliance on DL models in such critical applications is not without risks. These models are known to be vulnerable to adversarial perturbations—small, carefully crafted changes to input data that can lead to incorrect predictions [19]. In the fast-paced and competitive world of financial trading, such errors can result in serious consequences. What if an attacker can stealthily introduce small and *short-lived* perturbations to the data analysed by an ATS? Indeed, an attacker may tamper with a single data-point (the effects of which are unknown to the attacker—she cannot see the future!), but the perturbation will be “overwritten” shortly afterwards with the new (clean) data issued by the broker: a persistent influx of perturbed data would be spotted by the organization using the ATS. Such a threat model, denoting a realistically feasible attack, has never been investigated before in this domain.

Indeed, as we will show, there is a lack of works that provide an holistic perspective of the DL-specific risks to ATS. For instance, some papers studied attacks against “time-series forecasting” which is a problem that can be solved with DL and that is very relevant for ATS [32, 61]. However, the way such attacks are evaluated resembles the typical viewpoint of “adversarial machine learning papers”, i.e., the attack is deemed successful depending on how well it “fooled” the *targeted DL model* [61]. We argue that such an evaluation procedure is not only simplistic in the general sense (it is well known that DL models are a mere component of a much larger system [10]), but also that an appropriate assessment of an attack against ATS must stem from analysing *the gains/losses of the system* that occur as a consequence to the DL-model’s output.

Hence, in this work we formalize the concept of **ephemeral perturbation** (EP), which are particularly designed to attack the DL models embedded in ATS. Then, we empirically assess the effects of EP against a representative ATS, built through an original “ATS Security Framework”. We use publicly available data [6] to develop practical DL models for time-series forecasting, which are integrated into our ATS. After validating the performance of our ATS and ensuring it demonstrates a reasonable level of profitability under typical market conditions, we proceed to introduce various EP to the system. We assess the impact of these perturbations by

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
CODASPY '25, Pittsburgh, PA, USA

© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 979-8-4007-1476-4/2025/06
<https://doi.org/10.1145/3714393.3726490>

considering their effects on both (i) the DL models in isolation and (ii) the overall ATS. Our findings reveal that EP can subtly degrade the profitability of the ATS. Intriguingly, the degradation can be so small that the ATS still leads to a gain—but *inferior* to the gain without any EP: hence, the targeted organization will persist in using the ATS, thereby (unknowingly) *decreasing their returns* over time.

CONTRIBUTIONS. This paper explores the security risks associated with DL-powered ATS, a critical yet underexplored area in both security and financial computing literature. Specifically, we:

- highlight the oversight of financial applications in research focused on security of artificial intelligence (§2);
 - introduce and formalize the concept of ephemeral perturbations (EP), which can realistically affect DL-based ATS (§4);
 - propose a framework for security assessment of ATS (§3) which we use to study the impact of EP on an exemplary ATS (§5, §6).
- We also discuss and validate our findings with an user study with experts (§7). To foster future research, we release our resources [1].

2 Related Work

We first define the problem space of our paper by summarizing prior related work (§2.1), and then carry out a systematic literature review wherein we pinpoint the research gap tackled by this paper (§2.2).

2.1 Background

Our paper tackles two orthogonal research fields gravitating around artificial intelligence (AI): applications of AI in finance, and security of AI. We stress that the notion of “AI” encompasses both machine learning (ML) and deep learning (DL) methods.

AI in Finance. There is a deluge of ways to use AI in finance. An early review of various AI-based techniques is the 2010 paper by Bahrammirzaee [14]. Since then, however, this field has substantially advanced, as shown by more recent summarizations [28, 36, 41]. In particular, the advent of deep learning (DL) in this domain caused a paradigm shift [43, 63] due to its demonstrated superiority over traditional machine learning (ML) or statistical techniques (for, e.g., forecasting the stock market [71]). Noteworthy applications of AI in finance, for which there is evidence of real-world deployment [30], include: algorithmic trading [34, 51], portfolio management [42]), credit scoring [37], fraud detection [56, 64]. In this paper, we focus on the former: algorithmic trading systems (ATS) are a flourishing technology in the current financial landscape. ATS can greatly benefit from AI (and, specifically, from DL) thanks to its capabilities of predicting future market values [39, 46, 79]. Such forecasts can then be used by the ATS to “quickly” make informed decisions on whether to buy/hold/sell a given commodity. There is evidence suggesting that modern ATS do, in fact, rely on the forecasting capabilities of AI [3, 4].

Security of AI. The never-ending advances of AI induced many researchers to investigate the security of learning-based algorithms. This research field, typically referred to as “adversarial machine learning” [17–19, 72], literally exploded in the last decade, after the discovery that deep neural networks could be fooled by tiny perturbations in the input data [19]. Thousands of papers have hitherto studied how learning models can be affected by such “adversarial perturbations”, which can target any given model either during its training or inference stage, and which envision threat models

assuming attackers with various degrees of knowledge and capabilities [11]. Despite all such work, however, a pragmatic solution to “adversarial attacks” has yet to be found [10]. Moreover, despite the numerous domains in which AI has found applications (in research or in practice), most prior literature on AI security considers attacks against models devoted to computer vision (e.g., [86]), as highlighted in [10]. Hence, it is difficult to gauge how much previously proposed attacks can “practically” affect models embedded in other types of systems. The only way to do so is by formalizing specific threat models and assessing their effects.

2.2 Research Gap (Security of AI in Finance)

Prior works have investigated various security aspects of computational finance (e.g., in decentralized settings or blockchain [9, 38, 75]), and some papers are related to our focus on automated trading (e.g., [16, 85]). However, the security of AI-specific applications in finance has not been adequately scrutinised by prior work. This highlights a research gap that we aim to fill with our paper.¹

To provide evidence of such a gap, we first turn the attention to a recent work [10] revealing that, in 2019–2021, only two papers on “adversarial machine learning” (published in top-tier security venues) considered financial data [59, 60]. However, none of these works focused on the financial domain: both [59, 60] used financial data as a yet-another benchmark to validate some well-known attacks, but do not make considerations on how much the corresponding attacks may affect the owners of the targeted model. Yet, to provide a compelling contribution emphasizing that the AI-specific security aspects in finance have not been well studied, we carry out a comprehensive analysis of prior works.

2.2.1 Systematic Literature Review (top-tier venues). We begin by systematically reviewing prior works that have been published in top-tier security conferences (similarly to, e.g., [10, 12]). First, we collected the 7,266 papers that appeared in 9 top-tier venues (ACM CCS and AsiaCCS, IEEE S&P and EuroS&P, FC, ACSAC, ESORICS, NDSS, USENIX SEC) held 2014–2023; we only considered full papers (and not, e.g., short or workshop papers). Then, we inspect the abstract of all these publications. The goal is identifying candidates that may be related to “security of AI applications in finance”. We disentangle this goal by defining three criteria:

- **AI-related:** we consider a paper to be (potentially) about “AI applications” if the abstract of the paper mentions at least one of the following terms: “machine learning”, “artificial intelligence”, “deep learning”, “AI”, “ML”, “DL”.
- **Finance-related:** we consider a paper to be (potentially) about “finance” if the abstract of the paper mentions at least one of: “finance”, “econom”, “trading”.
- **Security-related:** we only considered security-related venues, so we are confident that the paper is about security. However, given that the security of AI is typically associated with the term “adversarial” (e.g., “adversarial perturbations / examples / ML”), we consider a paper to be within our scope if the term “adversarial” is mentioned at least once in the abstract.

¹We stress that the field of “market manipulation” [78] is orthogonal to our focus: our goal is to attack an ATS which leads to losses “only” to its owners; whereas market manipulation focuses on inducing changes that affect a wide-array of target groups.

We develop a script to carry out this keyword search. Overall, we find only two papers that have matches for each criteria, namely: [44, 52]. Finally, we manually review each of these papers [44, 52] to ascertain whether it truly deals with the security of AI in finance (with a focus on ATS). This process was done by two authors who independently reviewed each paper and discussed their findings. We found that none of these works can be considered to be about “security of AI in finance”, despite matching our criteria. Indeed, both [44, 52] are about generic AI security, and mention “finance” only as a potential application of AI and no finance-specific assessment is carried out. Notable examples of excluded papers (which do not meet all three criteria) are, e.g., [83], which deals with AI in finance (despite not mentioning any AI-related term in the abstract) and mentions “adversarial” to denote generative-adversarial networks, which is just a yet-another application of AI (i.e., it is not used to indicate a security assessment of AI); and [85], which is about security in finance and mentions “adversarial”, but is not about AI-specific security risks.

2.2.2 Extended Literature Review. Such a “negative result” inspired us to carry out a larger investigation on Google Scholar, which we do by following three steps (which we perform once in Jan. 2024, and a second time in May 2024 for validation):

- (1) *Broad Search.* We search for *peer-reviewed papers* (excluding, e.g., preprints) that match queries relevant for our scope. Specifically, we consider: “adversarial perturbations algorithmic trading”, “deep learning stock forecasting adversarial attack”, and four queries corresponding to (“adversarial attack” $\wedge$ (“financial forecasting” $\vee$ “algorithmic trading” $\vee$ “high-frequency trading” $\vee$ “portfolio management”)). Each of these six searches returned >10k results.
- (2) *Preliminary Screening.* For each search query, we inspect the returned papers to verify if the search terms occur in the considered paper. We inspected 390 papers in this way, i.e., until page 10 of the results returned by Google Scholar. (A preliminary check showed any paper beyond page 10 did not contain all the terms in the query.) Ultimately, we found that only 17 papers contained *all terms* specified in the respective query.
- (3) *Detailed Filtering.* We further analyse the 17 papers obtained from the previous step, to determine if they truly addressed ATS security. To this end, one author reviewed each paper and shared the corresponding conclusions with another author who acted as a validator. After this inspection, we identified only 6 papers which are relevant to security analysis of AI for ATS (e.g., we removed [47] because its contributions have no connection with security). We then applied the snowball method [80] on these 6 papers, looking for works that cite (or are cited by) them but our results did not change.

These six papers are: [23, 25, 32, 33, 57, 61]). Despite providing significant contributions, these six papers have two important limitations from a security viewpoint:

- *Powerful Attacker:* 4 papers [23, 25, 33, 61] envision an adversary with extraordinary knowledge (e.g., they know everything about the model) or capabilities (e.g., they can arbitrarily manipulate the input data): such assumptions may not reflect a realistic scenario [10], given that the internal components of ATS are kept secret, and indiscriminate manipulation of the input samples may be deemed anomalous by the ATS which would reject the input.

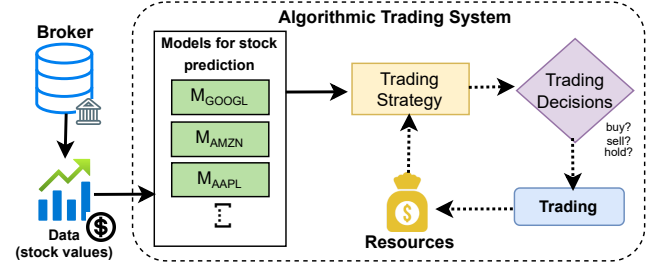


Fig. 1: Schema of an Algorithmic Trading System (ATS). The broker (e.g., a bank) sends stock-related data to a given organization which owns an ATS (dotted box). The ATS includes various DL models, used to make predictions on the basis of the input data. Such predictions are then used by the ATS to enact a given trading strategy, which must account for the available resources and decide what to do (i.e., buy/hold/sell). After making a decision, the resources are updated.

- *Model only:* 5 papers [25, 32, 33, 57, 61] consider the effects of the attack on the model in isolation—without considering the impact on the whole system (which is a recommendation by [10]). The only paper that carries out a system-wide evaluation is the work by Chen et al. [23], which considers reinforcement learning for portfolio management—which are orthogonal to ATS (our focus). Finally, we stress that only two papers [25, 61] release their source-code: this remark motivates our first technical contribution (§3).

Takeaway. Despite the increasing usage of AI in ATS, the system-wide impact that adversarial perturbations can have on AI-based ATS is still unclear. Our contributions seek to provide a foundation for future work in this domain.

3 ATS Security Framework (ATS-SF)

To the best of our knowledge (and as shown in §2), there is no publicly available and open-source implementation of an ATS that can be used to carry out security assessments. To fill this gap and promote future work in this domain, we propose our “ATS Security Framework,” ATS-SF, which we describe in this section.

Our ATS-SF framework seeks to enable the development of exemplary ATS which can be used by researchers to simulate the workflow of a full-fledged ATS for security assessments. We provide a schematic of our envisioned ATS in Fig. 1, inspired by [3, 45, 67]. The ATS (i.e., the rounded dotted box) receives data (i.e., the values of a certain set of stocks—e.g., GOOGL or AMZN) from a given broker (e.g., a bank). Such data is then automatically analysed by various models, which are collectively tasked to predict the “future” values of the stocks included in a given portfolio. The output of such models is then used by the ATS according to a given trading strategy, which will induce the ATS to make trading decisions (i.e., buy/hold/sell) on the stock market. Ideally, the decisions made by an ATS should yield a profit to its owners—either in the short or long term. Such a profit can be measured either via (a) the “cumulative daily returns”, which aggregates the gains/losses obtained after each day of use of the ATS [8, 22]; or via (b) the “Sharpe Ratio” [66, 74], which measures the performance of an investment by adjusting for its risk, providing a ratio of return to volatility [70].

Remark. Our framework assumes that the ATS makes its decisions based on historical prices only. We are aware that there may be additional information sources (e.g., news and social media) that can be used to guide such trading decisions [20, 24, 29, 35]. Our implementation of ATS-SF, being open source, can be extended to account also for such data—but we leave development of such an enhancement to future work.

3.1 Low-level Architecture and Functionalities

The design of our **ATS-SF** is centered around three core environments: Asset, Model, and Trade (summarized in Fig. 2). Such a design choice allows one to understand the effects of any change (potentially caused by an “attack”) to each environment—enabling a system-wide assessment. Let us explain how we devise our proposed **ATS-SF**, for which we must introduce some notation.

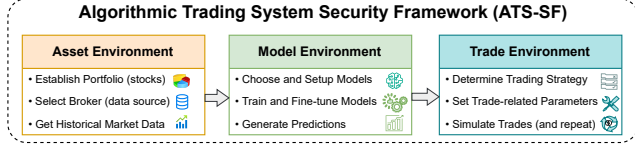


Fig. 2: Architecture of our ATS-SF. Our framework has three environments that allow fine-grained control on the entire management pipeline of an ATS, thereby enabling security assessments.

Asset Environment. First, the ATS assumes that its owners have specified a given portfolio, which we denote with $\mathcal{P}$. Such a portfolio is identified by a set of stocks; let s denote each stock in a portfolio, so that $\cup_s = \mathcal{P}$. There exist various ways that can be used to select/optimize the stocks included in any portfolio (e.g., [42, 73]). Obviously, the ATS expects to receive, as input, data (i.e., market values) pertaining to the stocks in $\mathcal{P}$ (obtainable from either public sources, such as Yahoo Finance [6]; or from private brokers). Such data can come in many forms in terms of *features* (e.g., opening or closing price of a given stock) and *frequency* (e.g., the values can arrive on a daily basis, or even more often—the latter could be used for high-frequency trading[68]). Our proposed **ATS-SF** enables the developer to freely choose any of these options.

Model Environment. Then, the ATS must analyse the input data and predict future values. To enable a wide-range of flexibility, we have realised **ATS-SF** so that it is *model agnostic*: the predictions can be made, e.g., via deep learning methods (such as Long-Short Term Memory neural networks, or LSTM), or via traditional time-series forecasting techniques (e.g., Autoregressive Integrated Moving Average, or ARIMA), or even via other types of learning algorithms (tailored for regression). The developer has complete choice on how to implement such models: they can, e.g., perform multivariate analyses for each stock—such as using the opening and closing price to predict the closing price; or they can have a single model that takes as input the values of all the stocks in the portfolio, and predict a value for each stock. Of course, depending on the adopted choices, more operations may be required—such as *training* the model(s) to fine-tune their parameters and optimize their performance. In what follows, we will use M_s to denote a model that is tasked to predict the values of stock(s) s .

Trade Environment. Finally, the ATS must make its trading decisions according to a given trading strategy (e.g., [62]), which must account for the output of the models M_s for all stocks $s \in \mathcal{P}$, the available resources (the ATS cannot “buy” any stock if there is no capital left), and any other contextual information (e.g., the historical trend of the stocks). The trading decisions are determined by providing rules that dictate the “buy/sell/hold” signals (i.e., if a buy signal is generated for a given stock s and there is sufficient available capital, the ATS will buy s). We also enable the developer to specify the portion of the portfolio used for the transaction (helpful for risk management). When making any given trade, the

ATS accounts for the transaction fee [55] as well as the slippage cost—both of which are user-specified. After the trade is made, our **ATS-SF** updates the available resources, accounting for gains and losses from the previous transactions as well as from the new values of the stocks in the portfolio.

The entire procedure (i.e., receiving the input data, making the predictions, determining the trading decisions, executing the trades, updating the resources) is automatically repeated by our **ATS-SF** unless specified otherwise. At the end of the simulation, our **ATS-SF** will display the overall results of the ATS by showing: the predictions of each model, including how much they differ from the actual values—measurable, e.g., via the root mean squared error (RMSE); and the daily returns and Sharpe Ratio of the ATS.

Disclaimer. **ATS-SF** is meant to *simulate* trading decisions over a pre-defined timespan, and does not guarantee that any parameter configuration will yield a net profit from the corresponding ATS (it is up to the developer to find the optimal configuration). Moreover, we do not claim that relying on our **ATS-SF** can lead to one gaining money from replicating the exact trading decisions—even if the results of the simulation show that the ATS consistently exhibits a remunerative Sharpe Ratio or daily returns. Hence, we *do not* endorse using **ATS-SF** for real trading!

3.2 Custom ATS: Implementation & Evaluation

We used our proposed **ATS-SF** to develop our custom ATS, which we will use as a baseline for our security assessment. We first describe how we developed the models and then explain how we developed the overarching ATS. Finally, we report the baseline performance of our prototype.

AI-models development. First, we collect data from Yahoo Finance [6], and we consider a timespan between Jan. 2010 and Oct. 2023. Then, we consider a portfolio $\mathcal{P}$ of 38 stocks, which is an holistic representation of the stock market. For each stock $s \in \mathcal{P}$, we develop a specific model, M_s ; hence, our ATS encompasses 38 models. Each model relies on LSTM (due to their recognized capability for similar tasks [31]). To develop each model, we perform a temporal cut to our data by applying an 80:20 split (a common practice [50]), identifying the training/testing partitions; moreover, each LSTM is tasked to predict the next “close price”, done by means of a multivariate time-series analysis of 5 features (“high”, “low”, “open”, “close”, “volume” – all used in, e.g., [27]) analysed over a temporal window of $w = 50$ days (used, e.g., in [54]); we assume trades done on a daily basis. We then train all our models on the training set, trying out various configurations to optimize their performance (e.g., changing number of layers, or neurons per layer) and measure their performance on the test set: the RMSE (over the entire test set) is 4.03 on average (slightly better than that achieved by [50]), confirming the quality of our models. We report the stocks in our portfolio and the RMSE achieved by each LSTM in Table 1.

Table 1: Stocks considered in the portfolio of our custom ATS. For each stock, we report the RMSE achieved by the its model over the test period. (Avg RMSE=4.03)

Stock	RMSE	Stock	RMSE	Stock	RMSE	Stock	RMSE	Stock	RMSE	Stock	RMSE
GOOGL	6.36	AMZN	4.52	AAPL	4.55	BP	1.17	BA	6.83	MMM	3.23
PEP	3.48	JNJ	2.50	PFE	1.14	HON	10.20	GE	1.87	T	0.44
MRK	3.34	ABBV	5.14	PG	3.60	VZ	0.85	TMUS	2.89	HSY	9.62
KO	0.97	WMT	1.47	JPM	3.32	DUK	2.27	SO	1.98	EXC	1.17
BAC	1.11	GS	14.11	V	9.85	AEP	1.70	AMT	9.27	PLD	3.49
XOM	2.95	CVX	4.34	COP	3.37	SPG	3.56	BHP	2.19	RIO	2.26
VALE	0.69	FCX	1.75								

Connecting the models to the ATS. After we have developed each model, we combine them together inside the ATS. To this end,

we consider the well-known “moving average crossover” trading strategy [13] which compares the predicted value (of the following day) with the previous predictions of each stock $s \in \mathcal{P}$ to generate the trading decisions. Specifically, we use the previous predictions to calculate the moving average in the short-term (5 days) and in the long-term (20 days): a buy (sell) signal is given when the short-term average crosses above (below) the long-term average; otherwise, no signal is generated, defaulting to a hold decision. Depending on the signal, a trade can be made if there are enough resources: for our simulation, we set an initial capital of 100,000\$ (common [66, 68]), and we specify a transaction cost of 0.005\$ per share [2]; the slippage cost is set at 0.02 to account for the difference between the expected and actual execution prices [53]. Furthermore, to help managing risk and ensuring diversification whenever a trade is made, only a fixed portion of 10% of the current portfolio value is used. If no resources are available (i.e., if there is not enough capital for a “buy” signal, or there are no stocks for a “sell” signal), a signal is ignored. After each trading day, the portfolio is evaluated to calculate the daily returns, reflecting the changes (in percentage) of the portfolio value w.r.t. the previous day. Similarly, the Sharpe Ratio is calculated by comparing the excess return of the portfolio with its volatility, thereby evaluating the risk-adjusted performance of the portfolio; to this end, we set the risk-free rate to 5.05% (given that our simulation spans over ≈ 2 years, and the 2-year treasury rate was 5.05% according to [5] on Oct. 23rd, 2023). Finally, to prevent lookahead bias, the trading signals are shifted backward by one day, so that they are based only on data available up to that day [15, 77].

Performance Evaluation. We let our ATS simulate trades for the entire test window: from Dec. 2021 to Oct. 2023 (recall that we partition our 13 years of historical market data in train:test with a 80:20 split). We report the baseline performance of our ATS in Fig. 3. Specifically, Fig. 3a shows the aggregated performance of the LSTM models (for all 38 stocks in the portfolio), denoting that, overall, our models are accurate at predicting the closing price. Then, Fig. 3b shows the cumulative daily returns over the entire ATS operation time, from which we see that the ATS is generating a profit of approximately 25% over two years. Finally, Fig. 3c shows the Sharpe Ratio: after the first days (which denote some stark fluctuations due to quick distribution of resources in the capital), the Sharpe Ratio of the ATS stabilizes and consistently remains above 0, denoting that the decisions made by the ATS are, ultimately, benefiting its potential owners—at least according to our simulation.

Takeaway. Our ATS has an appreciable performance, which would justify its deployment in the real world (further validated by our user study in §7.1). Such a property makes our ATS a good test subject for our security assessment.

4 Proposed Threat Model

Recall (§2.2) that prior related work envisioned threat models assuming powerful adversaries. As a result, the corresponding attacks were often very effective—at least from a “model only” perspective (e.g., the FGSM attack described in [32] degraded the model’s performance by reducing its accuracy from 95% to 60%).

While such scenarios are not impossible, we argue that “some” real attackers may rely on easier and more subtle strategies to

exploit the AI models deployed in ATS. We explain our vision by proposing our original threat model—and then compare it to the “adversarial attacks” envisioned in prior works (§4.4).

4.1 Target System

The system (i.e., the “defender”) targeted by the attacker resembles the exemplary ATS described in §3. For completeness, we formalize its key points here.

We consider an ATS whose decisions are driven by data received by a given broker—which is considered to be trusted. Such data is in the form of historical market-related values (e.g., closing price, opening price), which pertain to a portfolio ($\mathcal{P}$) encompassing a certain set of stocks; without loss of generality, we assume a daily frequency of broker-ATS communications [23]. The ATS integrates models tasked to predict the future values of a specific stock $s \in \mathcal{P}$, so that M_s is the model devoted to predicting the values of s ; without loss of generality, we assume that each stock has a specific model.

A model M_s produces an output by analysing the previous w values of stock s in a time-series fashion: given a point-in-time t , M_s returns the prediction of $t + 1$ as: $M_s(t + 1) = f(s, t, w)$, where f is the function learned by M_s during its training phase; ideally, $M_s(t + 1)$ should approximate the actual following value of s , i.e., $M_s(t + 1) \approx s_{t+1}$ (measurable via, e.g., RMSE [76]). The ATS takes all predictions of each model into account and then, depending on a given trading strategy and available resources, will enact certain trading decisions (e.g., buy/sell/hold any given stock). It is implicitly assumed that the ATS is expected to yield a profit to its owners (otherwise, they would not use it in the first place!).

4.2 Envisioned Attacker

Our attacker has limited knowledge and capabilities, which implicitly increase the real-world likelihood of our threat model [10].

- **Knowledge.** The attacker knows that the targeted ATS analyzes the historical market-data sent by the broker. However, the attacker does not know the entire portfolio considered by the ATS. Specifically, the attacker only knows a single stock $s' \in \mathcal{P}$ (for instance, assuming the ATS considered in our implementation which has a portfolio with the 38 stocks in Table 1, the attacker will only know, e.g., that $\mathcal{P}$ includes GOOGL). Moreover, the attacker lacks any knowledge on the models (including $M_{s'}$) integrated in the ATS, including the length of the analysis period w .
- **Capabilities.** The attacker can introduce some perturbations by manipulating the communications between the broker and the ATS (this is doable, e.g., via a man-in-the-middle attack—which are feasible in this context [61]). However, to avoid being detected, such perturbations must be “small” (if the price of one stock is substantially different, the ATS may reject the input [7]) and also “short-lived,” i.e., the attacker cannot send perturbations over many days (a single “small” deviation can be acceptable, but the owners of the ATS would react if they constantly receive wrong data from the broker). The attacker has no access to the ATS (which could be used for, e.g., query-based attacks [10]).

Our attacker has one goal: induce the targeted organization to *gain less money*. This is a realistic assumption: given the competitive nature of the financial market, an adversary may want to gain an advantage by reducing the earnings of their competitors (even if such competitors keep gaining money). From the viewpoint of the

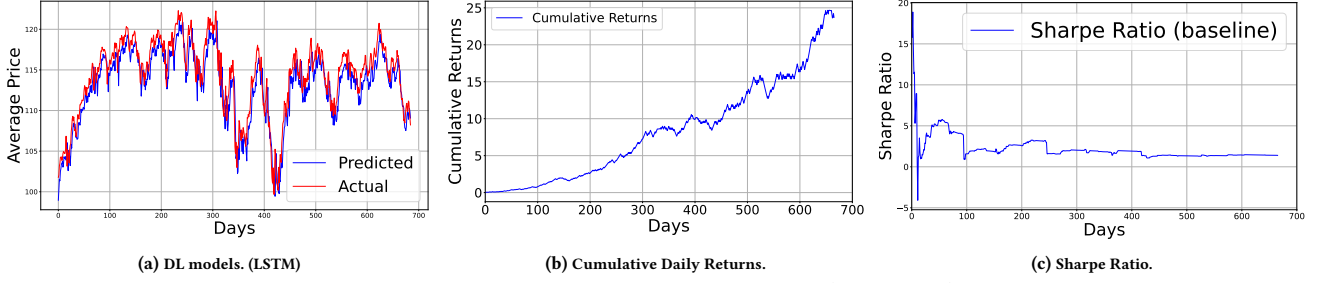


Fig. 3: Baseline ATS. We show the profitability of our self-developed ATS. The LSTM models effectively predict (avg RMSE=3.89) the future close price of the stocks in the portfolio (Fig. 3a). The ATS uses these predictions for its trading strategy, leading to trades that net a profit over-time, shown by increasing cumulative daily returns (Fig. 3b) and underscored by the Sharpe Ratio consistently above 0 (Fig. 3c).

ATS, such an effect can be measurable, e.g., by a Sharpe Ratio that, despite being inferior to the baseline value, is still above 0.

4.3 Strategy: Ephemeral Perturbations

To reach their goal, our attacker has only one option: “guess” an *ephemeral adversarial perturbation* (EP). Such an EP must be small enough to avoid raising suspicion—in the short-term (to avoid immediate reactions by the target organization) and in the long-term (if the ATS profitability drops excessively, then the organization would replace/stop using it). We formalize this strategy.

The attacker manipulates the values of the stocks that they know are analysed by the ATS, $s' \in \mathcal{P}$, which will likely lead to the corresponding model ($M_{s'}$) to output a different result, which may (or may not) induce the ATS to make a suboptimal trading decision. Our EP should last only a single time-point (i.e., only one day). Therefore, using an EP requires the attacker to make two choices:

- “when” the EP should be used—i.e., which day t to attack,
- “how large” the EP should be—i.e., what is the “adversarial” value² received by the ATS (and which will be analysed by $M_{s'}$) for the perturbed stock $s' \in \mathcal{P}$.

Hence, we can formally express an EP as a function:

$$\text{EP}(s, t, m) = s'_t \neq s_t \quad (1)$$

where s_t is the value of s at time t , s'_t is the adversarial value of s_t after the application of the EP, and m is a parameter which regulates the *magnitude* of the perturbation.

4.4 Comparison with prior threat models

We have already established that, at a high-level, prior works that considered “AI-specific attacks” in the context of computational finance have limitations (§2.2). Here, we perform a low-level comparison of our proposed threat model with respect to those envisioned in the ten works we found during our literature review, namely: [23, 25, 32, 33, 44, 52, 57, 59–61].

First, we observe that our attacker shares some traits with the “myopic” attacker proposed in [11], for which the targeted ATS is an “invisible” ML system [10]. This is a crucial observation: our attacker *cannot query the targeted model*, and *does not know anything about the targeted model*. Such a consideration automatically puts our attacker in a different league than “white-box” attackers that rely on knowledge of the learned model gradients to craft an adversarial example (this is done, e.g., in [23, 32, 33, 44, 57, 59–61]). Moreover, even “black-box” attacks (envisioned in [60]) which rely

²An attacker can modify s_t in any way, but ‘large’ changes (i.e., those leading to an s'_t s.t. $|s_t - s'_t| \gg 0$) would raise suspicion.

on querying the target model so as to craft a surrogate model that can then be used to apply the strategies for white-box attacks (by leveraging the transferability of adversarial examples [26]) cannot be staged by our attacker. We stress that Nehemeya et al. [61] as well as Goldblum et al. [33] also consider a “black-box” setting for their attack with no querying involved: they simply assess how much a perturbation to a given model can transfer to another model that uses a different algorithm/parameters. It is not explained (in either [33, 61]) how an attacker can, in the real-world, obtain such a surrogate model; however, we have reason to believe that it requires substantial resources (e.g., an insider of a company using the ATS can obtain some information about the models’ specifics).

Second, we observe that our attacker can only attempt a *short lived* (and small) perturbation. This is substantially different from, e.g., the attack proposed by Dang et al. [25], in which it is stated that “we suppose that the attacker is allowed to perturb all inputs at test time”. For instance, consider the models we developed for our custom ATS (§3.2): each model receives as input a time-window of $w = 50$ values. Our attacker can only change one value (i.e., the last one) whereas the attacker envisioned by Dang et al. [25] would be able to perturb all 50 values received as input by our models.

Third, our attacker can only operate during the *inference phase* of the model. In other words, our attacker is not capable of staging “poisoning attacks” which require the manipulation of data used to train the model—as done by Liu et al., and also by Nasr et al. [52, 60].

Simplicity. Compared to previously proposed attacks, ours is much more simple to realize: our attacker *knows nothing* of the targeted system, and only manipulates *one value* sent by the broker. We are not aware of any previously proposed attack that leverages a similar concept as our “ephemeral perturbations” in the financial context—motivating our upcoming evaluation.

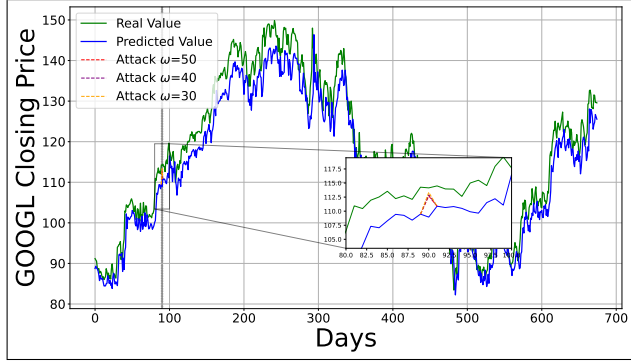
5 Security Assessment

We combine our previous two contributions, and use the ATS developed with our ATS-SF (§3) to assess the impact of perturbations stemming from our threat model (§4). We do so via two case studies, which entail either “indiscriminate” (§5.2) or “targeted” (§5.3) attacks. We first describe the common setup (§5.1).

5.1 Common Setup

Our two case studies share some assumptions. First, they both involve the same ATS, i.e., the one we developed in §3.2 (which we showed is able to generate a profit to its owners—see Fig. 3).

Recall that our attacker knows only one stock $s' \in \mathcal{P}$. For both case studies, $s' = \text{GOOGL}$ and, specifically, its closing price: this is



(a) **DL predictor.** We introduce a tiny EP (dotted line) on the closing price of GOOGL on day 90. The EP leads the LSTM, when predicting the closing price for day 91, to output a different value. The EP “disappears” on day 91.

Fig. 4: Exemplary results of an EP. We showcase what happens if a DL predictor and overarching ATS are targeted by some of our proposed EP. The blue line represents the baseline performance (y-axis), whereas the others represent the effects of various EPs (targeting the same day, but with different m) over our test timeframe (x-axis).

a sensible choice, since GOOGL is a strong asset that is likely to be included in many portfolios, and the closing price is crucial for making educated predictions at the end of the day. In both cases, the attacker also knows that the ATS receives their input data from Yahoo Finance (i.e., the broker): hence, the EP will manipulate the closing price of GOOGL provided by Yahoo Finance. Importantly, whenever we apply an EP to the clean data, we ensure that the ATS receives the EP *only* on the attacked day (t). The EP will be *deleted* the following day (because it is overwritten by the new data issued by the broker; see §4.3): hence, on day $t + 1$, the time-window analyzed by the ATS has *only correct stock values*.

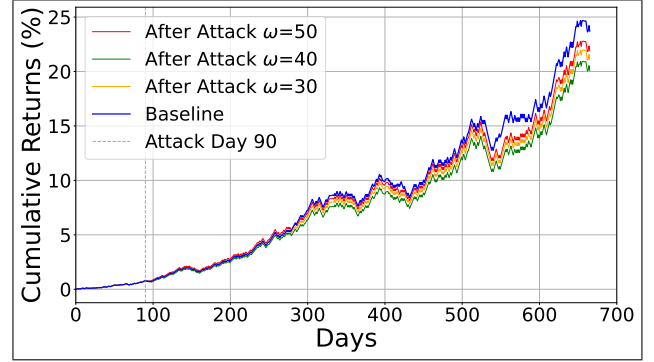
The major difference of our two case studies lies in two crucial aspects of an EP: *when* to launch an EP-based attack (i.e., how to choose t); and *how* to craft the EP (i.e., how to choose m). This will be explained in each case study.

5.2 First Case Study: Indiscriminate Attack

In this case study, we consider a “naive” attacker that does not make advanced consideration on their choice of either t or m . This is useful to examine “best case” scenarios for the defender.

Setup. To simulate this scenario in a bias-free way, we craft EP for *all days of the considered test window*. Of course, we will assess each day separately, but the intention is to get a broad understanding of the effects that an EP can have on our ATS on a “randomly chosen” day. For the magnitude, we take inspiration from [32], and consider an EP s.t. $s'_t = s_t + 2\sigma_s^\omega$, where σ_s^ω is the standard deviation of the value of stock s across the time window ω . In [32] it is assumed that $\omega = w$; however, our attacker *does not know* w . Hence, we consider three cases: $\omega = (30, 40, 50)$, i.e., one case ($\omega = 50$) wherein the attacker is ‘lucky’ and correctly guesses w ; and two cases in which the attacker makes a wrong guess. We report in Algorithm 1 (at the end of the paper) the pseudocode of our EP-crafting procedure.

Results. We craft our “indiscriminate” EP for all the 666 days of the test window. We report an exemplary effect of the EP applied to the 90th day of the test window in Figs. 4. Specifically, Fig. 4a shows how the EP affects the LSTM devoted to GOOGL: we can see that the EP induces this model to make a wrong prediction for a single day (i.e., the 90th); however, after this day, all other predictions are not affected because the EP is sanitized after the ATS receives the “fixed” data from the broker. From the perspective of the RMSE, this EP has



(b) **Whole ATS.** Effects of the EP introduced on day 90. Starting from the following day (day 91), the Cumulative Returns of the ATS drops (inducing a monetary loss) w.r.t. the baseline—despite the EP affecting only one day.

barely any impact (from 6.3692 to 6.3662, 6.3662 and 6.3652 for $\omega = 50, 40, 30$ respectively). However, the situation changes when we analyse the impact of this EP on the whole ATS, shown in Fig. 4b. We can see that, from day 91, the cumulative returns of the ATS start to deviate (i.e., they are worse) from the baseline. This is due to the ATS making a different trading decision on day 91 (due to the EP): such a decision propagates for the entire length of the test window. While this is the impact for just one day of our test window, we found that similar EP are effective in the wide majority of our considered test window: for $\omega = 50, 40, 30$ the Sharpe Ratio is lower than the baseline for 628, 646, 635 days out of 666 days, respectively—i.e., over 94% of our EP decrease the SR compared to the baseline (this is also confirmed by mostly inferior cumulative returns). These results are shown in the boxplots in Fig. 5, showing the aggregated impact (relative to the baseline Sharpe Ratio and cumulative returns) of all our EP on our ATS. As it can be seen by the notches of each boxplot (below 0 for the Sharpe Ratio, and below 1 for the Cumulative Returns), the wide majority of our EP substantially decrease the baseline profitability of the ATS.

5.3 Second Case Study: Targeted Attack

Here, we consider a more sophisticated approach: the attacker *waits* for an opportunity that would allow to craft a “meaningful” EP. This is useful to investigate worst case scenarios for the defender.

Setup. To simulate the above mentioned scenario, we consider an EP launched on days 47, 62, 495, 518, 552, 580 of the test window. On these days important *real-world events* occurred that were related to Google. For instance, on day 62 (i.e., Feb. 8, 2022) the company announced that it would pause hiring for two weeks, whereas on day 518 (i.e., Apr. 17, 2023), its AI chatbot, Bard, gave an incorrect answer in a promotional video. As a matter of fact, the value of GOOGL dropped significantly on day 63 (by 7.5%) and day 519 (by 9%). We explain our reasons for considering these six days in Table 2. On such days, a savvy attacker may find it opportune to attack an ATS. We consider two potential scenarios for targeted attacks. First, an attacker trying to “conceal” the price drop by reporting the previous value. Second, an attacker (potentially) “overestimating” the effects of the news by reporting a very high drop (with respect to the previous day), which we fix at 10% in our proof-of-concept experiment; of course, different percentages could be possible.

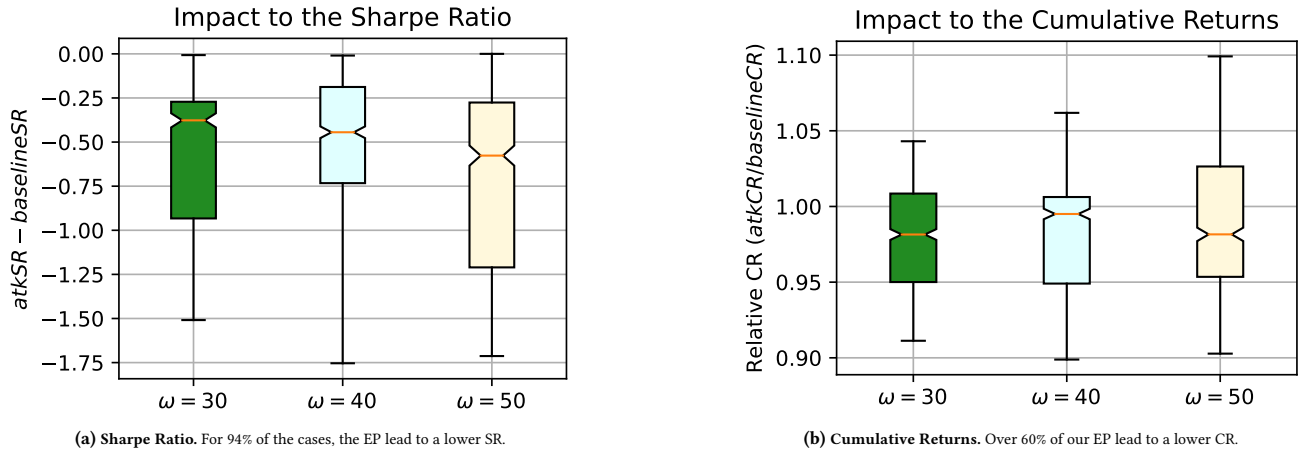


Fig. 5: Overall impact of our untargeted attacks. For each attacked day (of our 666 testing window), we compute: the difference between the Sharpe Ratio (SR) achieved by the ATS at the end of the simulation (i.e., at day 666) with the baseline SR (Fig. 5a); and the ratio between the cumulative returns (CR) achieved by the baseline ATS respect to when it is attacked by an EP. We then plot the distribution of these “impacts”. For the SR (Fig. 5a) numbers below 0 means that the SR was degraded by the EP; whereas, for the CR (Fig. 5b), numbers below 1 means that the CR was degraded by the EP. Overall, the attack is very successful: for each considered magnitude ($\omega=30,40,50$) the EP leads to a lower sharpe ratio and inferior cumulative returns in most cases. This is evident by looking at the *notches* of the boxplots (indicating the mean).

Table 2: Targeted Attacks: chosen days and explanations. We selected six specific dates (during which the ATS was operational) on which GOOGL stock prices dropped due to real-world events. The table reports the real-world price drop of the closing price of GOOGL (as reported on the following day) with respect to the attacked day.

Day	Date	Drop	Reason
47	Jan 24, 2022	7.2%	The company reported weaker-than-expected earnings.
62	Feb 8, 2022	7.5%	The company announced that it would be pausing hiring for two weeks.
495	Apr 17, 2023	9%	Its AI chatbot, Bard, gave an incorrect answer in a promotional video.
518	May 10, 2023	4.3%	The company reported weaker-than-expected earnings.
552	Jun 13, 2023	3.8%	The company announced that it would be slowing its pace of hiring.
580	Jul 11, 2023	4.1%	The company announced that it would be laying off employees.

Results. We measure the impact of the targeted attacks via the relative change (w.r.t. the baseline) in the cumulative returns (CR), shown in Table 3. The “overestimate” scenario inflicts a substantial loss to the ATS, decreasing the baseline CR by up to 28%, which makes the ATS almost unprofitable. In contrast, the “conceal” scenario is not very malignant (the CR may even increase).

Table 3: Impact of Targeted EP. CR difference (w.r.t. the baseline) for “conceal” and “overestimate” atk scenarios.

Atk Day	47	62	495	518	552	580
Conceal	2.0	-4.0	4.5	-0.5	-1.5	4.5
OverEst	-22.0	-28.0	-17.5	-23.5	-24.5	-20.0

Takeaway. We derive two lessons learned from our results. (1) *Some EP can lead to an unrecoverable loss by the targeted organization.* Our EP (which, in all cases, are launched in only one day, and involve only a tiny change of the closing price of one stock of the portfolio) lead to a worse SR w.r.t. the baseline. (2) *The effects on the ATS cannot be appreciated with model-only evaluations.* The EP affect the LSTM only for one day. Then, the effects of the EP disappear: the LSTM behaves exactly as if nothing happened—despite the ATS making the organization “gain less money” due to the EP.

6 Additional Experiments

We further enrich our security assessment by carrying additional experiments (§6.1) and providing a low-level analysis on some “negative results” (§6.2).

6.1 More Trading Strategies

In our evaluation, we used our EP to attack only one ATS (§5.1). Here, we expand our evaluation by considering two additional ATS relying on different trading strategies.

Setup. We consider the same set of 38 LSTM models used to develop our “primary” ATS (§3.2). However, instead of using the output of these models as input to an ATS that leverages the “moving average crossover” trading strategy, we consider two different trading strategies—both adopted also by prior work (e.g., [49, 82, 84]):

- *Rate of Change* [58]: this computes a reference metric called “Rate of Change” for every stock in the portfolio. By following best practices [58], we compute such a metric as follows: given a stock s , its “Rate of Change” (ROC_s^t) on day t is $ROC_s^t = \frac{s_t - s_{t-14}}{s_{t-14}} * 100$, with s_t being the value of stock s at time t . To trigger a buy signal, the ROC must be above 1; for a sell signal, the ROC should be below -1; otherwise, no decision is made (i.e., default to hold).
- *Bollinger Bands* [21]: the fundamental principle of this trading strategy is to make a decision depending on where the predicted value of a stock falls in a given reference “band”. In our case, we use the default parameters [48]: we compute the 20-day moving average of the closing price of each stock, and choose an upper/lower bound by adding/subtracting twice its standard deviation: if the prediction of the model is above the “upper” bound, then this triggers a buy signal; if it is below the “lower” bound, then this triggers a sell signal; otherwise, it is a hold.

We set the same initial conditions as in our “primary” ATS for both of these additional ATSs, and we let them run over the same test period. At the end of the simulation, in the absence of attacks, the Sharpe Ratio of both of these ATS is above 0. Specifically, for the ATS using Bollinger Bands, its Sharpe Ratio=0.26, whereas for the ATS using the Rate of Change, its Sharpe Ratio=0.72. In both cases,

however, the Sharpe Ratio is lower than the one achieved by our “primary” ATS (see Fig. 3c). In contrast, the cumulative returns of the Bollinger Bands ATS are almost negligible (-0.3%) despite its positive Sharpe Ratio (this is because this trading strategy adopts a very conservative approach); whereas for the Rate of Change they are higher (49%) than our “primary” ATS. This is because the Rate of Change is a very risky trading strategy which, in our case, “paid off”; however, such a huge risk factor leads to a lower Sharpe Ratio.

Results. Our EP are agnostic of the ATS (and of its underlying DL models), hence we test these two ATS against the exact same EP used against our “primary” ATS. The complete results are provided in Fig. 7 (for the Rate of Change trading strategy) and in Fig. 8 (for the Bollinger Bands trading strategy). To allow a high-level comparison, both of these figures have the same structure of Fig. 5. We can see some intriguing results.

- *EP vs Rate of Change.* In terms of cumulative returns, over 49% of our EP led to a net loss with respect to the baseline (see Fig. 7b). In contrast, only 20% of our EP led to a reduction of the Sharpe Ratio (see Fig. 7a), with an absolute (negative) difference that could go below -1.2 than the baseline. Such a discrepancy is due to the high-risk nature of this strategy: despite inducing a lower net revenue, the Sharpe Ratio was not excessively affected.
- *EP vs Bollinger Bands.* For the cumulative returns, 18.8% of our EP led to a net loss with respect to the baseline (see Fig. 8b). Intriguingly, however, 62% of our EP led to a drop in the Sharpe Ratio. And, in fact, the notches of the boxplots in Fig. 8a are all below the saddle point (of 0). These effects can be explained by the fact that the Cumulative Returns of this trading strategy were almost 0 without any EP, hence it makes sense that the introduction of an EP may not have excessive effects on this metric (and, actually, induce the ATS to gain more money, as remarked by the notches being over 1). While this ATS may appear to be “more robust” against EP than the others, its baseline profitability was also vastly inferior.

We can conclude that evaluating such system-wide properties of EP is fundamental to gauge their practical effects. Importantly: the two ATS considered in this expanded assessment were relying on the *exact same DL models* of our “primary” ATS. This shows that, e.g., in some cases the perturbations can have little impact (e.g., against the ATS using the Rate of Change).

6.2 Negative Result: A lesson learned

Here, we use one of the EP of our “primary” ATS to further highlight the importance of assessing the system-wide properties of adversarial perturbations (including, of course, our EP) against ATS.

Specifically, we focus on the EP that we craft for day 551 of our simulation. We showcase the effects of such an EP on the LSTM (analysing GOOGL) in Fig. 6. We can see that the predicted value of the LSTM for day 552 is around 101, but introducing the EP on day 551 leads the model to predict a value that is about 106. The value of the stock on day 551 was 105, i.e., *below* the value predicted due to the EP, but *above* the value predicted without an EP. Yet, despite the perturbation clearly leading to a wrong prediction by the considered model, such an EP *had no impact* on the overarching system—neither in terms of the Sharpe Ratio (which was 1.37, i.e., the same as the baseline), nor for the cumulative returns (which were 24% , i.e., they slightly increased from the baseline 23.7%).

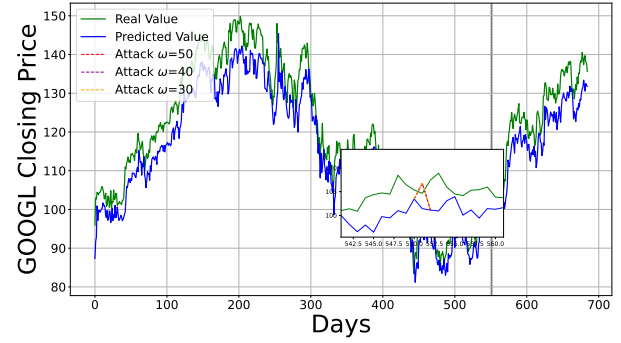


Fig. 6: Negative result. This EP, introduced for day 551, affected the LSTM model (as shown) but did not lead to any change in the profitability of our “primary” ATS.

We use such a negative result as a scaffold for two considerations.

- First, the EP led the LSTM to produce a different output—hence, from an “adversarial ML point of view”, the EP was successful. However, such an EP had no impact on the overarching ATS.
- Second, the EP induced a similar effect as those of adversarial perturbations envisioned by prior work. For instance, in [25, 61], the goal was to craft a perturbation that induced the model to output a wrong prediction in terms of buy/sell, with the logic that “if the model predicts that the price drops, then it is a sell signal—hence, my perturbation should induce the model to predict that the price increases” (and viceversa). This is a sensible goal: as a matter of fact, the trading strategies of our ATS also leverage similar rationales. However, as we showed, such an EP has no impact on our ATS—and prior works did not evaluate the impact of their perturbations on the overarching system.

The reason why this EP led to no impact is because, on that day, the ATS decided that it was better not to do anything with the GOOGL stock—defaulting to “hold” with or without the EP. For instance, the baseline system (which predicted a price drop) may not have had any GOOGL stock to sell; whereas the “attacked” system (seeing a price increase) may not have had enough resources to purchase a GOOGL stock. Moreover, even when a buy or sell signal is triggered, this does not mean that the ATS will carry out such an operation: besides available resources, there is also the risk factor to consider. Perhaps, in the case of buy, the ATS may have preferred to buy another stock for which the corresponding LSTM (which was not attacked) predicted a huge increase.

Lesson Learned. ATS are complex systems and their profitability depends on a plethora of factors. Carrying out model-only evaluations may overestimate the efficacy of a given attack. This is why we advocate future work to expand their scope and carry out system-wide assessments—which are enabled by our ATS-SF.

7 Discussion

Our paper has addressed security issues that may affect the ATS deployed by real organizations, and seeks to provide a foundation for further research in this field. Here, we reflect on our contributions by outlining the opinion of experts in the financial industry (§7.1), by discussing the limitations of our work (§7.2), and by providing additional analyses of our threat model (§7.3).

7.1 User Study with Experts

In Sept–Dec 2023, we reached out to 7 experts who work in the FinTech industry and have experience in the field of AI for finance,

including ATS. We presented our research to the experts and asked for their opinion via three questions:

- Is our ATS practical in terms of its performance?
- Does our threat model reflect a feasible scenario?
- Does our EP represent a security risk?

We acknowledge that, due to the small size of our sample we cannot claim that the answers we received fully reflect the opinion of a broad range practitioners worldwide. For confidentiality reasons, we cannot provide the complete answers we received or further details about our respondents.

Overall, the responses of our interviewees were positive for all three questions. The answers confirmed that the positive Sharpe Ratio would justify the deployment of such an ATS for real-life trading, and that our chosen portfolio is reasonable. It was also noted that our threat model constitutes a tangible risk; some experts even acknowledged the likelihood of man-in-the-middle attacks between brokers and ATS and hence the risk of EP. Finally, some experts even remarked that organizations may use ATS in a “fire and forget” fashion, i.e., once the ATS is shown to yield some profit in the short-term, it will be deployed for several months until its performance is re-assessed. This implies that a hard-to-identify EP would adversely affect the trading decisions for a long time.

7.2 Limitations and Disclaimers

Our *ATS-SF* framework (§3) seeks to establish the means for security assessments of DL-based ATS. To enable a fine-grained control over its functionality and for better understanding of the operational characteristics of ATS, we have chosen to implement such a framework from scratch despite the existence of toolkits/platforms for ATS simulation, e.g., Backtrader [67] or Quantiacs [3]. We acknowledge that off-the-shelf tools may attain better real-world fidelity or offer more advanced capabilities for portfolio management. A comprehensive comparison of our framework with such tools is beyond the scope of this work.

Our security assessment is meant to provide a proof-of-concept evaluation of the potential impact of our threat model (§5). We cannot claim that any *operational* ATS is affected by EP in the same way as shown in our experiments; nor we claim that the attack is “universally devastating”. However, research on the security of AI in finance is scarce (as we showed in §2), and our findings reveal that even when the impact on the model is negligible, the overarching system may still be significantly affected. Such a factual consideration was not apparent from prior work (e.g., [61]).

Finally, our focus is in DL-based ATS. However, there are other ways to develop predictive models for stock price prediction (e.g., ARIMA [65, 71]): security assessment of these methods is outside our scope—but our resources enable one to evaluate these techniques, too (we provide examples in our repository [1]).

7.3 Additional Analyses on our Threat Model

We analyse some aspects of our threat model that can be used to extend our findings, but also as intriguing avenues for future work.

Robustness considerations. Our EP entails applying an imperceptible change to a single data point. Albeit we assume that such a change is introduced with a malicious purpose, it is entirely possible that such perturbations are caused by “neutral” events. For

instance, a broker’s may have received wrong data which is transmitted to the ATS. Hence, evaluating the effects of EP can be useful not only as a security assessment, but also as a means to investigate the overall robustness of ATS against data perturbations. We leave development of countermeasures against EP to future work (facilitated by our tools). Given our results, we believe a pragmatic defense to EP to be an intriguing avenue for this research domain.

More knowledge of the Portfolio Our threat model assumes the attacker knows only one stock in the portfolio ($s' \in \mathcal{P}$), but this can be extended. The attacker could instead know a subset $S' \subseteq \mathcal{P}$, allowing them to choose multiple stocks $s \in S'$ for crafting EPs. This increased knowledge could lead to stronger effects. However, obtaining such information is costly (i.e., the attacker cannot access the ATS from within). The attacker might infer portfolio details through privacy violations or insider actions [81], which are avenues for future work.

Alternative Goals and Strategies Our threat model envisions a constrained attacker aiming to harm an organization by reducing the yield of its ATS. Other methods exist to achieve this goal. For instance, an attacker with some control over communications between the broker and the ATS might launch a Denial of Service attack, preventing trades on a specific day. However, such a strategy is conspicuous, and the organization would likely react—which may lead to the attacker being detected. Conversely, it is also possible that the attacker uses the EP for “poisoning” a DL model: if an EP is not sanitized, it can be stored by the ATS. Such an EP can lead to changes in the predictions of multiple days (i.e., the EP can potentially affect all the following w days), but it can also have more collateral effects if the EP is used in the re-training process of the models. Our *ATS-SF* enables future work to investigate these ancillary adversarial scenarios.

8 Conclusions

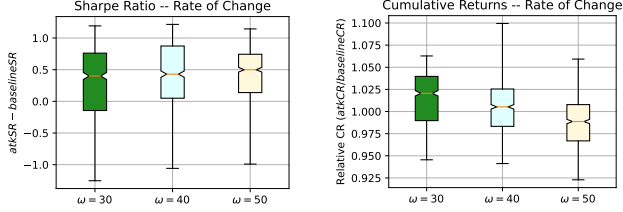
We elucidated the vulnerability of DL systems in finance to a novel, subtle type of adversarial perturbation, which we defined as “ephemeral perturbation” (EP). Compared to the limited prior works that studied the security of DL applications in computational finance, our research showed that our EP (which stem from a more constrained attacker than those envisioned in prior threat models) have an almost negligible impact on the performance of the affected DL model, but can reduce the profitability of a DL-powered algorithmic trading system (ATS). The latter finding is of crucial importance: security assessments studying the impact of data perturbations on the whole system are scarce in the financial context.

Future work should pay more attention to this application domain of DL—e.g., by devising countermeasures against our proposed EP; or assessing the effects of EP in high-frequency contexts, whose transactions occur every second (instead of daily); or even by relaxing some of the underlying assumptions of our proposed threat model, and studying their effects on the ATS end-to-end. Our proposed security assessment framework (*ATS-SF*) and our resources enable downstream research to carry out all such analyses.

Open source. Our experiments are reproducible, and our resources are available in a publicly available repository [1], in which we also report additional experimental results and visualizations of our assessment, as well as “how-to” guides.

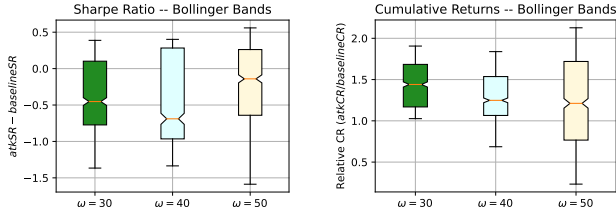
Acknowledgement

The authors want to thank the organizers and attendees of WACCO'24 for the constructive feedback, and the Hilti Foundation for funding this research (which has also been funded by the SAMLAF project).



(a) **Sharpe Ratio.** For 19.5% of the cases, the EP lead to a lower SR. (b) **Cumulative Returns.** Over 49% of our EP lead to a drop in the baseline CR.

Fig. 7: Overall impact of EP against the ATS using the Rate of Change trading strategy. For each attacked day (of our 666 testing window), we compute: the difference between the Sharpe Ratio (SR) achieved by the ATS at the end of the simulation (i.e., at day 666) with the baseline SR; and the ratio between the cumulative returns (CR) achieved by the baseline ATS respect to when it is attacked by an EP. We then plot the distribution of these “impacts”.



(a) **Sharpe Ratio.** For 62% of the cases, the EP lead to a lower SR. (b) **Cumulative Returns.** For 18.8% of the cases, the EP lead to a lower CR.

Fig. 8: Overall impact of EP against the ATS using the Bollinger Bands trading strategy. For each attacked day (of our 666 testing window), we compute: the difference between the Sharpe Ratio (SR) achieved by the ATS at the end of the simulation (i.e., at day 666) with the baseline SR; and the ratio between the cumulative returns (CR) achieved by the baseline ATS respect to when it is attacked by an EP. For Fig. 8b, we normalized the values between 0 and 1 before making the plot (since the baseline CR was a negative number)

References

- [1] 2024. Code Repository of our paper. <https://github.com/AdvijeR/ep-ats/>.
- [2] 2024. Interactive Brokers—Commissions. <https://www.interactivebrokers.com/en/pricing/commissions-stocks.php>.
- [3] 2024. Quantiacs. https://quantiacs.com/documentation/en/theory/theoretical_basis.html.
- [4] 2024. Starfetch – Leading the way in AI powered trading. <https://starfetch.ai/>.
- [5] 2024. Treasury Rate. https://ycharts.com/indicators/2_year_treasury_rate.
- [6] 2024. YahooFinance: Financial Data. <https://finance.yahoo.com/most-active>.
- [7] Mohiuddin Ahmed, Nazim Choudhury, and Shahadat Uddin. 2017. Anomaly detection on big data in financial markets. In *IEEE/ACM ASONAM*.
- [8] Monira Essa Aloud and Nora Alkamees. 2021. Intelligent algorithmic trading strategy using reinforcement learning and directional change. *IEEE Access* 9 (2021), 114659–114671.
- [9] Lars Ankile, Matheus XV Ferreira, and David Parkes. 2023. I See You! Robust Measurement of Adversarial Behavior. In *Multi-Agent Security Workshop@ NeurIPS'23*.
- [10] Giovanni Apruzzese, Hyrum S Anderson, Savino Dambra, David Freeman, Fabio Pierazzi, and Kevin Roundy. 2023. “Real Attackers Don’t Compute Gradients”: Bridging the Gap Between Adversarial ML Research and Practice. In *SaTML*.
- [11] Giovanni Apruzzese, Rodion Vladimirov, Aliya Tastemirova, and Pavel Laskov. 2022. Wild Networks: Exposure of 5G Network Infrastructures to Adversarial Examples. *IEEE Transactions on Network and Service Management* (2022).
- [12] Daniel Arp, Erwin Quiring, Feargus Pendlebury, Alexander Warnecke, Fabio Pierazzi, Christian Wressnegger, Lorenzo Cavallaro, and Konrad Rieck. 2022. Dos and don’ts of machine learning in computer security. In *USENIX Security*.

Algorithm 1 Applying an EP to time-series data of a given stock

Require: Time-series data $\mathcal{TS}$, EP magnitude m , time-series window size w , window-size expected by the attacker ω

Ensure: Set of adversarially perturbed time-series data $\overline{\mathcal{TS}}$

```

1: // Procedure for generating an adversarial time-series
2: function APPLYEP( $\mathcal{TS}$ ,  $t$ ,  $m$ ,  $\omega$ )
3:    $\mathcal{TS}' \leftarrow \mathcal{TS}$  ▷ Copy original time-series
4:   if  $m = \text{std}$  then ▷ Our Indiscriminate Attack (§5.2)
5:      $\sigma \leftarrow \text{std}(\mathcal{TS}[t - \omega : t])$  ▷ std.dev.-based EP
6:      $\mathcal{TS}'[t] \leftarrow \mathcal{TS}[t] + 2 \cdot \sigma$ 
7:   else ▷ Our Targeted Attack (§5.3)
8:      $\mathcal{TS}'[t] \leftarrow \text{custom}$  ▷ Ad-hoc crafting of EP
9:   end if
10:  return  $\mathcal{TS}'$ 
11: end function
12:  $\overline{\mathcal{TS}} \leftarrow \emptyset$ 
13: for ( $t = w$ ,  $t \in \mathcal{TS}$ ,  $t = t + 1$ ) do
14:    $\overline{\mathcal{TS}} \leftarrow \overline{\mathcal{TS}} \cup \text{APPLYEP}(\mathcal{TS}, t, m, \omega)$ 
15: end for

```

(Note: the magnitude m can be freely defined by an user)

- [13] Üzeyir Aycel and Yunus Santur. 2022. A new moving average approach to predict the direction of stock movements in algorithmic trading. *J. New Results in Science* (2022).
- [14] Arash Bahrammirzaee. 2010. A comparative survey of artificial intelligence applications in finance: artificial neural networks, expert system and hybrid intelligent systems. *Neural Computing and Applications* (2010).
- [15] Matthew Baron, Wei Xiong, and Zhijiang Ye. 2023. Measuring time-varying disaster risk: An empirical analysis of dark matter in asset prices. *SSRN:4189008* (2023).
- [16] Massimo Bartoletti, James Hsin-yu Chiang, and Alberto Lluch Lafuente. 2022. Maximizing extractable value from automated market makers. In *Int. Conf. Financial Cryptography and Data Security*.
- [17] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrđić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. 2013. Evasion attacks against machine learning at test time. In *ECML PKDD*.
- [18] Battista Biggio, Blaine Nelson, and Pavel Laskov. 2012. Poisoning attacks against support vector machines. In *ICML*.
- [19] Battista Biggio and Fabio Roli. 2018. Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognition* (2018).
- [20] Christian Borch. 2022. Machine learning and social theory: Collective machine behaviour in algorithmic trading. *European Journal of Social Theory* (2022).
- [21] Palod Chanrungrameekul, Juthathip Phuangbut, and Rujira Chaysiri. 2024. Bridging Bollinger Bands and Machine Learning in SET100 Stock Trading Strategies. In *2024 16th International Conference on Knowledge and Smart Technology (KST)*. 51–56. <https://doi.org/10.1109/KST61284.2024.10499648>
- [22] JAMES CHEN. 2020. Cumulative Return. <https://www.investopedia.com/terms/c/cumulativereturn.asp>.
- [23] Yu-Ying Chen, Chiao-Ting Chen, Chuan-Yun Sang, Yao-Chun Yang, and Szu-Hao Huang. 2021. Adversarial attacks against reinforcement learning-based portfolio management strategy. *IEEE Access* (2021).
- [24] Stuart Colianni, Stephanie Rosales, and Michael Signorotti. 2015. Algorithmic trading of cryptocurrency based on Twitter sentiment analysis. *CS229 Project* (2015).
- [25] Raphaël Dang-Nhu, Gagandeep Singh, Pavol Bielik, and Martin Vechev. 2020. Adversarial attacks on probabilistic autoregressive forecasting models. In *ICML*.
- [26] Ambra Demontis, Marco Melis, Maura Pintor, Matthew Jagielski, Battista Biggio, Alina Oprea, Cristina Nita-Rotar, and Fabio Roli. 2019. Why do adversarial attacks transfer? explaining transferability of evasion and poisoning attacks. In *USENIX Security*.
- [27] Guangyu Ding and Liangxi Qin. 2020. Study on the prediction of stock price based on the associated network model of LSTM. *Int. J. Machine Learning and Cybernetics* (2020).
- [28] Matthew F Dixon, Igor Halperin, and Paul Bilokon. 2020. *Machine learning in finance*. Springer.
- [29] Fernando GDC Ferreira, Amir H Gandomi, and Rodrigo TN Cardoso. 2021. Artificial intelligence applied to stock market trading: a review. *IEEE Access* (2021).

- [30] FINMA. 2021. AI in the Swiss financial market. <https://web.archive.org/web/20230922150131/https://www.finma.ch/en/documentation/dossier/dossier-fintech/kuentstliche-intelligenz-im-schweizer-finanzmarkt>.
- [31] Thomas Fischer and Christopher Krauss. 2018. Deep learning with long short-term memory networks for financial market predictions. *EJOR* (2018).
- [32] Michael Gallagher, Nikolaos Pitropakis, Christos Chrysoulas, Pavlos Papadopoulos, Alexios Mylonas, and Sokratis Katsikas. 2022. Investigating machine learning attacks on financial time series models. *Computers & Security* 123 (2022), 102933.
- [33] Micah Goldblum, Avi Schwarzschild, Ankit Patel, and Tom Goldstein. 2021. Adversarial attacks on machine learning systems for high-frequency trading. In *ICAIF*.
- [34] Peter Gomber and Kai Zimmermann. 2018. Algorithmic trading in practice. *The Oxford Handbook of Computational Economics and Finance* (2018).
- [35] Raúl Gómez Martínez, Miguel Prado Román, and Paola Plaza Casado. 2019. Big data algorithmic trading systems based on investors' mood. *JBF* (2019).
- [36] John W Goodell, Satish Kumar, Weng Marc Lim, and Debidutta Pattnaik. 2021. Artificial intelligence and machine learning in finance: Identifying foundations, themes, and research clusters from bibliometric analysis. *Journal of Behavioral and Experimental Finance* (2021).
- [37] Björn Rafn Gunnarsson, Seppe Vanden Broucke, Bart Baesens, María Óskarsdóttir, and Wilfried Lemahieu. 2021. Deep learning for credit scoring: Do or don't? *EJOR* (2021).
- [38] Yue Guo, Harish Karthikeyan, Antigoni Polychroniadou, and Chaddy Huussin. 2024. PriDe CT: Towards Public Consensus, Private Transactions, and Forward Secrecy in Decentralized Payments. In *IEEE S&P*.
- [39] Kristian Bondo Hansen. 2020. The virtue of simplicity: On machine learning models in algorithmic trading. *Big Data & Society* 7, 1 (2020), 2053951720926558.
- [40] James B Heaton, Nick G Polson, and Jan Hendrik Witte. 2017. Deep learning for finance: deep portfolios. *Applied Stochastic Models in Business and Industry* (2017).
- [41] Yves Hilpisch. 2020. *Artificial Intelligence in Finance*. O'Reilly Media.
- [42] Yuh-Jong Hu and Shang-Jen Lin. 2019. Deep reinforcement learning for optimizing finance portfolio management. In *Amity Int. Conf. Artificial Intelligence*.
- [43] Jian Huang, Junyi Chai, and Stella Cho. 2020. Deep learning in finance and banking: A literature review and classification. *Frontiers of Business Research in China* (2020).
- [44] Uyeong Jang, Xi Wu, and Somesh Jha. 2017. Objective metrics and gradient descent algorithms for adversarial examples in machine learning. In *ACSAC*.
- [45] Stefan Jansen. 2018. *Hands-On Machine Learning for Algorithmic Trading: Design and implement investment strategies based on smart algorithms that learn from data using Python*. Packt Publishing Ltd.
- [46] Constandina Koki, Stefanos Leonardos, and Georgios Piliouras. 2022. Exploring the predictability of cryptocurrencies via Bayesian hidden Markov models. *Research in International Business and Finance* (2022).
- [47] Adriano Koshiyama, Nick Firoozye, and Philip Treleaven. 2020. Algorithms in future capital markets: a survey on AI, ML and associated algorithms in capital markets. In *ICAIF*.
- [48] Chaitanya Deepthi Kothapalli, Guggilam Navya, Unnikiran Jaladhi, Shaik Ruhi Sulthana, Dasari Lokesh Sai Kumar, and Surapaneni Phani Praveen. 2023. Predicting Buy and Sell Signals for Stocks using Bollinger Bands and MACD with the Help of Machine Learning. In *International Conference on Sustainable Computing and Smart Systems (ICSCSS)*.
- [49] P Kumar. 2022. A comparative study of Bollinger Bands and rate of change with reference to NIFTY. (2022).
- [50] Raghavendra Kumar, Pardeep Kumar, and Yugal Kumar. 2021. Analysis of financial time series forecasting using deep learning model. In *Int. Conf Cloud Comput., Data Sci. & Eng.*
- [51] Arthur le Calvez and Dave Cliff. 2018. Deep learning can replicate adaptive traders in a limit-order-book financial market. In *IEEE SSCI*.
- [52] Shigang Liu, Jun Zhang, Yu Wang, Wanlei Zhou, Yang Xiang, and Olivier De Vel. 2018. A data-driven attack against support vectors of SVM. In *ACM AsiaCCS*.
- [53] Dongdong Lv, Shuhan Yuan, Meizi Li, Yang Xiang, et al. 2019. An empirical study of machine learning algorithms for stock daily trading strategy. *Mathematical problems in eng.* (2019).
- [54] Sidra Mehtab, Jaydip Sen, and Abhishek Dutta. 2021. Stock price prediction using machine learning and LSTM-based deep learning models. In *Symp. Machine Learning and Metaheuristics Algorithms, and Applications*.
- [55] Jason Milonios, Ciamac C Moallemi, and Tim Roughgarden. 2024. Automated market making and arbitrage profits in the presence of fees. In *Int. Conf. Financial Cryptography and Data Security*.
- [56] Ankit Mishra and Chaitanya Ghorpade. 2018. Credit card fraud detection on the skewed data using various classification and ensemble techniques. In *IEEE SCEECS*.
- [57] Gautam Raj Mode and Khaza Anuarul Hoque. 2020. Adversarial examples in deep learning for multivariate time series regression. In *AIPR Workshop*.
- [58] Jacinta Chan Phooi M'ng and Andrew Hong Jia Jean. 2014. The Usefulness of a New Technical Indicator, Rate of Change-Alpha (ROC- α) on Stock Markets: A Study of Malaysian Top Capitalization Stocks. (2014).
- [59] Milad Nasr, Reza Shokri, and Amir Houmansadr. 2019. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *IEEE S&P*.
- [60] Milad Nasr, Shuang Songi, Abhradeep Thakurta, Nicolas Papernot, and Nicholas Carlin. 2021. Adversary instantiation: Lower bounds for differentially private machine learning. In *IEEE S&P*.
- [61] Elinor Nehemya, Yael Mathov, Asaf Shabtai, and Yuval Elovici. 2021. Taking Over the Stock Market: Adversarial Perturbations Against Algorithmic Traders. In *ECML PKDD*.
- [62] Giuseppe Nuti, Mahnoosh Mirghaemi, Philip Treleaven, and Chaiyakorn Yingsaeer. 2011. Algorithmic trading. *Computer* (2011).
- [63] Ahmet Murat Ozbayoglu, Mehmet Ugur Gudelek, and Omer Berat Sezer. 2020. Deep learning for financial applications: A survey. *Applied Soft Computing* (2020).
- [64] Georgios Palaiokekrassas, Sandro Scherrers, Jason Ofeidis, and Leandros Tassioulas. 2024. Leveraging machine learning for multichain defi fraud detection. In *IEEE Int. Conf. Blockchain and Cryptocurrency*.
- [65] Muskaan Pirani, Paurav Thakkar, Pranay Jivani, Mohammed Husain Bohara, and Dweepna Garg. 2022. A comparative analysis of ARIMA, GRU, LSTM and BiLSTM on financial time series forecasting. In *IEEE ICDCECE*.
- [66] Antonio Riva, Lorenzo Bisi, Pierre Liotet, Luca Sabbioni, Edoardo Vittori, Marco Pinciroli, Michele Trapletti, and Marcello Restelli. 2021. Learning FX trading strategies with FQI and persistent actions. In *ACM ICAIF*.
- [67] Daniel Rodriguez. 2006. Backtrader. <https://github.com/mementum/backtrader>.
- [68] Mesboul Haque Sazu. 2022. How machine learning can drive high frequency algorithmic trading for technology stocks. *Int. J. Data Science and Advanced Analytics* (2022).
- [69] Omer Berat Sezer, Mehmet Ugur Gudelek, and Ahmet Murat Ozbayoglu. 2020. Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. *Applied soft computing* (2020).
- [70] William F Sharpe. 1998. The sharpe ratio. *Streetwise—the Best of the Journal of Portfolio Management* 3 (1998), 169–85.
- [71] Sima Siami-Namini, Neda Tavakoli, and Akbar Siami Namin. 2018. A comparison of ARIMA and LSTM in forecasting time series. In *IEEE ICMLA*.
- [72] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. In *ICLR*.
- [73] Wenpin Tang, Xiao Xu, and Xun Yu Zhou. 2022. Asset selection via correlation blockmodel clustering. *Expert Systems with Applications* (2022).
- [74] Thibaut Théate and Damien Ernst. 2021. An application of deep reinforcement learning to algorithmic trading. *Expert Systems with Applications* (2021).
- [75] Hien Thi Thu Truong, Miguel Almeida, Ghassan Karame, and Claudio Soriente. 2019. Towards secure and decentralized sharing of IoT data. In *IEEE Int. Conf. Blockchain*.
- [76] Mehar Viji, Deeksha Chandola, Vinay Anand Tikkiwal, and Arun Kumar. 2020. Stock closing price prediction using machine learning techniques. *Procedia computer science* (2020).
- [77] Weiguan Wang and Johannes Ruf. 2022. Information leakage in backtesting. *SSRN 3836631* (2022).
- [78] Xintong Wang and Michael P Wellman. 2020. Market manipulation: An adversarial learning framework for detection and evasion. In *IJCAI*.
- [79] Yongfeng Wang and Guofeng Yan. 2021. Survey on the application of deep learning in algorithmic trading. *Data Science in Finance and Economics* (2021).
- [80] Claes Wohlin. 2014. Guidelines for snowballing in systematic literature studies and a replication in software engineering. In *EASE*.
- [81] Daniel W Woods and Rainer Böhme. 2021. SoK: Quantifying cyber risk. In *IEEE S&P*.
- [82] Keyue Yan, Yimeng Wang, and Ying Li. 2023. Enhanced Bollinger Band Stock Quantitative Trading Strategy Based on Random Forest. *Artificial Intelligence Evolution* (2023).
- [83] Yiming Zhang, Yiyue Qian, Yujie Fan, Yanfang Ye, Xin Li, Qi Xiong, and Fudong Shao. 2020. dstyle-gan: Generative adversarial network based on writing and photography styles for drug identification in darknet markets. In *ACSAC*.
- [84] Yuhao Zheng, Xinyi Li, and Yuanjun Feng. 2022. Research on the Quantitative Trading Strategy Based on Bollinger Band Strategy and Polynomial Regression Model. In *2022 IEEE 2nd International Conference on Data Science and Computer Application (ICDSCA)*. IEEE, 1255–1260.
- [85] Liyi Zhou, Kaihua Qin, Christof Ferreira Torres, Duc V Le, and Arthur Gervais. 2021. High-frequency trading on decentralized on-chain exchanges. In *IEEE S&P*.
- [86] Pascal Zimmer, Sébastien Andreina, Giorgia Azzurra Marson, and Ghassan Karame. 2024. Closing the Gap: Achieving Better Accuracy-Robustness Tradeoffs Against Query-Based Attacks. In *AAAI Conf. Artificial Intelligence*.