An Adversarial Strategic Game for Machine Learning as a Service using System Features

Extended Abstract

Guoxin Sun
University of Melbourne
Melbourne, Australia
guoxins@student.unimelb.edu.au

Tansu Alpcan
University of Melbourne
Melbourne, Australia
tansu.alpcan@unimelb.edu.au

Seyit Camtepe
CSIRO Data61
Sydney, Australia
seyit.camtepe@data61.csiro.au

Andrew C. Cullen
University of Melbourne
Melbourne, Australia
andrew.cullen@unimelb.edu.au

Benjamin I. P. Rubinstein
University of Melbourne
Melbourne, Australia
brubinstein@unimelb.edu.au

ABSTRACT

Machine-learning-as-a-service (MLaaS) dramatically decreases the barrier of entry to machine learning through accessible, externally trained model building and deployment. However, numerous studies have shown that MLaaS models are vulnerable to adversarial attacks, which can alter input data with small perturbations and deceive the underlying machine learning algorithms. In this paper, we propose a novel approach for detecting and mitigating adversarial attacks in MLaaS. Our approach leverages previously overlooked system-level features in combination with data-driven methods to detect the generation process of adversarial examples. To guide the mitigation process, we model the dynamic interactions between an adaptive adversary, an imperfect anomaly detector, and a broader defensive system as a non-cooperative strategic game with imperfect information. We use experimental data from a realistic small-scale MLaaS ecosystem to construct the game components, such as players' utilities and detection accuracy. Our experimental results indicate that an adversarial attack against MLaaS defended by our method requires up to six times more cloud service accounts compared to other state-of-the-art frameworks. These promising results demonstrate the importance of considering realistic system settings when developing and evaluating adversarial attacks and defenses.

KEYWORDS

Security games; Adversarial defense; Machine-Learning-as-a-Service.

ACM Reference Format:
Guoxin Sun, Tansu Alpcan, Seyit Camtepe, Andrew C. Cullen, and Benjamin I. P. Rubinstein. 2023. An Adversarial Strategic Game for Machine Learning as a Service using System Features : Extended Abstract. In *Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), London, United Kingdom, May 29 – June 2, 2023*, IFAAMAS, 3 pages.

1 INTRODUCTION

Machine Learning as a Service (MLaaS) significantly increases the accessibility of deep learning by obviating the need for acquiring both hardware and human talent. However, the very nature of MLaaS as a product designed for non-experts presents it as an enticing target for attackers. Of particular concern are black-box, query-based mechanisms, in which adversarial examples are constructed at inference time. This is achieved through large volumes of queries, which iteratively elicit a change in the machine learning model's behaviour [5, 6]. Such attacks pose significant security and financial risks to both users and service providers [14, 16]. The task of detecting and repelling such attacks is complicated, as attackers often attempt to minimize the detectability of their attacks. In many contexts, this is represented by minimizing the normed distance between true and adversarial examples. However, an equally important practical metric for MLaaS is the number of accounts an attacker must create on the service to successfully attack the model.

While existing defensive works cover adversarial robustness on both the model- [13, 17] and sample-levels [6, 12, 15], they often overlook system-level features, which are an important fingerprint of attacker behavior. Furthermore, the MLaaS environment also introduces an inherent need to not only detect attacks, as is the case in prior works, but also to introduce structures that automatically mitigate both the existence and impact of attacks.

In this paper, we present a novel approach for detecting and mitigating adversarial attacks in MLaaS. Our key contributions can be summarized as follows.

(1) We utilize *system-level features* for the purpose of adversarial attack detection as a novel alternative to existing model input- or structure-focused approaches.
(2) To defend against adaptive adversaries, we design a *large-scale security game* to guide the attack mitigation process, where we model the complex interactions between multiple parties and consider realistic attack detector characteristics.
(3) Different *attacker query behavior* is considered for the first time to replicate the complexity of real-world attacks and defenses when implementing a query-efficient black-box adversarial attack [5].
(4) We develop a *small-scale MLaaS testbed*, which allows us to monitor realistic system-level features under both benign

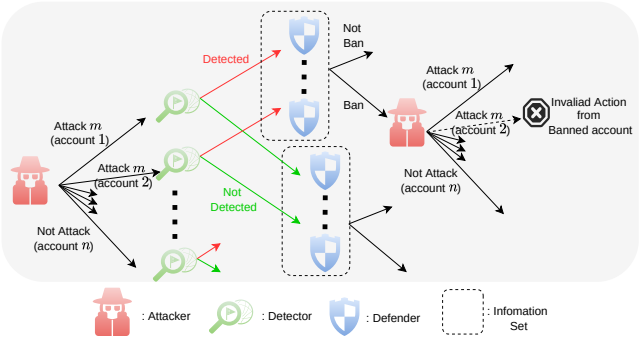

Figure 1: High-level representation of the game, with a subset of action pathways shown for clarity. Each game stage starts with the *Attacker* deciding how aggressive the Hop-Skip-Jump attack should be or not to attack at all for each of the available accounts, followed by the detector that provides the *Defender* a detection report. Lastly, the *Defender* decides on the attack mitigation strategy. The entire game contains multiple stages and terminates when all of the Attacker's pre-created accounts are suspended by the Defender or the maximum game depth is reached.

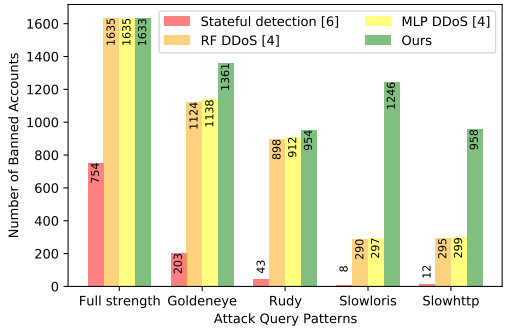

Figure 2: The number of accounts banned when the attacker initiates adversarial attacks. The attack query patterns (x-axis) are extracted from the CIC DoS dataset. Attacking MLaaS is more costly when our approach is deployed.

and adversarial operating conditions. Our realistic server setup is powered by the Flask web framework and Docker containers, with orchestration by way of Kubernetes.

2 SYSTEM-LEVEL ATTACK DETECTION

To detect the generation process of adversarial attacks, we leverage a set of system-level features, including resource consumption (i.e., CPU and memory consumption captured by *cAdvisor* [7]), system calls (monitored by *Falco* [1]), user logs (provided by *flask* micro web framework [8]), and communication-related data (e.g., packet length and inter-arrival time monitored by *tShark* [2]).

We employ a supervised learning approach to produce an anomaly detection model using system features generated both by benign and malicious clients. Specifically, we train several representative machine learning models, including deep feed-forward neural networks (DNNs), support vector machines, and gradient boosting techniques. After comparing the performance of these models, we have selected the DNN model as our final detector for this application due to its superior capability.

3 SECURITY GAME MODEL

The attack mitigation process is modeled as interactions between the *Attacker* (an intelligent adversary), the imperfect anomaly detector, and the *Defender* (the defense system). In our problem setting, the *Attacker* has a total number of N accounts created beforehand, and each account has six action options: five attack query patterns and one benign user pattern, all extracted from the CIC DoS dataset [9]. The *Defender* optimizes the (MLaaS cloud) account banning mitigation strategy based on the imperfect detection report and the anticipation of the downstream attack effects. This structure lends itself to a multi-stage, non-cooperative dynamic game in the presence of imperfect information.

We use experimental data from a realistic small-scale MLaaS ecosystem to construct the game components (e.g., players' utilities and detection accuracy). We introduce chance nodes into the game tree to represent the attack detector's characteristics taking the detector's inaccuracies (i.e., false positives and false negatives against different attack types) into consideration. A *chance node* can be seen as a fictitious (nature) player who performs actions according to a fixed probability distribution [3, 18]. We calibrate the probability distributions of the *chance* nodes based on real detection performance from the experiments performed in our MLaaS system. We utilize Nash equilibrium solutions of the game to guide the *Defender*. To reduce computational complexity, the Nash solutions are approximated using the external sampling Monte Carlo Counterfactual Regret Minimization (MCCFR) [11] algorithm in OpenSpiel [10]. This game structure and information flows are summarised in Figure 1.

4 EXPERIMENTAL EVALUATION

We compare our method with state-of-the-art defense approaches, including one adversarial attack detection method, the query-based stateful detection method [6], and two detectors from a DDoS attack detection proposal [4]. We used the number of banned accounts within a fixed attack period as our comparison metric, which is the same metric as used in the stateful detection method [6]. As shown in Figure 2, our defense method outperformed the comparison baselines, requiring the attacker to create significantly more accounts in the MLaaS system than when attacking the stateful method or the DDoS detection methods. For example, the attacker needed 6152 accounts to conduct attacks against our proposal, compared to only 1020 accounts needed to attack the stateful method. Additionally, the performance of the DDoS detection methods decreases dramatically as the attack pattern becomes more stealthy (e.g., from Full strength to Slowloris). Our method performs consistently well against various attack patterns. Across the set of tested scenarios, we saw a reduction in the number of accounts required for completing the attack by between 82.5 and 98.9%, for the stateful and the DDoS detection methods respectively, but only drops by about 41.6% with our method.

ACKNOWLEDGMENTS

This research was supported through the Next Generation Technologies Fund Program in partnership with the Defence Science and Technology Group in the Department of Defence, Australia, and the CSIRO Data61 Ph.D. scholarship program.

REFERENCES

[1] 2022. The Falco project. https://falco.org/docs/
[2] 2022. Tshark(1) Manual Page. https://www.wireshark.org/docs/man-pages/tshark.html
[3] Tansu Alpcan and Tamer Başar. 2010. *Network security: A decision and game-theoretic approach*. Cambridge University Press.
[4] Mazhar Javed Awan, Umar Farooq, Hafiz Muhammad Aqeel Babar, Awais Yasin, Haitham Nobanee, Muzammil Hussain, Owais Hakeem, and Azlan Mohd Zain. 2021. Real-Time DDoS Attack Detection System Using BigData Approach. *Sustainability* 13, 19 (2021), 10743.
[5] Jianbo Chen, Michael I Jordan, and Martin J Wainwright. 2020. HopSkipJumpAttack: A Query-Efficient Decision-Based Attack. In *IEEE Symposium on Security and Privacy (SP)*. 1277–1294.
[6] Steven Chen, Nicholas Carlini, and David Wagner. 2020. Stateful Detection of Black-Box Adversarial Attacks. In *Proceedings of the 1st ACM Workshop on Security and Privacy on Artificial Intelligence*. 30–39.
[7] Google. 2022. Google/cadvisor: Analyzes resource usage and performance characteristics of running containers. https://github.com/google/cadvisor
[8] Miguel Grinberg. 2018. *Flask web development: developing web applications with python*. " O'Reilly Media, Inc.".
[9] Hossein Hadian Jazi, Hugo Gonzalez, Natalia Stakhanova, and Ali A Ghorbani. 2017. Detecting HTTP-based application layer DoS attacks on web servers in the presence of sampling. *Computer Networks* 121 (2017), 25–36.
[10] Marc Lanctot, Edward Lockhart, Jean-Baptiste Lespiau, Vinicius Zambaldi, Satyaki Upadhyay, Julien Pérolat, Sriram Srinivasan, Finbarr Timbers, Karl Tuyls, Shayegan Omidshafiei, Daniel Hennes, Dustin Morrill, Paul Muller, Timo Ewalds, Ryan Faulkner, János Kramár, Bart De Vylder, Brennan Saeta, James Bradbury, David Ding, Sebastian Borgeaud, Matthew Lai, Julian Schrittwieser, Thomas Anthony, Edward Hughes, Ivo Danihelka, and Jonah Ryan-Davis. 2019. OpenSpiel: A Framework for Reinforcement Learning in Games. *CoRR* abs/1908.09453 (2019). arXiv:1908.09453 [cs.LG] http://arxiv.org/abs/1908.09453
[11] Marc Lanctot, Kevin Waugh, Martin Zinkevich, and Michael Bowling. 2009. Monte Carlo Sampling for Regret Minimization in Extensive Games. *Advances in Neural Information Processing Systems* 22 (2009).
[12] Jiajun Lu, Theerasit Issaranon, and David Forsyth. 2017. SafetyNet: Detecting and Rejecting Adversarial Examples Robustly. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 446–454.
[13] Aran Nayebi and Surya Ganguli. 2017. Biologically inspired protection of deep networks from adversarial attacks. *arXiv preprint arXiv:1703.09202* (2017).
[14] Luca Pajola and Mauro Conti. 2021. Fall of Giants: How popular text-based MLaaS fall against a simple evasion attack. In *2021 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 198–211.
[15] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. 2016. Distillation as a Defense to Adversarial Perturbations Against Deep Neural Networks. In *IEEE Symposium on Security and Privacy (SP)*. 582–597.
[16] Adnan Qayyum, Aneeqa Ijaz, Muhammad Usama, Waleed Iqbal, Junaid Qadir, Yehia Elkhatib, and Ala Al-Fuqaha. 2020. Securing machine learning in the cloud: A systematic review of cloud machine learning security. *Frontiers in big Data* 3 (2020), 587139.
[17] Ali Shafahi, Mahyar Najibi, Mohammad Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. 2019. Adversarial Training for Free! *Advances in Neural Information Processing Systems* 32 (2019).
[18] Guoxin Sun, Tansu Alpcan, Benjamin IP Rubinstein, and Seyit Camtepe. 2021. Strategic Mitigation Against Wireless Attacks on Autonomous Platoons. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. 69–84.