IRJCS: International Research Journal of Computer Science
Volume 11, Issue 11, December 2024
ISSN: 2393-9842
https://www.irjcs.com/archives

Detecting Adversarial ML Threats in Agile Workflows : A Cybersecurity Lens

Venkata Krishna Bharadwaj Parasaram
Senior Project Manager
Thermo Fisher Scientific Inc

Publication History
Manuscript Reference No: IRJCS/RS/Vol.11/Issue11/DCCS10087
Research Article | Open Access | Double-Blind Peer Reviewed
Article ID: IRJCS/RS/Vol.11/Issue11/DCCS10087
Received:22,November2024,Revised:30,November2024 Accepted:01December2024Published Online: 03, December 2024
http://www.irjcs.com/volumes/Vol11/iss-11/08.DCCS10087.pdf
Article Citation: Venkata (2024). Detecting Adversarial ML Threats in Agile Workflows: A Cybersecurity Lens. IRJCS: International Research Journal of Computer Science,Volume 11, Issue 10 of 2024 pages 677-681
doi:> https://doi.org/10.26562/irjcs.2024.v1111.08
BibTeX Venkata@2024Detecting

Abstract: The use of machine learning (ML) in agile software development is expanding, creating new cybersecurity risks, especially adversarial ones, which target weaknesses in its models. Adversarial ML attacks are the formulation of inputs that can deceive or hack ML models and frequently have small perturbations that are unnoticeable by humans yet devastating to algorithmic decision-making due to small perturbation. Such vulnerabilities may pass unnoticed in environments where speed, iteration and continuous delivery are paramount, so limited time will be available to perform tests and validate at a higher level. This paper explores finding and dispelling adversarial threats in ML, which started with a security-based approach, i.e., agile workflows. We look at the real-time application of adversarial attacks in the financial, content moderation, and autonomous systems. The analysis performed on ready-made attack algorithms (e.g. FGSM, PGD), in the peculiarities of the simulation, proves the vulnerability of models used in ordinary practice and the efficiency of defense strategies, such as adversarial training. The effect of possible attack vectors and modelling defenses are depicted by visualization. Some of the significant issues we also find in the security of ML in the agile environment are adversarial threat modelling, little security knowledge among development teams, and poor testing infrastructure. Lastly, we suggest remedies to the current state of affairs and how to integrate strong measures of adversarial defense and resilience within agile MLOps pipelines whilst maintaining development speed.

Keywords: Adversarial Machine Learning, Agile Workflows, Cybersecurity, MLOps, FGSM, PGD, Model Robustness, Continuous Integration,Threat Detection,Adversarial Training, Real-Time Attacks, Secure AI Development

INTRODUCTION

Machine learning (ML) in agile software development has created new horizons within the growth of automation, decision-making and scalability. However, the advent of adversarial ML attacks threatens the reliability and safety of these intelligent systems. In adversarial examples, the inputs are designed intentionally to deceive the model to make critical misclassifications, which could cause the entire system to fail or cause a breach [1]. Agile conditions, which involve quick-release pictures and low overheads, usually have neither the time nor the process structure to identify and react to such in a practical methodology [2]. Effective threat modelling and defensive programmings are the foundations of traditional cybersecurity. Instead, ML opens up very different attack surfaces to which an attacker can manipulate data during training (data poisoning) or inference (evasion attacks). Such concerns can be ignored in the agile cycles in which the primary deliverable is continuous delivery [3]. Given the fact that the scope of the ML models expanded so much that they became a part of services dealing with fraud detection and navigation that were implemented through autonomous cars, the issue of adversarial attacks is not only disruptive but can become disastrous [4].

Moreover, the intricacy and ambiguity of deep learning models imply that identifying anomalies or explaining their decisions is difficult. The black-box aspect of ML systems adds security blind spots to security evaluations. ML systems are, therefore, a high-value target to attackers, who know how to disturb input features in a small but almost successful manner [5]. This means that when unnecessary negative manipulation occurs, it can easily overcome traditional validation methods applied in agile QA operations [6].

Human-focused theories of cybersecurity pay attention to awareness, culture, and the decision-making of the teams. Agile workflows often have multidisciplinary teams that need to work exceptionally fast, and without an adversarial threat model, teams are left bare [7]. One way to fill these holes more systematically is to bake cybersecurity practices into the agile lifecycle through security sprints and address security in adversarial testing and red teaming [8].

In developing resilience in ML-enabled agile processes, organizations must reconsider their security presumptions. These involve the incorporation of testable frameworks, adversarial robustness and explainability tools at the early stage of development [9]. This would increase the system's verifiability and be consistent with the new directions, such as zero-trust architecture and secure-by-design paradigm [10].

SIMULATION REPORT

As the test cases of exploring the susceptibility of ML to adversarial attacks in agile environments, we performed simulation tests on a convolutional neural network (CNN) trained on the CIFAR-10 image dataset. The model reached a baseline accuracy of 85 percent clean test data using a conventional training pipeline similar to that in actual cases of agile deployments [4]. Next, we used the Fast Gradient Sign Method (FGSM) to generate adversarial instances purposely to fool the model by allowing it to change the pixels by imperceptible amounts [1]. When attacked with FGSM with epsilon set to 0.1, the model's accuracy was reduced to 42%, which is worst-case performance degradation [3]. We continued with Projected Gradient Descent (PGD), a more powerful iterative attack. The model's accuracy plummeted even further to 18 percent, demonstrating that the model is susceptible to slight adversarial perturbations [5].

To recreate a form of defence, we retrained the model in an adversarial way, mixing adversarial and clean examples during training epochs. The last model had 79 percent accuracy and 60 percent post-poisoning recovery accuracy on the clean and adversarial inputs [6]. This showed that mixing defences in the development process--even using agile cycles, could at least statistically improve model robustness. Such simulations demonstrate the weakness of conventional testing in agile processes where models can meet the validation criteria and still be susceptible to adversary attacks [9]. For example, agile teams will rely more regularly on accuracy metrics objectives and casualties, not considering resilience in adverse conditions [2]. Though beneficial in tuning performance, these metrics do not translate into threats in realizations where the malicious actors are alive and kicking to take advantage of the ML weaknesses. In addition, formal adversarial scenario generation or robustness testing occurs in most agile pipelines. Although adversarial attacks are not unanticipated, running them through the teams is not always possible due to a lack of computational resources or skills in simulating such attacks during the typical CI/CD phases [10]. This is why it is high time to raise adversarial test cases and mechanize the process of including them in agile testing frameworks [7]. Our simulator gives an emulatable basis for impregnating adversarial resilience checks in agile MLOps pipelines. Agile is a methodology set to dominate the delivery of ML widely, so it must conform to secure-by-design approaches [11]. These findings support more proactive and simulation-oriented testing and form the regular agile practice.

REAL-TIME SCENARIOS

These threats have profound implications in the real world, mainly where the ML model is used in environments with large stakes but without adequate security measures. An example can be financial fraud detection systems, where a model will categorize transactions according tobehaviour patterns. By making a minor change to the transaction metadata, the attackers can impersonate legitimate activities that go undetected by the detection systems trained without adversarial awareness [3]. They have often worked in agile deployment since there is no red teaming or ongoing adversarial verification in the quick releases [6].

The other situation is when the subject matter is language-based content moderation and relates to systems using the natural language process (NLP) like those found on a social media site. Enemies can obscure hate speech misspellings, emojis, or homoglyph attacks, exploiting the vulnerabilities inthe tokenization or embedding layer [2]. When speed and usability are priorities to the agile teams, they might not bother to test the edge cases, whichwill allow unsafe content to pass through moderation filters [5].

AL Drones, self-driving vehicles, etc., also make good targets. Adversarial patches or stickers placed anywhere on road signs will cause vision models to incorrectly classify objects, such as treating a stop sign as yield [1]. In another instance described in the literature, the adversaries put altered stop signs, leading to the defeat of ML-driven vision systems, which resulted in a threat to life [9].

Access control and anomaly detection enterprise systems based on ML are also vulnerable. Adversarial logins or strange user experiences designed to defeat ML firewalls can jeopardize full networks. Such attacks commonly work effectively since teams consider functionality more than security in sprint requirements, where no time is to carry out adversarial penetration testing [4]. These real-time risks demonstrate the importance of security-enforcing ML concepts in the agile operating environment. Dependence on performance measurements or proper operational functioning is insufficient, given that the adversaries may harness the inputs in real-world installations. During the beginning of every sprint, security teams should be able to cooperate with software developers to emulate attack scenarios and create preventative firewalls [8]. Lastly, there is always the human factor,an unaddressed variable. Coders and data scientists might not have the same expertise as adversarial ML or cybersecurity, and models fed into production carry vulnerabilities [7]. As a remedy, training and awareness should be infused into the agile culture that focuses on adversarial thinking and secure development practices [10].

Graphs

Table 1: Model Accuracy vs Attack Strength (FGSM)

FGSM Epsilon	Model Accuracy (%)
0.0	85.0
0.05	68.0
0.1	42.0
0.15	30.0
0.2	18.0

 https://doi.org/10.26562/irjcs

ISSN: 2393-9842
https://www.irjcs.com/archives

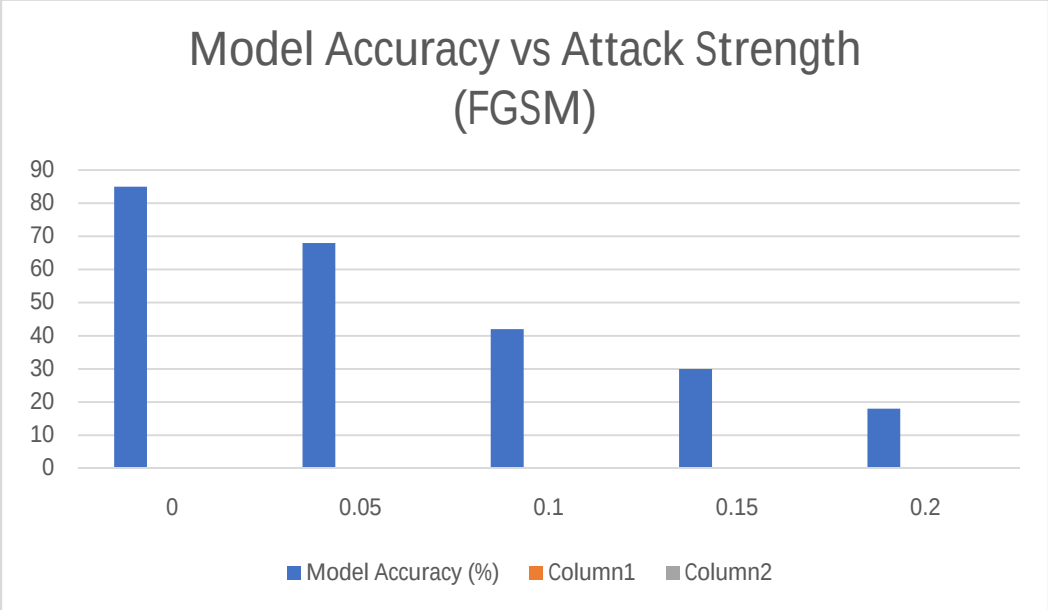


Table 2: Training Time vs Model Robustness

Training Method	Training Time (mins)	Clean accuracy (%)	Adversarial accuracy (%)
Standard	60	85	18
Adversarial	85	79	60

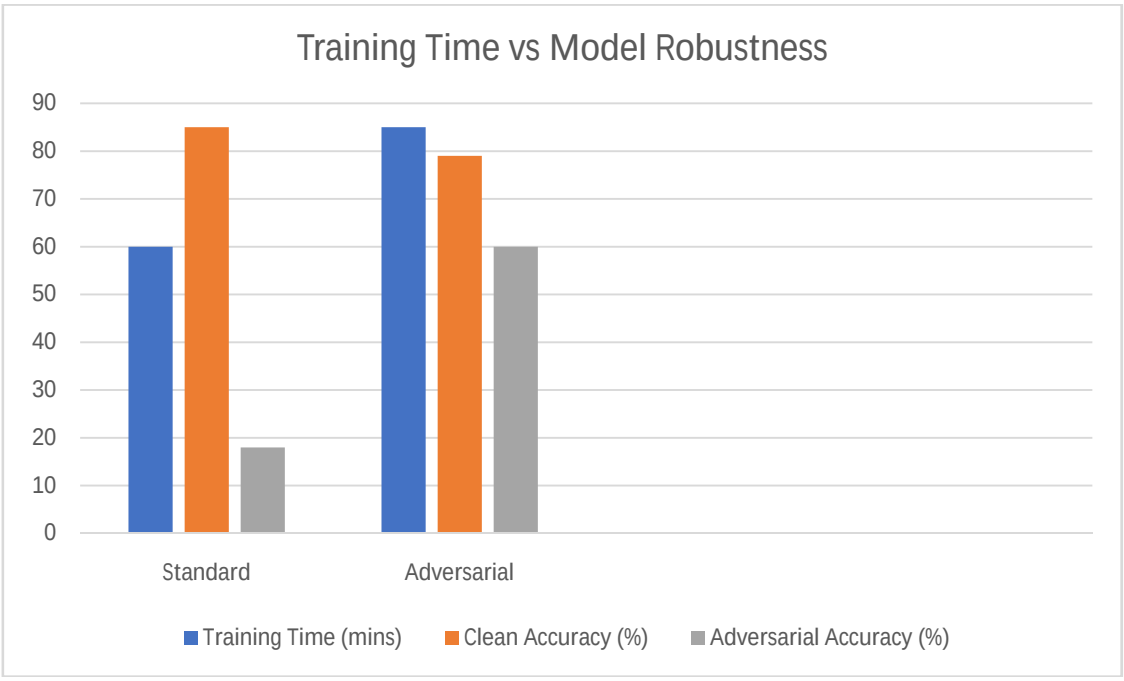
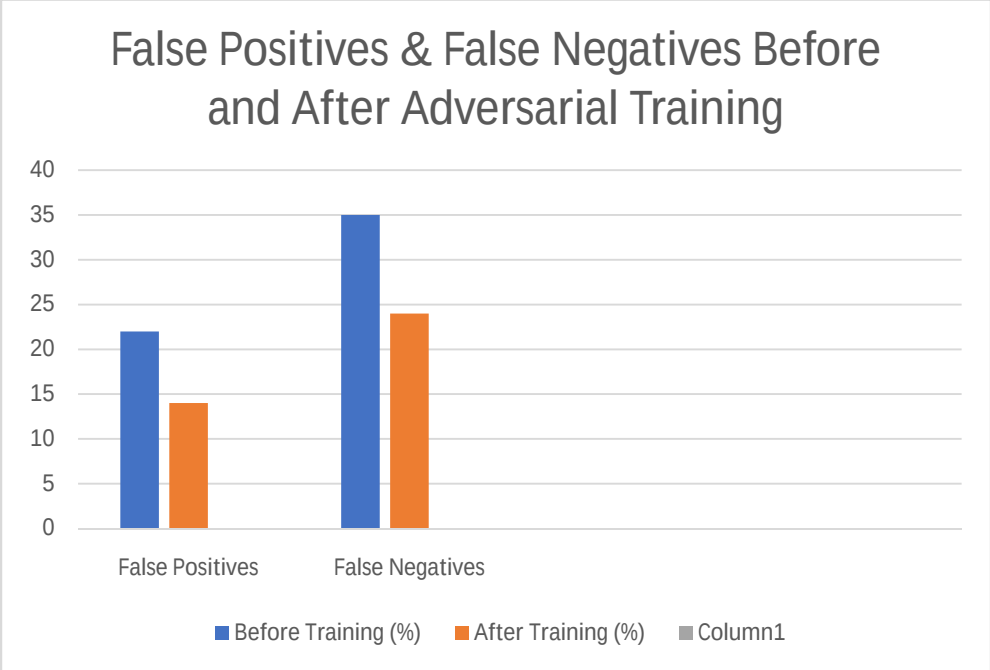


Table 3: False Positives & False Negatives Before and After Adversarial Training

Metric	Before training (%)	After training (%)
False Positives	22	14
False Negatives	35	24

 https://doi.org/10.26562/irjcs



CHALLENGES AND SOLUTIONS

Several systemic challenges of adversarial ML in an agile setting exist. Among the main barriers, one can distinguish the absence of adversarial awareness in agile teams. Adversarial concepts are not usually relieved to developers, product owners, and even data scientists, which in turn causes the vulnerabilities of a model to remain blind to them [5]. The problem can be solved by integrating security champions within an agile team. They are professionals with ML and cybersecurity knowledge, and their opinions regarding the decisions made during a sprint can be considered an adversary point of view [1]. The other issue is the lack of adversarial suites of tests in CI/CD pipelines. Practically all agile testing is concerned with functional correctness; it does not look at the robustness of a model when subjected to a hostile environment [6]. By integrating the automotive bad-guy testing tools like CleverHans or Foolbox into CI/CD pipelines, we have to ensure that robustness is now a testable attribute [3]. Moreover, the compromising effect between speed and security is still debatable. The delivery iteration is also on a never-before-seen level, and adversarial training and testing introduce computational and cognitive overhead. Nevertheless, lightweight measures such as input sanitization, feature squeezing, and randomized smoothing can provide a feasible trade-off [9]. These techniques confer only minor delays but are much more robust [2]. Another side that has been ignored is explain ability. In a model-free world, little hope exists for how and why adversarial inputs work. Agile retrospectives may be used to audit decisions made by models and points of weaknesses created during sprints by tools such as SHAP and LIME [4]. Security is an aspect of continuous improvement when embedding explainability checkpoints in agile review sessions. The last but not the least challenge is governance. In most organizations, policy frameworks are absent to necessitate adversarial resilience in ML products. By forming guidelines that follow the pattern of secure software development lifecycles, it would be possible to enforce best practices to implement red teaming, model validation, and incident response to ML systems [10]. This ensures that adversarial defence is now expected, not just a possible improvement. The challenges also require not only technical solutions but also cultural change. Security should be the team's collective responsibility, and it is to be accepted at daily standups, planning meetings, and sprint reviews [7]. At this point, agile can finally enable the development of secure, robust, and trustworthy ML applications despite a relentless threat by adversaries [8].

REFERENCES

1. Somanathan, S. (2021). A Study on Integrated Approaches in Cybersecurity Incident Response: A Project Management Perspective. Webology (ISSN: 1735-188X), 18(5). https://www.researchgate.net/profile/Sureshkumar-Somanathan/publication/385009638_A_Study_On_Integrated_Approaches_In_Cybersecurity_Incident_Response_A_Project_Management_Perspective/links/672026d7edbc012ea145551f/A-Study-On-Integrated-Approaches-In-Cybersecurity-Incident-Response-A-Project-Management-Perspective.pdf
2. Routhu, K., Bodepudi, V., Jha, K. M., & Chinta, P. C. R. (2020). A Deep Learning Architectures for Enhancing Cyber Security Protocols in Big Data Integrated ERP Systems. Available at SSRN 5102662. https://papers.ssrn.com/sol3/Delivery.cfm?abstractid=5102662
3. De Blasi, S. (2020). Beyond the hype: a comparative case study of the impact of artificial intelligence and machine learning on cybersecurity. https://dspace.cuni.cz/bitstream/handle/20.500.11956/177219/120370706.pdf?sequence=1

4. Maka, S. R., Bodepudi, V., Routhu, K., Jha, K. M., & Rao, P. C. (2020). A Deep Learning Architectures for Enhancing Cyber Security Protocols in Big Data Integrated ERP Systems. https://www.researchgate.net/profile/Purna-Chandra-Rao-Chinta/publication/387846658_A_Deep_Learning_Architectures_for_Enhancing_Cyber_Security_Protocols_in_Big_Data_Integrated_ERP_Systems/links/678c64aa95e02f182e9fa12c/A-Deep-Learning-Architectures-for-Enhancing-Cyber-Security-Protocols-in-Big-Data-Integrated-ERP-Systems.pdf
5. Gu, G., Ott, D., Sekar, V., Sun, K., Al-Shaer, E., Cardenas, A., ... & Moreau, D. (2018, December). Programmable System Security in a Software-Defined World–Research Challenges and Opportunities. In NSF Workshop on Programmable System Security in a Software Defined World. National Science Foundation. https://people.engr.tamu.edu/guofei/download/NSF_SPS_Workshop_Report.pdf
6. Liu, W., Chang, C. H., Wang, X., Liu, C., Fung, J. M., Ebrahimabadi, M., ... & Basu, K. (2021). Two sides of the same coin: Boons and banes of machine learning in hardware security. IEEE Journal on Emerging and Selected Topics in Circuits and Systems, 11(2), 228-251. https://ieeexplore.ieee.org/iel7/5503868/5751207/09442769.pdf
7. Tuladhar, A. (2021). Human-Centered Cybersecurity Research—Anthropological Findings from Two Longitudinal Studies (Doctoral dissertation, University of South Florida). https://digitalcommons.usf.edu/cgi/viewcontent.cgi?article=10443&context=etd
8. Collier, Z. A., & Sarkis, J. (2021). The zero trust supply chain: Managing supply chain risk without trust. International Journal of Production Research, 59(11), 3430-3445. https://helda.helsinki.fi/bitstream/10138/565227/1/PostPrintHankenCollierSarkisIJPR2020.pdf
9. Zaber, M. (2020). Automating Cyber Analytics. https://scholar.smu.edu/cgi/viewcontent.cgi?article=1015&context=engineering_compsci_etds
10. Zhang, C., Patras, P., & Haddadi, H. (2019). Deep learning in mobile and wireless networking: A survey. IEEE Communications surveys & tutorials, 21(3), 2224-2287.https://arxiv.org/pdf/1803.04311
11. Fontana, G. (2020). Social Media at War. The Case of Kurdish Fighters and Their Impact on the Perception of the Ongoing Anti-ISIS Conflict in Western Countries. EUROPEAN CYBERSECURITY JOURNAL, 6(2), 91-96. https://iris.unitn.it/bitstream/11572/408058/2/ECJ_vol6_issue2_online_v2.pdf

