Vol. 19 (2026): Proceedings of 2026 International Conference on Financial Technology, Mathematical Statistics, and Business Analytics (ICFMB 2026)
E-ISSN: 3105-0522 P-ISSN: 3105-0514
DOI: https://doi.org/10.70267/icfmb.202619619627

The Attacks and Defenses Mechanisms of Algorithmic Trading Systems driven by Deep Learning

Pengcong Wu*

Xiamen University Malaysia, Jalan Sunsuria, Bandar Sunsuria, 43900 Sepang, Selangor, Malaysia

**Corresponding author: Pengcong Wu.*

Abstract

Deep learning (DL) is now widely applied in quantitative finance, especially in algorithmic trading systems (ATS). However, its defense against adversarial attacks remains vulnerable due to inherent characteristics that pose substantial risks to financial markets. This paper provides a review of the security risks ATS faces, the attacks against it, and the defenses against them. It first presented the theoretical background by decomposing the ATS framework, briefly introducing relevant knowledge of adversarial attacks in DL, and highlighting the specific constraints of financial time series and regulatory oversight. Then, it integrated and analyzed various adversarial attack methods targeting ATS, proposed innovative shifts in the perspectives and criteria for evaluating the impact of adversarial attacks, compared existing defense mechanisms, and discussed their effectiveness and limitations. Finally, it provided theoretical support and practical guidance for building safer and more robust algorithmic trading systems in the future.

Keywords

cyber security, adversarial attacks, DL (deep learning), ATS (algorithmic trading system)

1. Introduction

Financial markets are consistently among the primary targets for adversarial attacks. Even subtle data manipulation or substitution can affect not only a single model but also, potentially, systematically impact the entire market. However, existing academic research on the practical application of adversarial machine learning in the financial sector remains relatively limited [1]. Meanwhile, there is publication bias, namely that the current literature likely overrepresents successful defenses, possibly creating a false perception of higher defense success rates than in the real world [2]. Therefore, it is significant to review the attacks and defenses in this field, as they are critical to maintaining the stability of financial systems.

This paper conducted a study on the following key questions: what types of adversarial attacks deep-learning-driven ATS faces, how these attacks affect trading performance and corporate profitability, and how the adaptability and effectiveness of up-to-date defense mechanisms are. To address these questions, the paper deconstructed the architecture of ATS, analyzed the theoretical foundation of adversarial attacks and the influence of the particularity of financial times series on the adversarial attacks, sorted out the present main

existing adversarial attack methods (such as Ephemeral Perturbations (EP)[1], Slope Attack [3], etc.) and corresponding defenses mechanisms, and summarized the limitations of current defense mechanisms. Finally, this paper offered directions for preventing adversarial attacks and ensuring security for next-generation ATS.

2. Theoretical Background

2.1 Decomposition of the Security Framework for Algorithmic Trading Systems

Rizvani et al. proposed dividing the ATS Security Framework into three environments: Asset Environment, Model Environment, and Trade Environment [1]. Asset Environment is responsible for determining the portfolio. And data on stocks used for portfolio determination is usually available on public resources or from private brokers, which means attackers can execute a Man-in-the-Middle Attack through the data input interface.

Model Environment is tasked with analyzing the input data and making predictions. Predictions are often achieved through DL or various linear regression algorithms like ARIMA, whose characteristics make the Model Environment vulnerable to adversarial attacks, making it a primary target for attackers.

Trade Environment is the last chronologically. ATS determines the “buy/sell/hold” signal based on the Model Environment's prediction and the preset threshold, and executes the final trading decision on capital allocation [1]. Attackers influence decision-making by tampering with the “buy/sell/hold” signal. Nevertheless, such manipulations do not work out sometimes, when the impact of manipulation on the signal is not strong enough (e.g., the metrics that were originally close to a “buy” signal changes to the ones that are close to a “sell” signal, while the final decision remains “hold”) or under the constraints of available resources (e.g., insufficient capital to be allocated for asset purchases) [1].

2.2 Theoretical Foundations of Adversarial Attacks Driven by Deep Learning

2.2.1 A Simple Classification of Adversarial Attack Types by Different Cafeteria

Given Adversary's Goal: The working process of a deep learning model can be divided into two phases: the “training phase” and the “inference phase”. Attacks targeting the training phase are mainly Poisoning Attacks, in which attackers inject adversarial samples into the training dataset to induce the generation of models with flaws for malicious purposes. As for the inference phase, Evasion Attacks are the main type of attack, which mislead the model into making incorrect predictions by feeding perturbed input samples [5].

Given Adversarial Specificity: It is grouped into Untargeted Attacks and Targeted Attacks [5]. Untargeted Attacks only require the victim model to produce erroneous predictions, regardless of the type of error, while Targeted Attacks aim to force the model to generate predictions toward designated outcomes. The rate of the latter is usually lower than that of the former [5].

There are three categories of attacks: White-box, Grey-box, and Black-box. White-box attackers possess full and precise knowledge of the model’s details, whereas Black-box attackers have no access to the targeted model's internal information. The only way for black-box attackers to obtain information is through interaction with the model and analysis of its outputs to infer it [5]. Gray-box attackers have limited information about the model but lack complete knowledge of its internal details [6].

2.2.2 The Lifestyle of an Adversarial Attack

Zhou advanced a five-stage framework, similar to the APT model, to present the lifecycle of an adversarial attack [5]: Stage 1 is Vulnerability Analysis. This stage is similar to the first stage of APT(Reconnaissance). It analyzes and explores the weakness by learning more about DNNs and identifying the factors that significantly influence their robustness. Then the attacker can exploit the vulnerabilities to design attack methods. In this way, the attack success rate would improve [5].

Stage 2 is Crafting. The goal of this stage is similar to the Establish Foothold stage of APT. Its methods focus on achieving a “successful entry” into the targeted DNNs [5] by exploiting information from Stage 1 and generating adversarial samples [7].

Stage 3 is Post-crafting. Similar to Lateral Movement of APT, the targeted information in this stage can be considered an extension of "successful entry," which emphasizes transferability and model-specific features [5]. Transferability can increase the likelihood of attacks on black boxes, or unknown DNNs [8]. Model-specific features enable attackers to carry out successful attacks under more challenging conditions.

Stage 4 is Practical Application. This stage shares similar characteristics with "Impediment". The application focuses on addressing potential problems from both digital and real-world perspectives, aiming to have an actual impact on the targeted system[5]. Simultaneously, the robustness of the samples obtained previously also improves [10].

Stage 5 is Revisiting Imperceptibility. The final stage shares the same goal as the last step of APT. The attackers make the distortions as minimal as possible to avoid recognition, most likely. Meanwhile, they also ensure the attack success rate [5].

2.3 Constraints Imposed by the Specificities of Financial Time Series and Financial Market Regulatory Mechanisms

2.3.1 Market Microstructure Constraints

Market microstructure constraints, such as trading hours, liquidity, and tick sizes, render perturbations less feasible [2].

2.3.2 Sound Regulatory Oversight

The financial market is highly noisy [11, 12], which makes it easy to conceal attacks but forces the attackers to introduce larger perturbations to achieve impactful influence. However, financial systems are typically equipped with sophisticated anomaly-detection mechanisms. Hence, the attacks must remain statically stealthy and avoid disrupting the natural continuity and noise characteristics of time series. It is exactly the ingenuity that the attacks discussed in the following sections, such as EP [1].

3. Literature Review

3.1 Attacks on the Model Environment of ATS

3.1.1 Slope Attack

The goal of Slope Attack is to manipulate the slope that the system predicts to the desired one [3]. Luszczynski [3] introduced and tested two types of Slope Attack using the N-HiTS model, which belongs to the Model Environment.

The first type is General Slope Attack. In general, attackers would want to slightly tamper with the data input to make it harder to detect the attack. Therefore, they would only affect the endpoints, leaving most of the data points in the middle of the time series unaffected. Applying the algorithm related to gradient descent, like the one that was proposed by Luszczynski [3], GSA finds the slope utilizing the discrete formula:

$$\mathrm{m} = \frac{y_2 - y_1}{x_2 - x_1} \tag{1}$$

Where (x_1, y_1) and $(x_2, y_2)$are the endpoints of the prediction [3]. Then, to induce the model to output the desired slope, GSA uses the following loss function to calculate the loss. Based on the loss, once the prediction goes away from the slope the attackers like, the objection function corrects it. The loss function is,

$$\{\text{loss} = \begin{matrix} ce^{-tdm} & \text{if } t \in -1,1 \\ cm^2 & \text{if } t = 0 \end{matrix} \tag{2}$$

where t is the target direction, and c and e are hyperparameters [3].

The second one is Least-Squares Attack (LSSA). The inherent nature of GSA makes this attack not change the overall trends of data, like increasing or decreasing, which means it cannot induce the target audience to purchase the stocks when the value of stocks rises or otherwise. However, LSSA is designed to alter the overall trend, rather than restricting adversarial modifications to only endpoints.

LSSA utilizes least squares to calculate the slope and applies the same loss function mentioned earlier, as follows [3]:

$$m = \frac{\sum_{i=0}^{N}(x-\tilde{x})(y-\tilde{y})}{(x-\tilde{x})^2} \tag{3}$$

Allowing for that, it does not change the endpoints; LSSA can reduce the likelihood of detection by combining with Adversarial Generative Adversarial Networks (A-GAN). In the architecture of the A-GAN designed by Luszczynski [3], a second critic was attached to the GAN architecture: N-HiTs, the target model. The generator learned the latent features of the original dataset and generates input data for N-HiTs. Then, LSSA was performed with a factor to balance the original generator loss and adversarial loss. Lastly, A-GAN reduced the LSSA loss by optimizing its own generation process and generated more realistic data without sacrificing maximization error [3].

3.1.2 Attacks on Order Book

Gradient-Based Attacks on Order Book also uses gradient iteration to conduct attacks. The order book records sell orders and buy orders. When the best price to buy, α (the highest price that someone will buy), is greater than or equal to the best price to sell β (the lowest price that someone will sell), namely $\alpha \geq \beta$, orders are operated and deleted from the book [13]. Because the input to the order book is a weighted average of prices and quantities at various price points, such as the size-weighted average (SWA) [13], changing the quantities of orders at any price would change the value of the input. Gradient-Based Attacks on Order Book exploit this peculiarity, which changes the number of orders at different prices rather than the prices themselves to affect the model's prediction [13].

Regarding this, Goldblum et al. [13] proposed that attackers can use a differentiable order book propagation model to simulate the placement, execution, and removal of fake orders submitted by attackers in the market. Then, the cross-entropy loss is calculated to measure the difference between the real label and the model's prediction. Subsequently, gradient ascent is adopted to iteratively optimize the adversarial sequence, making all the model's rise-and-fall signals opposite to the actual situation and thereby maximizing the attack effect [13].

Unlike GSA and LSSA, which are both targeted attacks, Gradient-Based Attacks on Order Book is an untargeted attack. Universal Transfer Attack is a targeted attack that does not necessarily require knowledge of future trends and leverages transfer learning, so attackers do not need to hold model details. It selects a set of snapshots of the order book in training data $\{x_i\}$ which is corresponding to target labels $\{y_j\}$ to feed into the surrogate model. It defines a function that includes a classification loss and a relative magnitude penalty. Then, through gradient descent to perform optimization and regeneration, it obtained a stationary order sequence $\{a_i\}$. Subsequently, once carrying out the attack, the attackers simply inject the sequence $\{a_i\}$ into the order book so that the attack will work out, regardless of the architecture of the target model [13].

3.2 System-level Attacks on ATS

3.2.1 Ephemeral Perturbations

Attackers manipulate communication between the broker and the ATS via a Man-in-the-Middle (MitM) attack to inject malicious or erroneous data [1]. Let s as each stock in a portfolio, s_t as the value of s at time t, s_t' is the adversarial value of s_t after the application of the Ephemeral Perturbations (EP), and m as a parameter which regulates the magnitude of perturbation. Then, an EP can be expressed in a function like this [1]:

$$EP(s,t,m) = s_t' \neq s_t \tag{4}$$

The attackers would only need to consider two choices before attacking: "when" and "how large," namely, what day or time to start equation t and the magnitude represented by m.

Judged only on these, EP makes changes to the input data of the forecasting model, which seems as if it just attacked the model environment. However, its effect is primarily at the system level.

The goal of EP differs from that of other attacks on the ATS Model Environment, which aim solely to manipulate the prediction and decision for a single trade. EP aims to degrade the system's long-term business

indicators, such as the Sharpe ratio, rather than merely affecting a single prediction. It exploits vulnerabilities in the data lifecycle and business logic while maintaining stealth and imperceptibility. The adversarial data EP injects are stored in the ARS database and in the internal Local Cache/Buffer on the day of the attack. The next day, the system frequently updates the database with the actual data without synchronously refreshing the sliding window buffer in memory. Due to data coverage in the database, it is extremely difficult for forensic investigations to trace the original attack samples. This dynamic vanishing, along with the minor discrepancy between its input data and the original values, ensures stealthiness again. The data in the sliding window of length κ will be used for indicator calculation, model prediction, and potential model retraining in the subsequent κ days. They will persistently interfere with system decision-making, leading to long-term degradation of performance indicators such as the system's Sharpe ratio and resulting in long-term damage at the system level [1].

3.2.2 Trojan Malware for Model Interface

Luszczynsk pointed out in [3] that systems like ATS that deploy and utilize models are vulnerable to attacks, including, but not limited to, adversarial attacks, and an advanced attack named Trojan Malware for Model Interface.

Attackers launch Trojan attacks via the system’s interfaces. They can modify the prediction catalog's code to inject adversarial attacks and bypass the model's filtering/detection system, or delete any calls that turn off gradients, creating conditions for more powerful white-box attacks [3]. This attack scope has gone beyond the single dimension of ATS prediction models and instead leverages ATS infrastructure to carry out the attack. Therefore, it is classified as an attack targeting the ATS system level.

If the attack is performed by external attackers, namely someone who did not participate in the development of the model and the project package, there is a defense method: hashing the project to encrypt it and prevent tampering. In this way, when installing the program, the integrity of the directory can be verified by comparing the computed hash value with the original hash value [3].

If the attack is carried out by internal attackers, namely internal members of the development team, it can only be identified and eliminated through a rigorous code review mechanism [3].

3.3 Existing Defense Mechanisms

Existing Defense Mechanisms can be divided into two categories: adversarial training-based defenses and detection-based defenses.

3.3.1 Training-based Defense

This is the most frequently mentioned defensive strategy. When training a prediction model, both normal and adversarial samples are used to retrain the model, thereby improving its robustness [2]. Nevertheless, this methodology exhibits certain limitations. Adversarial training may degrade learning capacity on clean data and reduce prediction accuracy, resulting in direct economic losses [14]. Furthermore, adversarial training requires model retraining that consumes substantial computational resources, resulting in poor extensibility for deep neural networks [5], and is difficult or even nearly infeasible to implement in ATS defense scenarios (e.g., Luszczynski [3] noted that the N-HiTS model adopted in his experiment on slope attacks faced extremely high engineering complexity in generating corresponding adversarial training data, rendering the approach nearly unfeasible).

3.3.2 Detection-based Defense

A detector or discriminator is typically constructed using architectures such as k-nearest neighbors (kNN), artificial neural networks (ANNs) [14], and 4-layer convolutional neural networks (CNNs) [3]. It identifies whether an input sample is a normal sample or an adversarial example by recognizing certain characteristics of deep neural networks (DNN).

The drawback of this defense lies in the insufficient accuracy of detectors (such as the ANN-based detector in [14]), especially against emerging attacks (for example, the two slope-based attacks in [3] are detected by a 4-layered CNN with only a 50% to 60% probability, maintaining high stealthiness). Moreover, some detectors

have a short lifespan of sustained high performance, requiring defenders to update them every few weeks. Undoubtedly, this is impractical in real-world scenarios [14]. Additionally, detectors consume computational resources, and since ATS used in high-frequency trading demands substantial computing power, running detectors may reduce the system's operational efficiency and lower profits [14].

Apart from the two kinds above, the writer also found an input-purification defense mechanism. Unlike detection-based defenses, this defense purifies data not by filtering input samples, but by eliminating perturbations through preprocessing. Representative implementations include PixelDefend [15] and Defense-GAN[16]. It should be noted that both methods are only designed to defend deep learning models against adversarial attacks, and no existing literature has documented their application in experiments investigating defense mechanisms for deep learning-driven ATS. Given the high noise and other characteristics of financial transaction environments [2], this approach is theoretically difficult to implement.

4. Discussion & Synthesis

4.1 Comparison of Similarities and Differences Among the Attacks Mentioned

Except for the classification and comparison in the Literature Review based on the differing overall targets of the attacks, further in-depth, multi-dimensional comparisons can be conducted on these attacks (except for Trojan Malware for Model Interface; this method is essentially not a type of adversarial attack and differs substantially from all other attacks mentioned above).

Table 1: Adversarial specificity comparison

Type	Name	
Untargeted Attacks	Universal Transfer Attack on Order Book [13]	EP, which can launch untargeted indiscriminate attack, or targeted attack [1].
Targeted Attackss	GSA [3], LSSA [3], Gradient-Based Attacks on Order Book [13]	

Table 2: Adversary's knowledge comparison

Type	Name
Black-box Attacks	Universal Transfer Attack on Order Book [13]
White-box Attacks	GSA [3], LSSA [3], Gradient-Based Attacks on Order Book [13]
Grey-box Attacks	EP [1]

EP is classified as a Grey-box Attack because it requires partial knowledge of the model: for instance, the training data for the targeted model in [1] was sourced from Yahoo Finance, and the attack targeted the closing price of GOOGL [1].

Table 3: Based on applicable trading frequency

Applicable Trading Frequency	Name
High-Frequency Trading	Universal Transfer Attack on Order Book, Gradient-Based Attacks on Order Book [13]
Medium- and Low-Frequency Trading	GSA, LSSA [3]
Low-Frequency Trading	EP [1]

EP not only has a larger impact scope and broader applicability, but also requires harder constraints for launching attacks that are more aligned with real-world conditions. Nevertheless, there are aspects requiring further investigation, such as whether the attack target can be extended from low-frequency to high-frequency transactions.

4.2 Evolution of the Perspective on Adversarial Attack Effectiveness Evaluation

The effectiveness evaluation criteria for adversarial attacks are undergoing a critical shift from assessing the “model error” generated by prediction models after being attacked to evaluating the “system financial loss” incurred by the entire system after an attack.

In prior studies, distortion of model predictions has been universally adopted to measure attack effectiveness. For instance, in research on order-book trading systems, attackers maximize cross-entropy loss

to achieve the strongest attack capability [13]; in studies on slope-based attacks, attackers also evaluate attack performance by maximizing the mean absolute error (MAE) and root mean square error (RMSE), or by forcing changes to the slope of the prediction sequence [3]. However, in practice, the evaluation of attack effectiveness in all these studies is based on an ideal premise: that changes in the model environment can necessarily be translated into changes in the trade environment.

However, this idealization overlooks the fact that multiple practical constraints may force decision-makers to deviate from trading strategies strictly derived from prediction models, including risk exposure, insufficient purchasing capital, and insufficient tradable shares under the control of investors. The study in [1] found that although EP successfully induced misleading predictions from the LSTM model at the model level, the trading system had no actual adverse impact on the Sharpe Ratio and Cumulative Returns [1].

4.3 Trade-off Between Attack Stealthiness and Attack Intensity and Methodological Innovation

We find that the attacks mentioned in Section 3 may not strictly follow the adversarial attack lifecycle [5], but their overall process is generally consistent with the framework, specifically including the Revisiting Imperceptibility step. Due to the high inherent noise of financial systems, coupled with the stringent requirements of their security detection mechanisms [2], all attacks attach great importance to Revisiting Imperceptibility, and this consideration is ranked second in priority only after vulnerability analysis. It can be concluded that after identifying and analyzing vulnerabilities, the primary task is to maximize the imperceptibility of the attack as much as possible, and then maximize the attack intensity on this basis.

For example, the GSA reduces attack visibility by modifying endpoints alone. In contrast, the LSSA modifies a sequence of data, resulting in significantly higher attack visibility than GSA, which only modifies endpoints. Consequently, generative adversarial networks such as A-GAN have been adopted to generate adversarial samples, which improves the authenticity and imperceptibility of the modified data [3].

However, compared with these attacks that improve imperceptibility solely by optimizing input adversarial samples, the implementation of EP features certain innovation. EP not only reduces attack visibility by constraining the absolute difference between the first-order derivative of the input sample value s_t' and the real sample value s_t, but also adopts an "overwritten shortly afterward" mechanism. This approach is more imperceptible and successfully evades long-term auditing, unlike other existing attacks. Moreover, the attack effect of EP is more persistent and broader in its impact.

4.4 Limitations of Existing Defense Mechanisms and Major Future Development Directions

Existing adversarial training-based defenses and detection-based defenses consume considerable computational resources, with the former incurring particularly high costs. Moreover, adversarial training-based defenses often reduce model prediction accuracy in exchange for improved robustness [5, 14], while detection-based defenses still have substantial room for improvement in recognition accuracy. Overall, when facing low-cost, rapidly evolving adversarial attacks, existing defense mechanisms still incur high deployment and operational costs and remain difficult to deploy in practical applications. The ATS is currently confronted with the dilemma where attacks are stronger and easier to execute than defenses. Therefore, future research should be oriented toward improving the success rate of defense against attacks, reducing the construction and computational costs of defensive systems (for instance, applying knowledge distillation when conducting adversarial training in the manner proposed in [17] may serve as a feasible solution), and lowering end-to-end latency to adapt to high-frequency trading scenarios. Additionally, research should gradually shift its focus beyond improving model robustness and toward systematic research aligned with the systematic assessment of attacks.

5. Conclusion

This paper reviews the existing state-of-the-art typical novel attack methods targeting deep learning-driven algorithmic trading systems (ATS), evaluates the potential losses caused by such attacks, identifies the innovative characteristics of these novel attack methods, summarizes the current status of existing defense mechanisms, and points out their limitations, thereby providing a theoretical foundation for future research on

attack and defense mechanisms for deep learning-driven ATS. This paper also calls on future research to shift the focus from "model-level defense" to "system-wide defense" and to deploy defense strategies from a more systematic and macroscopic perspective.

References

[1] Rizvani, A., Apruzzese, G., & Laskov, P. (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)* (pp. 1-12). ACM. https://doi.org/10.1145/3714393.3726490

[2] Kurniawan, A., Putra, M. G., Hakim, D. L., & Ariyanto, M. (2025). Temporal adversarial attacks on time series and reinforcement learning systems: A systematic survey, taxonomy, and benchmarking roadmap. *Preprints*. https://www.preprints.org/manuscript/202601.0598

[3] Luszczynski, D. (2025). *Targeted manipulation: Slope-based attacks on financial time-series data*. arXiv. https://doi.org/10.48550/arXiv.2511.19330

[4] Pirani, M., Thakkar, P., Jivrani, P., Bohara, M. H., & Garg, D. (2022, April 23–24). *A comparative analysis of ARIMA, GRU, LSTM and BiLSTM on financial time series forecasting* [Paper presentation]. 2022 IEEE International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE), Ballari, India. https://doi.org/10.1109/ICDCECE53908.2022.9793213

[5] Zhou, S., Liu, C., Ye, D., Zhu, T., Zhou, W., & Yu, P. S. (2022). Adversarial attacks and defenses in deep learning: From a perspective of cybersecurity. *ACM Computing Surveys, 55*(8), 1–39. https://doi.org/10.1145/3547330

[6] Feng, H., Li, S., Shi, H., & Ye, Z. (2024). A comparative analysis of white box and gray box adversarial attacks to natural language processing systems. In *Proceedings of the 2024 2nd International Conference on Image, Algorithms and Artificial Intelligence (ICIAAI 2024)*. https://doi.org/10.2991/978-94-6463-540-9_65

[7] Goodfellow, I. J., Shlens, J., & Szegedy, C. (2015). Explaining and harnessing adversarial examples. In *Proceedings of the 3rd International Conference on Learning Representations*. https://doi.org/10.48550/arXiv.1412.6572

[8] Demontis, A., Melis, M., Pintor, M., Jagielski, M., Biggio, B., Oprea, A., Nita-Rotaru, C., & Roli, F. (2019). Why do adversarial attacks transfer? Explaining transferability of evasion and poisoning attacks. In *Proceedings of the 28th USENIX Security Symposium* (pp. 321–338). https://doi.org/10.48550/arXiv.1809.02861

[9] Chang, H., Rong, Y., Xu, T., Huang, W., Zhang, H., Cui, P., Zhu, W., & Huang, J. (2020). A restricted black-box adversarial framework towards attacking graph embedding models. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*. https://doi.org/10.48550/arXiv.1908.01297

[10] Yakura, H., & Sakuma, J. (2019). Robust audio adversarial example for a physical attack. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence* (pp. 5334–5341). https://doi.org/10.24963/ijcai.2019/74

[11] Gupta, S., Walia, N., Singh, S., & Gupta, S. (2023). A systematic literature review and bibliometric analysis of noise trading. *Qualitative Research in Financial Markets*, *15*(1), 190–215. https://doi.org/10.1108/QRFM-09-2021-0154

[12] Cui, W. (2026). The explicable market microstructure noise. *Journal of the American Statistical Association*. Advance online publication. https://doi.org/10.1080/01621459.2026.2622104

[13] Goldblum, M., Schwarzschild, A., Patel, A., & Goldstein, T. (2021). Adversarial attacks on machine learning systems for high-frequency trading. In Proceedings of the 2nd ACM International Conference on AI in Finance (ICAIF'21) (Article No. xxx, pp. 1–9). Association for Computing Machinery. https://doi.org/10.1145/3490354.3494367

[14] Nehemya, E., Mathov, Y., Shabtai, A., & Elovici, Y. (2021). Taking over the stock market: Adversarial perturbations against algorithmic traders. In Y. Dong, N. Kourtellis, B. Hammer, & J. A. Lozano (Eds.), *Machine Learning and Knowledge Discovery in Databases. Applied Data Science Track: ECML PKDD 2021, Proceedings, Part IV* (LNAI Vol. 12978, pp. 221-236). Springer. https://doi.org/10.1007/978-3-030-86514-6_14

[15] Song, Y., Kim, T., Nowozin, S., Ermon, S., & Kushman, N. (2018). PixelDefend: Leveraging generative models to understand and defend against adversarial examples. In Proceedings of the 6th International Conference on Learning Representations. https://doi.org/10.48550/arXiv.1710.10766

[16] Samangouei, P., Kabkab, M., & Chellappa, R. (2018). Defense-GAN: Protecting classifiers against adversarial attacks using generative models. In *Proceedings of the 6th International Conference on Learning Representations*. https://doi.org/10.48550/arXiv.1805.06605

[17] Li, D., Li, X., Li, Z., & Guo, Y. (2025). Ensemble adversarial training with knowledge distillation. In *2025 4th International Conference on Intelligent Mechanical and Human-Computer Interaction Technology (IHCIT)* (pp. 179–184). Beijing, China. https://doi.org/10.1109/IHCIT66787.2025.11199002

Funding

This research received no external funding.

Conflicts of Interest

The authors declare no conflict of interest.

Acknowledgment

This paper is an output of the science project.

Open Access