Machine Learning under Malware Attack

Raphael Labaca-Castro

Machine Learning under Malware Attack

Springer Vieweg

Raphael Labaca-Castro
München, Germany

The following text is a reprint of a dissertation with homonymous title submitted to the Universität der Bundeswehr München in 2022.

ISBN 978-3-658-40441-3 ISBN 978-3-658-40442-0 (eBook)
https://doi.org/10.1007/978-3-658-40442-0

This Springer Vieweg imprint is published by the registered company Springer Fachmedien Wiesbaden GmbH, part of Springer Nature.
The registered company address is: Abraham-Lincoln-Str. 46, 65189 Wiesbaden, Germany

For the loving souls who inspired it
but will never read it.

Acknowledgments

I would like to sincerely thank everyone who supported me throughout this adventure. This PhD has been an amazing journey.

First and foremost, I am deeply grateful to my *Doktormutter*, Prof. Dr. Gabi Dreo Rodosek, for trusting me to join her team, for providing me with all the support and freedom I needed during my research, for connecting me with excellent professionals, and above all for always fostering a positive attitude when needed.

I would like to thank my co-advisor, Prof. Dr. Lorenzo Cavallaro, for his expertise, experience, and great advice for my research, as well as for bringing joy to our meetings and for the ice breakers and pleasant conversations during the cold times of video calls. I would also like to thank him for agreeing to work with me even after I said that all pasta is the same.

I would like to thank my colleagues and friends at the Research Institute CODE, especially Nils and Klement, with whom this PhD was very fun. I would like to thank them for our countless technical discussions, even though the three of us worked in different research areas. Over these years, I learned so much from other fields as I did from my own, and that was mostly because of you. I would also like to sincerely thank Volker and the university staff, who have been constantly helping me since the beginning of this journey to navigate this process with all resources needed and to bypass any administrative hurdles.

Thanks are due to Google X for hosting me twice as an AI resident in both California and Munich, to Joe for allowing me to gain his attention, and to everyone that I have worked with. I am really grateful for this opportunity to collaborate with world-class researchers. Working with you made me realize how intellectual humility seems to be related to acquired knowledge.

Thanks are also due to all colleagues and collaborators, to Christian and Jessica for being my first publication team, to Corinna for helping me achieve clarity throughout my first steps in PhD life, to Luis for the many interesting discussions around machine learning, and to Fabio and Feargus for the countless video calls we had while preparing our research. I hope we all meet personally soon in London or Munich. Thanks to Juanma and Fran who took the time to review the abstracts. I would also like to thank Marc and my team at work, who showed me nothing but patience and support while I was finalizing my dissertation and working full time.

Though not directly connected only to this PhD, I would like to thank my dear friends, especially those in Paysandú, Uruguay. Despite the distance, you have been there for me as a second family, and I am extraordinarily fortunate for having you in my life.

This work would not have been possible without my loving wife, Bel. Thank you so much for letting me share the adventure of life with you. Thank you for the frequent laughter, our road trips with *mates*, and for listening to me speak about my research on almost a daily basis. I could never imagine better companionship throughout these years.

Thank you Su for changing my life since I invaded yours, and thank you Lache, who is no longer with us but would have asked me to fix the television remote with my new degree. Thank you for the many lessons, especially for helping me understand that responsibility and discipline may lead to achieve great things, but at the end nothing will be more important than a united family.

Finally, I would like to particularly thank my parents for giving me life, love, grit, and strength, for raising me in multiple places, for kick-starting my passion toward discovering the world, for ultimately contributing to my *Fernweh*, and for supporting me in tough moments when I was unsure of myself but somehow you were. No effort will ever compare to what my family did for me to be here today.

I would like to dedicate a few last words to those who are now starting their academic journey. There are many reasons to not pursue a PhD. However, being able to explore new knowledge is an inspiring endeavor that is worth chasing. Learning to learn is perhaps the most underestimated task, yet it is highly useful to succeed in research and life.

While I have always imagined a PhD degree would be the end, I understand now that it is only the beginning, and I am glad to finally notice that there is no end.

Abstract

Machine Learning (ML) has become key in supporting decision-making processes across a wide array of applications, ranging from autonomous vehicles and streaming recommendations to network traffic analysis. Because of the massive amounts of information available, researchers and practitioners have largely focused on improving the performance of ML models. These efforts led to novel applications that can now overcome human performance nearly in every domain.

Despite being highly accurate, these algorithms have been shown to be prone to returning incorrect predictions. This means that the trustworthiness of a model may be compromised in terms of both training and testing time. In other words, committed adversaries may be able to manipulate input data (e.g., images and network traffic) to resemble objects from another class, performing what is known as *evasion* attacks. They may also use another strategy called *poisoning*, which includes injecting undesirable data into the training set. Both approaches aim to mislead the model to predict the wrong label for a given object.

While adversarial attacks may target how a model is induced to predict a certain class for an object (e.g., classifying a red traffic light as green), this is normally sufficient with a wrongly predicted label or the opposite class for binary classification, which is generally the case in the context of malware (e.g., classifying malicious software as harmless).

In the event of manipulating objects for evasion attacks, these are known as *adversarial examples* and pose multiple security risks. On many occasions, as in malware classification, such behavior needs to be further examined to assess the degree to which predictions can be trusted.

Therefore, studying adversarial attacks can help identify systemic weaknesses in ML classifiers. These attacks reveal weak spots in the model that allow carefully manipulated objects to be incorrectly classified, hence compromising the quality of predictions. In fact, by investigating multiple strategies to generate successful adversarial examples, models can be evaluated from multiple perspectives and potentially hardened against adversarial examples. However, it is worth noting that while attacks generally materialize in the feature domain and are convertible to the problem space, where they exist in the real world, this is not always the case in the context of malware, especially in Portable Executable (PE) files. In this context, generating real adversarial malware examples often requires modifications that preserve binary integrity at the byte level. Thus, creating effective attacks using PEs is not a trivial task.

In this study, we present a framework that contains a suite of input-specific attacks using PE malware targeting ML-based classifiers, in which the adversaries have limited knowledge about target classifiers. We explore multiple approaches in which the adversary leverages hard labels from the model and does not have any prior knowledge of the architecture or access to the training data. To deeply understand the model's behavior, we additionally study full-knowledge attacks based on gradient information.

We introduce universal adversarial attacks in the problem space for PEs. The underlying goal here is to show whether the generation of adversarial examples can be automated and generalized without relying exclusively on input-specific attacks to generate effective adversarial examples.

We also propose a defense strategy that leverages knowledge from the aforementioned attacks to increase the cost of generating adversarial examples and, therefore, improve the target model against carefully crafted objects produced by adaptive adversaries. We envision a holistic approach that facilitates the identification of systemic vulnerabilities and enhances the classifier's resilience at a reasonable cost.

Next, we perform a statistical analysis of malware features by evaluating the impact of real-world attacks in the feature domain, which provides clarity for model predictions under unexpected input.

Finally, we release our Framework for Adversarial Malware Evaluation and make the source code available to encourage participation and further research into this fascinating topic and promote the evaluation and building of more resilient malware classifiers.

Kurzfassung

Maschinelles Lernen (ML) ist zu einem zentralen Baustein für die Unterstützung von Entscheidungsprozessen in einer Vielzahl von Anwendungen geworden, die von autonomen Fahrzeugen über Streaming-Empfehlungen bis hin zur Analyse des Netzverkehr reichen. Aufgrund der großen Menge an verfügbaren Informationen haben sich Forscher und Praktiker weitgehend auf die Verbesserung der Leistung von ML-Modellen konzentriert. Diese Bemühungen haben zu neuartigen Anwendungen geführt, die nun in fast jedem Bereich die menschliche Leistung übertreffen können.

Es hat sich gezeigt, dass diese Algorithmen trotz ihrer hohen Genauigkeit dazu neigen, falsche Vorhersagen zu machen. Dies bedeutet, dass die Vertrauenswürdigkeit eines Modells sowohl in Bezug auf die Trainings- als auch die Testzeit beeinträchtigt werden kann. Mit anderen Worten: Engagierte Angreifer können die Eingabedaten (z. B. Bilder und Netzverkehr) so manipulieren, dass sie Objekten einer anderen Klasse ähneln, und damit sogenannte *evasion*-Angriffe durchführen. Sie können auch eine andere Strategie anwenden, welche als *poisoning* bezeichnet wird und das Einspeisen unerwünschter Daten in den Trainingsdatensatz beinhaltet. Beide Ansätze zielen darauf ab, das Modell dazu zu verleiten, die falsche Bezeichnung für ein bestimmtes Objekt vorherzusagen.

Feindliche Angriffe können darauf abzielen, wie ein Modell dazu gebracht wird, eine bestimmte Klasse für ein Objekt vorherzusagen (z. B. eine rote Ampel als grün zu klassifizieren). Es reicht hierbei normalerweise aus, wenn ein falsches Label oder, bei einer binären Klassifizierung, die entgegengesetzte Klasse vorhergesagt wird, was im Allgemeinen im Zusammenhang mit Malware der Fall ist (z. B. die Klassifizierung von bösartiger Software als harmlos).

Manipulierte Objekte für Ausweichangriffe werden als *adversarial examples* bezeichnet und stellen mehrere Sicherheitsrisiken dar. In vielen Fällen, wie z. B.

bei der Klassifizierung von Schadsoftware, müssen diese manipulierten Objekte weiter untersucht werden, um zu beurteilen, inwieweit den Vorhersagen vertraut werden kann.

Die Untersuchung feindlicher Angriffe kann daher dazu beitragen, systemische Schwachstellen in ML-Klassifikatoren zu ermitteln. Diese Angriffe decken Schwachstellen im Modell auf, die es ermöglichen, dass sorgfältig manipulierte Objekte falsch klassifiziert werden, wodurch die Qualität der Vorhersagen beeinträchtigt wird. Durch die Untersuchung mehrerer Strategien zur Generierung erfolgreicher adversarial examples können Modelle aus verschiedenen Blickwinkeln bewertet und potenziell gegen adversarial examples abgehärtet werden. Es ist jedoch anzumerken, dass Angriffe in der Regel in der Merkmalsdomäne (*feature-space*) durchgeführt und anschließend in den Problemraum (*problem-space*) konvertiert werden, in dem sie in der realen Welt vorkommen. Dies ist im Zusammenhang mit Malware, insbesondere bei Portable Executable (PE)-Dateien, nicht immer der Fall. Die Generierung von echten adversarial malware examples erfordert oft Änderungen, welche die binäre Integrität auf Byte-Ebene bewahren. Daher ist die Erstellung effektiver Angriffe mit PEs keine triviale Aufgabe.

In dieser Studie stellen wir ein Framework vor, welches eine Reihe von eingabespezifischen Angriffen mit PE-Malware enthält. Diese zielen auf ML-basierte Klassifikatoren ab, wobei die Angreifer nur begrenzte Kenntnisse über die Zielklassifikatoren haben. Wir untersuchen mehrere Ansätze, bei denen der Angreifer feste Kennzeichnungen des Modells ausnutzt und keine Vorkenntnisse über die Architektur oder Zugriff auf die Trainingsdaten hat. Um das Verhalten des Modells besser zu verstehen, untersuchen wir zusätzlich Angriffe bei vollständigem Wissen, die auf Gradienteninformationen basieren.

Wir stellen universelle gegnerische Angriffe im Problemraum für PEs vor. Das zugrunde liegende Ziel ist es, zu zeigen, ob die Generierung von adversarial examples automatisiert und verallgemeinert werden kann, ohne sich ausschließlich auf eingabespezifische Angriffe zu verlassen, um effektive adversarial examples zu generieren.

Wir schlagen außerdem eine Verteidigungsstrategie vor, die das Wissen aus den oben erwähnten Angriffen nutzt, um die Kosten für die Generierung von feindlichen Angriffen zu erhöhen und somit das Zielmodell gegen sorgfältig ausgearbeitete Objekte zu verbessern, die von adaptiven Gegnern erzeugt werden. Wir stellen uns einen ganzheitlichen Ansatz vor, der die Identifizierung von systemischen Schwachstellen erleichtert und die Widerstandsfähigkeit des Klassifikators zu vertretbaren Kosten erhöht.

Als Nächstes führen wir eine statistische Analyse von Malware-Merkmalen durch, indem wir die Auswirkungen realer Angriffe im Merkmalsdomäne bewerten, was Klarheit für Modellvorhersagen bei unerwarteten Eingaben schafft.

Abschließend veröffentlichen wir unser Framework for Adversarial Malware Evaluation und stellen den Quellcode zur Verfügung, um weitere Forschung zu diesem faszinierenden Thema zu fördern und die Evaluierung und Entwicklung von widerstandsfähigeren Malware-Klassifikatoren zu unterstützen.

Resumen

El aprendizaje automático, en inglés conocido como machine learning (ML), se ha convertido en un elemento clave para apoyar los procesos de toma de decisiones en un amplio abanico de aplicaciones, que van desde los vehículos autónomos y las recomendaciones en streaming hasta el análisis del tráfico de la red. Debido a la enorme cantidad de información disponible, los investigadores y profesionales se han centrado en gran medida en mejorar el rendimiento de los modelos de ML. Estos esfuerzos han dado lugar a nuevas aplicaciones que ahora pueden superar el rendimiento humano en casi todos los ámbitos.

A pesar de ser muy precisos, estos algoritmos han demostrado ser propensos a realizar predicciones incorrectas. Esto significa que la fiabilidad de un modelo puede verse comprometida tanto en el tiempo de entrenamiento como en el de prueba. En otras palabras, los adversarios pueden ser capaces de manipular los datos de entrada (por ejemplo, imágenes y tráfico de red) para que se parezcan a objetos de otra clase, realizando lo que se conoce como ataques de *evasión*. También pueden utilizar otra estrategia denominada *envenenamiento*, que incluye la inyección de datos no deseados en el conjunto de entrenamiento. Ambos enfoques tienen como objetivo inducir al modelo a predecir una clasificación incorrecta para un objeto determinado.

Para que los ataques adversarios logren el objetivo de inducir a un modelo a predecir una determinada clase para un objeto (por ejemplo, clasificar un semáforo rojo como verde), normalmente es suficiente con una etiqueta predicha erróneamente o la clase opuesta para la clasificación binaria, que es generalmente el caso en el contexto del malware (por ejemplo, clasificar código malicioso como inofensivo).

En el caso de la manipulación de objetos para ataques de evasión, éstos se conocen como *ejemplos adversarios* y suponen múltiples riesgos de seguridad. En

muchas ocasiones, como en la clasificación de malware, es necesario examinar más a fondo este tipo de comportamiento para evaluar el grado en que se puede confiar en las predicciones.

Por lo tanto, el estudio de los ataques adversarios puede ayudar a identificar las debilidades sistémicas de los clasificadores de ML. Estos ataques revelan puntos débiles en el modelo que permiten clasificar incorrectamente objetos cuidadosamente manipulados, comprometiendo así la calidad de las predicciones. De hecho, al investigar múltiples estrategias para generar ejemplos adversos exitosos, los modelos pueden ser evaluados desde múltiples perspectivas y potencialmente fortalecidos contra ejemplos adversarios. Sin embargo, cabe señalar que, si bien los ataques suelen materializarse en el dominio de las características (*feature space*) y son convertibles al dominio del problema (*problem space*), donde existen en el mundo real, no siempre es así en el contexto del malware, especialmente en los archivos ejecutables portátiles (EP). En este contexto, la generación de ejemplos reales de malware adverso suele requerir modificaciones que preserven la integridad binaria a nivel de bytes. Por lo tanto, crear ataques eficaces utilizando EP no es una tarea trivial.

En este estudio, presentamos un *framework* que contiene un conjunto de ataques específicos de entrada (*input-specific*) utilizando malware EP dirigido a clasificadores basados en ML, sobre los cuales los adversarios tienen un conocimiento limitado. Exploramos múltiples enfoques en los que el adversario aprovecha las etiquetas duras (*hard labels*) del modelo sin tener ningún conocimiento previo de la arquitectura ni acceso a los datos de entrenamiento. Para comprender en profundidad el comportamiento del modelo, estudiamos además los ataques de conocimiento completo basados en la información de gradiente.

Formulamos ataques adversarios universales en el espacio de problemas de los EP. El objetivo subyacente aquí es mostrar si la generación de ejemplos adversarios puede ser automatizada y generalizada sin depender exclusivamente de ataques específicos de entrada para generar ejemplos adversarios efectivos.

También proponemos una estrategia de defensa que aprovecha el conocimiento de los ataques antes mencionados para aumentar el coste de la generación de ejemplos adversarios y, por lo tanto, mejorar la seguridad del modelo objetivo contra los objetos cuidadosamente elaborados producidos por adversarios adaptativos. La intención es generar un enfoque holístico que facilite la identificación de vulnerabilidades sistémicas y mejore la resistencia del clasificador a un coste razonable.

Posteriormente, realizamos un análisis estadístico de las características del malware evaluando el impacto de los ataques del mundo real en el dominio de

las características, lo que proporciona claridad para las predicciones del modelo bajo entradas inesperadas.

Por último, publicamos nuestro Framework for Adversarial Malware Evaluation y ponemos a disposición el código fuente para fomentar la participación y la investigación de este fascinante tema y promover la evaluación y construcción de clasificadores de malware más resistentes.

Resumo

O aprendizado de máquina, em inglês machine learning (ML), tornou-se fundamental no apoio aos processos de tomada de decisão numa vasta gama de aplicações, desde veículos autônomos e recomendações de streaming até à análise do tráfego na rede. Devido à enorme quantidade de informação disponível, os pesquisadores e profissionais concentraram-se em grande parte na melhoria do desempenho dos modelos ML. Estes esforços conduziram a novas aplicações que podem agora superar o desempenho humano em quase todos os domínios.

Apesar de serem altamente precisos, estes algoritmos têm demonstrado ser propensos a retornar previsões incorretas. Isto significa que a fiabilidade de um modelo pode ser comprometida tanto em termos de formação como de teste. Em outras palavras, os adversários comprometidos podem ser capazes de manipular dados de entrada (por exemplo, imagens e tráfego de rede) para se assemelharem a objetos de outra classe, realizando o que é conhecido como ataques de *evasão*. Eles podem também utilizar outra estratégia chamada *poisoning*, que inclui a injeção de dados indesejáveis no conjunto de dados treino. Ambas as abordagens visam enganar o modelo para prever a etiqueta errada para um determinado objeto.

Embora os ataques contraditórios possam visar a forma como um modelo é induzido a prever uma determinada classe para um objeto (por exemplo, classificar um semáforo vermelho como verde), isto é normalmente suficiente com uma etiqueta mal prevista ou a classe oposta para a classificação binária, o que é geralmente o caso no contexto de malware (por exemplo, classificar um software malicioso como inofensivo).

No caso de manipulação de objetos para ataques de evasão, estes são conhecidos como *exemplos adversários* e representam múltiplos riscos de segurança.

Em muitas ocasiões, como na classificação de malware, tal comportamento precisa ser examinado mais detalhadamente para avaliar o grau em que as previsões podem ser confiáveis.

Portanto, o estudo de ataques adversos pode ajudar a identificar fraquezas sistêmicas nos classificadores ML. Estes ataques revelam pontos fracos no modelo que permitem a classificação incorreta de objetos cuidadosamente manipulados, comprometendo assim a qualidade das previsões. De fato, ao investigar múltiplas estratégias para gerar exemplos adversários bem-sucedidos, os modelos podem ser avaliados de múltiplas perspectivas e potencialmente fortalecidos contra exemplos adversários. No entanto, vale a pena notar que embora os ataques se materializem geralmente no domínio das características (*feature space*) e sejam convertíveis no espaço do problema (*problem space*), onde existem no mundo real, nem sempre é esse o caso no contexto do malware, especialmente nos arquivos Portable Executable (PE). Neste contexto, gerar exemplos reais de malware adversário requer frequentemente modificações que preservem a integridade binária ao nível do byte. Assim, a criação de ataques eficazes utilizando PE não é uma tarefa trivial.

Neste estudo, apresentamos um quadro que contém um conjunto de ataques específicos de entrada usando malware de PE que visam classificadores baseados em ML, no qual os adversários têm um conhecimento limitado sobre classificadores-alvo. Exploramos múltiplas abordagens nas quais o adversário aproveita etiquetas duras (*hard labels*) do modelo e não tem qualquer conhecimento prévio da arquitectura ou acesso aos dados de formação. Para compreender profundamente o comportamento do modelo, estudamos adicionalmente os ataques de conhecimento completo com base em informação de gradiente.

Formulamos ataques adversários universais no espaço do problema para os PEs. O objectivo aqui é mostrar se a geração de exemplos de adversários pode ser automatizada e generalizada sem depender exclusivamente de ataques específicos de entrada para gerar exemplos efectivos de adversários.

Propomos também uma estratégia de defesa que aproveita o conhecimento dos ataques mencionados anteriormente para aumentar o custo da geração de exemplos adversários e, portanto, melhorar o modelo alvo contra objetos cuidadosamente trabalhados produzidos por adversários adaptativos. A intenção é realizar uma abordagem holística que facilite a identificação de vulnerabilidades sistêmicas e aumente a resiliência do classificador a um custo razoável.

Posteriormente, realizamos uma análise estatística das características do malware, avaliando o impacto dos ataques do mundo real no domínio das características, o que proporciona clareza para as previsões do modelo sob entrada inesperada.

Finalmente, divulgamos o nosso Framework for Adversarial Malware Evaluation e disponibilizamos o código fonte para fomentar a participação e investigação adicional sobre este fascinante assunto e promover a avaliação e construção de classificadores de malware mais resistentes.

Contents

Acronyms

Framework-specific

AIMED	Automatic Intelligent Manipulations to Evade Detection
AIMED-RL	Automatic Intelligent Manipulations to Evade Detection with RL
ARMED	Automatic Random Manipulations to Evade Detection
FAME	Framework for Adversarial Malware Evaluation
GAME-UP	Generate Adversarial Malware Examples with Universal Perturbations
GAINED	Generative Adversarial Intelligent Network to Evade Detection
GRIPE	Gradient Relevant Injection to Portable Executables

Framework-agnostic

ACER	Actor Critic with Experience Replay
AML	Adversarial Machine Learning
APK	Android Package Kit
CFG	Control Flow Graph
CNN	Convolutional Neural Network
COFF	Common Object File Format
DDQN	Double DQN
DiDQN	Distributional DQN
DiDDQN	Distributional Double DQN
DLL	Dynamic Link Library
DQN	Deep Q-Network
FGSM	Fast Gradient Sign Method

FNR	False Negative Rate
FPR	False Positive Rate
GAN	Generative Adversarial Network
GBDT	Gradient Boosted Decision Tree
GNN	Graph Neural Network
GP	Genetic Programming
IDS	Intrusion Detection System
IAT	Import Address Table
IPS	Intrusion Prevention System
LightGBM	Light Gradient Boosting Machine
LK	Limited Knowledge
LR	Logistic Regression
MDP	Markov Decision Process
ML	Machine Learning
MTD	Moving Target Defense
OS	Operating System
PE	Portable Executable
PDF	Portable Document Format
PK	Perfect Knowledge
RL	Reinforcement Learning
SVM	Support Vector Machine
TPR	True Positive Rate
UAP	Universal Adversarial Perturbation
UER	Universal Evasion Rate
UPX	Ultimate Packer for eXecutables

List of Figures

List of Tables