Aalborg Universitet

Robustness of Defenses against Deception Attacks

Panum, Thomas Kobber

DOI (link to publication from Publisher):
10.54337/aau429781336

Publication date:
2021

Document Version
Publisher's PDF, also known as Version of record

Link to publication from Aalborg University

Citation for published version (APA):
Panum, T. K. (2021). *Robustness of Defenses against Deception Attacks*. Aalborg Universitetsforlag. https://doi.org/10.54337/aau429781336

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal -

Take down policy
If you believe that this document breaches copyright please contact us at vbn@aub.aau.dk providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from vbn.aau.dk on: December 05, 2025

ROBUSTNESS OF DEFENSES AGAINST DECEPTION ATTACKS

BY

THOMAS KOBBER PANUM

DISSERTATION SUBMITTED 2021

AALBORG UNIVERSITY
DENMARK

Robustness of Defenses against Deception Attacks

Ph.D. Thesis

Thomas Kobber Panum

Thesis submitted March 2021

Dissertation submitted:	March, 2021
PhD supervisor:	Prof. Jens Myrup Pedersen Department of Electronic Systems Aalborg University
PhD Co-supervisor:	Assoc. Prof. René Rydhof Hansen Department of Computer Science Aalborg University
PhD committee:	Associate Professor Ulrik M. Nyman (chairman) Aalborg University
	Professor Kevin Curran Ulster University
	Professor Søren Hauberg Technical University of Denmark
PhD Series:	Technical Faculty of IT and Design, Aalborg University
Department:	Department of Computer Science

ISSN (online): 2446-1628
ISBN (online): 978-87-7210-925-1

Published by:
Aalborg University Press
Kroghstræde 3
DK – 9220 Aalborg Ø
Phone: +45 99407140
aauf@forlag.aau.dk
forlag.aau.dk

© Copyright: Thomas Kobber Panum

Printed in Denmark by Rosendahls, 2021

Abstract

Security advancements of computer systems have caused adversaries to explore alternative entry points for their attacks. Instead of attacking the systems directly, attack vectors that are initiated by social interactions have increased in popularity[45, 1]. These types of attacks are known to exploit a variety of social influences to deceive victims into performing a harmful action intended by the adversary [22, 12]. Typical defense solutions attempt to detect these attacks using machine learning techniques. Numerous of these solutions have reporting impressive detection rates for these types of attacks. However, the existence of these seemingly effective solutions remain in strong contrast to the high frequency of attacks in real-world settings [39].

Having this contradicting discrepancy, motivated the research of this thesis, that seek to explore a fundamental property of such defenses, namely their performance against an adaptive adversary. This type of performance is *adversarial robustness*, and is fundamentally different from the ability to detect empirical attacks. This stems from the fact that adversarial robustness seeks to reflect the expected performance against attacks that actively attempt evade detection, which a solution might experience in real-world settings.

In this thesis, I initially set out to explore the adversarial robustness of defenses against a widely established type of deception attack, phishing attacks. In this process, define a set of axioms for the functional properties of attacks, that serve as guideline for assessing detection strategies that influential and recent methods have adopted. A part of this assessment, is a demonstration of relatively simple perturbation techniques that emphasize the fragility of the detection solutions. Additionally, it is shown that a detection solution that apply a deep metric model [2], is more vulnerable to known test-time attacks than initially reported. Consequently, suggesting a fragility of deep metric models similar to traditional classifiers that rely on neural network architectures.

A prerequisite for the assessment, was to establish a dataset for the study of machine learning approaches. This lead to the design a tool for gathering highly detailed information about websites and their interconnected structure of content.

The discovered fragility of deep metric models, motivated a formalization of a robust optimization for such models. This formalization contributed to the design of a powerful attack algorithm for test-time attacks, that account for previous uncertainties of sampling method and perturbation target. With the established attack algorithm in place, a proposed robust training objective enhance robustness among commonly used datasets within the field.

Overall, this research highlights that both influential and recent methods for detecting deception attacks contain relatively simple failure modes, when exposed to an adversary that seek evasion. Improvements to the underlying methods of recent solutions, demonstrated that their robustness can be enhanced. However, these results remain empirical thus further guarantees and proofs of attainable adversarial robustness are still open problems.

Resumé

Forbedringer af computersystemer har ført til at angribere er begyndt at afsøge nye former for angreb. Det har gjort at, istedet for at angribe systemerne direkte, er angreb som starter ved sociale interaktioner øget i popularitet[45, 1]. Disse typer angreb er kendt for at anvende en række sociale mekanismer til at bedrage ofrene til at udføre en række skadelige handlinger, som er ønsket af angriberen [22, 12]. Typiske forsvarsløsninger forsøger at detektere angrebene ved brug af maskinlæring teknikker. En lang række er disse løsninger har fremført imponerende detektionsrater for angreb. Dette faktum for effektivitet står dog i stor kontrast til den stigende frekvens af angreb, set i den virkelige verden [39].

Denne modstrid danner motivationsgrundlag for denne tese, som forsøger at afdække en essentiel egenskab for disse forsvarsløsninger, deres evne til at detektere angreb mod en adaptiv angriber. Denne evne kaldes *adversarial robustness*, og fundamentalt forskellig fra evnen til at kunne detektere emperiske angreb. Det skyldes det faktum at adversarial robustness forsøger at beskrive den forventlige detektionsevne mod angreb som aktivt prøver at undvige detektion, hvilket er realistisk scenarie den virkelige verden.

Denne tese starter ud med at udforske etablerede forsvarsløsningers mod et velkendt form for bedragelsesangreb, phishing angreb. Dette gøres ved at definere en mængde af aksiomer som omhandler de funktionsmæssige egenskaber som denne type angreb besider. Disse aksiomer anvendes som en guideline til at evaluere detektionsstrategier som har været betydningsrige eller afspejler nyere metoder. En del af denne evaluering er en demonstration af simple maskerings teknikker som udtrykker skrøbeligheden af forsvarsløsningerne. Udover dette, bliver det tydelige gjort at en nyere metode som anvender en *deep metric model* [2] er mere sårbar overfor *test-time attacks* end den oprindelig evaluering. Dette resultatet udtrykker en potentiel skrøbelighed af deep metric modeller som er sammenlignlig med traditionelle klassifikationsmodeller som også anvender neurale netværk arkitekture.

Et krav for at kunne udføre denne evaluering var at etablere et datasæt til reproducere den nævnte maskinlæringsmodel. Dette krav førte til design af et værktøj som kan indsamle en detaljeret information omkring websites og deres interne struktur af indhold.

Den fundne skrøbelighed af deep metric modeller motiverede en formalisering af et robust optimerings mål for denne type af modeller. Denne formalisering førte til design af en kraftfuld angrebsalgoritme, som tager højde for de tidligere usikkerheder omkring sampling metoder og input type. Med den etablerede angrebsalgoritme formuleres et *robust training objective* til af for højde robustheden mod en række populære datasæt indenfor problemområdet.

Forskningen fra denne tese fremhæver at både betydningsrige og nyere metoder til detektion af bedragelsesangreb indeholder en række simple fejl, som kommer til udtryk når en angriber aktivt prøver at undgå detektion. Ligeledes bliver der fremvist at nogle af de underliggende metoder kan forbedres med højere robusthed. Disse resultater forbliver dog empiriske, og efterlader de omtalte metoder i en tilstand hvor garantier og beviser for adversarial robustness stadig er et åbent problem.

Acknowledgments

This section, and the limited amount of people mentioned here, can never do justice for the immense support I have received throughout the creation of this thesis from friends, colleagues, and most of all, my family. Thereby, I would like to send my appreciation for the support from people not mentioned within this section.

Firstly, this thesis would have never been completed without my indulgent wife, Maria. Her support and indulgence enabled me to get through the stressful times and submission deadlines. Additionally, during the coming of this thesis, she became the mother of my two beloved children, Joanna and Jakob. They completely changed my perception on life, and bring eternal joy and cleaning tasks for their dad.

Earlier in my life, I was faced with a saint preaching the gospel of “practice makes perfect”, my hard-working and loving mom. Raising me almost single-handedly, while providing me with endless love and freedom during my childhood, have both inspired and enabled me to pursue the career path leading to this thesis.

Since I returned to Aalborg University, I have been thankful for the forthcomingness provided by my main supervisor, Jens Myrup Pedersen. Jens chose me as the candidate for the project despite having no prior collaboration history, and gave me the freedom to pursue interesting research directions, for that, I will always be forever grateful.

In times of misery and frustration, finishing a thesis can seem like an infeasible task. One person that will always shine as the light that supported me through these difficult times is my co-supervisor, René Rydhof Hansen. Without the advice and encouragement from René, I would not have been able to perform the research required for completing this thesis. Whether it was casual small talks, or a more formal setting with slides, our talks always left me with excitement, curiosity, and confidence. I hope someday I will be able to just partially reach the dissemination and supervision skills of René. Any future students of his should feel lucky to have received him as a supervisor.

As I entered the office of my starting date, I was met by my bigger “PhD brother”, Kaspar Hageman. Starting one month prior to me, Kaspar and I have shared our PhD journey both as individual researchers but also collaborators. Kaspar has been very indulgent for listening to (almost) any wild research idea that has crossed my mind while providing me with valuable feedback. Having a nearby and supportive person, experiencing similar challenges and frustrations as yourself, have been immensely helpful in getting through the difficult times and for that I will always be forever thankful.

As a part of my external stay, I was invited by Somesh Jha to UW-Madison in the

state of Wisconsin. This opportunity had a massive influence on my research, by giving me options I would never have imagined, and gave me insight into the mindset of extraordinary researchers. UW-Madison, being home to the legendary statistician George Box, and Somesh will always be a cornerstone in my academic career.

During the coming of the research for this theses, I received the title of *dad* and enlightenment about the fruit quantities consumed by newborns. Thankfully, my fruit-producing co-worker, Tatiana Kozlova Madsen, have has very supportive by both providing endless amounts of apples for my kids and great company during lunch.

Despite my personal research activities being slightly distant to the typical activities within the Wireless Communication Networks, the section head, Preben Mogensen has been very understanding and supportive.

Funding. Thanks to NVIDIA for their GPU Grant Program that supplied me with GPU resources for my research.

Dedication

I wish to dedicate this thesis to Bjarne Haugaard, the study counselor of my public school, that found me highly unfit for high school and further education at the time.

Contents

Preface

This thesis came together at Aalborg University's Department of Electronic Systems, while under the kind supervision of Prof. Jens Myrup Pedersen and co-supervision Assoc. Prof. René Rydhof Hansen (Aalborg University). Scientific contributions are represented in the form of a collection of papers. The selected papers are a mixture of peer-reviewed publications and a preprint currently under review. Peer-reviewed publications has had their layout revised to align with the thesis, however their content remains unchanged.

Chapter 1 provides an introduction and motivates the area of research of the thesis. Following this, a more detailed outline of the thesis is presented, in addition to highlighted contributions of each individual paper. Part I covers preliminary concepts to provide a background across topics central for the thesis. Part II contains the collection of papers that form the scientific contributions covered within the scope of this thesis. Lastly, Part III concludes on the research contributions of the thesis and discusses future directions of research.

Thomas Kobber Panum
Aalborg University, March 2021

Chapter 1

Introduction

Humans have the ability to perceive the world surrounding them and rely on this ability for visual recognition to solve a diversity of tasks [35]. Certain of these tasks include using perception as a measure for distinguishing identities, such as persons recognizing each other by their facial features and companies being identified by their logo. Inherently, this demand for distinction is often directly coupled with various levels of trust, and thus certain identities being associated with different privileges. As a measure to advance privileges, adversaries might seek to deceive, i.e. exploit the recognition to manipulate the identification, in order to obtain unintended privileges. Thereby, establishing a direct link between the robustness of the visual recognition and levels of security.

Efforts over the recent decades have sought out to automate visual recognition tasks, thus driving the design of systems that attempt to mimic human perception in a computational setting [32, 56, 21]. A popular technique for designing such systems is machine learning, which seeks to learn the ability from empirical observations [62, 41, 10].

Concretely, this technique seeks to "learn" a solution to the perception problem by formalizing it as an optimization problem. The objective of this optimization problem is to uncover a set of parameters for a mathematical model that yields a desired output and ideally achieve a low error rate. Recent advancements have enabled for effectively approximating parameter sets containing multi-millions parameters, as a measure to solve more complex problems, and have served useful for a variety of tasks, e.g. visual recognition [28]. These advancements have been associated with the term "deep learning" [20], and have been widely used for visual recognition tasks and greatly increase in the accuracy for these tasks [20]. This includes applications of object recognition [56, 21, 67], object detection [47, 46, 48], or visual similarity [55, 66, 50].

Similarly to human perception, these applications are reliant on the underlying models to achieve certain levels of robustness, as the inability thereof can have security implications and cause harmful consequences [17]. In certain scenarios, adversaries have incentive to actively explore how input can be perturbed such that the system outputs a desired outcome of the adversary. Concretely, the objective of the adversary is to alter benign input until the predictions become false, while the semantics of the input remain unchanged. This leads to new demands for robustness referred to as *adversarial robustness*. One particular class of applications, for which adversarial robustness is essential, is defense applications that infer whether a certain activity is adversarial or benign [65, 2]. These solutions are expected to inspect some type of activity, with the intent of intervening if it is considered malicious, as a measure to reduce or eliminate its harmful consequences. These applications are naturally challenged by the desires of adversaries, that seek to avoid intervention of their adversarial actions.

Effectively this poses additional challenges for the learning system, captured by machine learning, as it is continuously faced with activities from the adversary that actively change footprint in order to avoid intervention. From a statistical modeling perspective, this creates an out-of-distribution learning scenario where a model is desired to perform on input of a different distribution than input distribution during training. This is different from the frequently applied i.i.d. assumption, which assumes input distributions are independent and identical at training- and at test-time [43].

1 Thesis Statement

This thesis was established with the intent of investigating the robustness of defenses against deception attacks and further clarify properties of the underlying methods used for creating these defenses. Particularly, machine learning-based defenses that rely on empirical observations as a measure to design and evaluate inference.

This lead to the formulation of the following thesis statement:

> *Quantifying the effectiveness of defenses against deception attacks cannot only rely on historical attacks, as these attacks might under-represent the true attack surface. This can cause defenses to adopt features for inference which demonstrate useful for the under-representation of attacks, while being ineffective for the true attack surface. For machine learning-based defenses, which seek to automate the discovery of attack features, it can cause the defense to adopt features that are unrelated to the functional properties of attacks and thus let unseen attacks evade detection. Adversarial robustness remains an open question for machine learning methods, a new proposed technique improves this robustness for a model class that has seen use in recent defenses.*

2 Outline and Contributions

This thesis is structured into three parts, Part I covers preliminary concepts to provide a background on deception attacks (Chapter 2) and machine learning in a security perspective (Chapter 3). Part II contains a two-fold collection of papers, and their respective contributions. The first group covers papers that investigate research problems related to the thesis statement and serve as the *core* contributions, and are listed below alongside highlighted contributions of each paper.

- **Paper A** *(TMA 2019)*: Investigates the design of a sophisticated web crawler for gathering extensive information of websites using the interactions performed by highly complex parsers of browser engines. Concretely, the designed web crawler utilizes the Blink browser engine (found in Google Chrome), to crawl and gather subsequent web requests (media content, AJAX, and more) performed by the browser engine. Implementation of the crawler is open sourced and serves as a tool to gather relevant data of an environment in which deception attacks are known to be present.

- **Paper B** *(ICML 2020)*: Proposes a method for the discovery of Granger causality among interdependent event sequences with multiple event types. Previous work has often been limited by the flexibility or explainability of the underlying models, which cause them to be unable to uncover Granger causality for certain variations of event sequences. Fundamentally the proposed method applies a neural point process to an event sequence problem, and uncover the Granger causality by applying axiomatic attribution methods. Experiments demonstrate that the proposed method exceeds state-of-the-art methods across a variety of commonly used datasets.

- **Paper C** *(USENIX CSET 2020)*: Assesses the ability of an adversary to construct attacks that evade detection across influential and recent phishing detection solutions. Contributions of this work include defining a set of axioms for phishing attacks, identification of a series of common strategies used among selected detection solutions, and demonstrating the receptiveness of simple attacks for solutions of these strategies. These results suggest that previously stated performances are not reflecting the adversarial robustness, which is essential for these solutions to be useful and applicable. Closingly, a set of design guidelines are proposed to support future detection solutions to obtain higher adversarial robustness.

- **Paper D** *(In review)*: Explores the adversarial robustness of deep metric learning models trained using metric losses. Contributions of this study includes a formulation of the robust optimization objective for these types of models, and a proposed attack algorithm. Popular models are demonstrated to be fragile towards adversarial perturbations, similar to the traditional classification setting, contradicting results reported by other studies applying these techniques. Lastly, a proposed robust training objective improves the adversarial robustness throughout any of the applied models.

The second group of papers relates to the design of a security education platform named *Haaukins*. These papers cover research contributions related to the design and implementation of a solution for hosting virtual security exercises, while being suited for a higher education teaching environment, such as high schools and universities. These contributions established preliminary knowledge of the adversarial mindset for deception attacks. Papers of this group are listed below:

- **Paper E** *(ICALT 2019)*: Explores the design of a platform for teaching information security to high schools students. The work highlights how previous platforms either lack realism or are designed for professional competitions, which cause them to be unfit for education. Following this, the problem domain is analyzed to derive a set of design goals. These design goals motivate design choices and the implementation of the platform, which is free of use and provided as open source. The designed platform, named Haaukins, provides teachers with great flexibility to design and implement various types of exercises. Notably for our teaching at Aalborg University, we implemented a set of exercises that required the use of deception attacks for completion.

- **Paper F** *(EDUCON 2021)*: This work extends the previous platform to be applicable in higher degrees of education, particularly for a teaching environment found across universities. This extension covers an additional set of design goals that are being adopted by new technical solutions. These changes enable the platform to become more scalable, have more flexibility for user's tools, and provide a centralized method for obtaining complex exercises.

The last part of the thesis, Part III, concludes on thesis statement (Chapter 10) and provides perspectives on the future of the research field (Chapter 11).

Part I

Preliminary Concepts

Chapter 2

Deception Attacks

Private information is information meant only to be shared with certain entities. For this to be viable, one is implicitly expected to be able to distinguish entities by their identity. In scenarios of valuable information, this can establish an incentive for adversaries to explore the ability to fool the identification mechanism to obtain a false identity. This false identity attempts to mimic some form of existing identity, such that an entity with valuable private information is willing to exchange the information. Attacks that rely on this mechanism are what we refer to as *deception attacks*. Social engineering attacks is a class of attacks within computer security that have seen practical use of this mechanism [22]. These attacks seek to exploit the users of a given computer system, opposed to exploiting the computer system directly, by using social influences [12]. A social engineering attack that has seen frequent use in recent decades is *phishing attacks* [45, 1]. Phishing attacks have historically had varying definitions [3]. Lastdrager [31] performed a meta-analysis of literature on phishing attacks and reached the following consensual definition of phishing attacks:

> *Phishing is a scalable act of deception whereby impersonation is used to obtain information from a target.*

This definition, and the described attacks, serve as the presentation of deception attacks throughout the research carried out in this thesis.

Chapter 3

Machine Learning

For certain problems, it remains infeasible to explicitly formalize an algorithm capable of solving them. As an example, formalizing the functionality of human perception with manual labor and procedures found in a typical programming language is simply too complex. A popular technique for addressing some of these complex problems is machine learning, a class of algorithms that *learns* from *experience* [1]. The phase of learning in these algorithms is often referred to as *training* and involves solving a mathematical optimization problem. For the sake of relevance of this thesis, the focus of machine learning methods will remain in the domain of *supervised learning*, which involves training a model to perform prediction based on data that consist of input-output pairs. Supervised learning have largely been influenced by the principles of *Empirical Risk Minimization* [62], which formalize optimization objective as finding the set of model parameters $\theta \in \Theta$ that minimizing some defined measure of error (risk) with respect to some data (empirical evidence).

Given a model is a parameterized function $f_\theta(\mathbf{x})$ [2] that outputs some prediction for the datapoint $\mathbf{x}$. These predictions is typically of the form of (i) a probability distributions over some fixed set of classes (probability of certain objects being present in an input image [28]), (ii) numerical measure (price of stock in the future), or (iii) some abstract representation (vectors useful for comparing similarity of words [40]). Importantly, models are not strictly limited to these types of output.

The training procedure, i.e. finding a set of suitable model parameters θ, is then formalized as solving the following optimization problem for over some data set $S = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ [62]:

$$\underset{\theta \in \Theta}{\arg\min} \; \mathbb{E}_{(\mathbf{x},y)\sim S} \; l(f_\theta(\mathbf{x}), y) \; . \tag{3.1}$$

[1] Mitchell [41] defines Machine Learning as: *A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E."*

[2] The existence of non-parametric models is acknowledged (e.g. nearest neighbor), but these are considered out of scope due to their infrequent use in the application domain of this thesis.

Here, $l(f_\theta(\mathbf{x}), y) \in \mathbb{R}$ is a loss function that typically expresses the discrepancy of the output with (khe risk), respect to some desired output y, through a numerical value. Solving this optimization problem requires an algorithm, which is typically tied to the class of the model undergoing training. A popular choice of model class is multilayer perceptrons [51], frequently referred to as *neural networks*. This type of models typically use stochastic gradient descent [25, 49] (or variations thereof [26, 36]) to approximate the set of model parameters θ.

Following the training procedure, the trained model $f_\theta(\cdot)$ is evaluated on an disjoint and unseen test dataset. This evaluation seeks to capture the trained model's ability to perform on unseen data points, this performance is often denoted as the model's ability to *generalize*. As a motivating example, assume that we have trained a model for classifying objects in images. A typical evaluation would then include measuring the accuracy, percentage of correctly inferred objects across images from the test dataset. Given that a trained model demonstrates a desirable performance, it is typically applied to a problem domain using a computer system that has integrated the respective model and thus made its functionality available.

1 Security of Machine Learning

With machine learning models being integral components of certain computer systems, researchers have sought out to characterize properties of systems incorporating them [7]. Papernot [43] connected the CIA triad [44](confidentiality, integrity, and availability) from information security to the attack surface of machine learning systems. Linking *confidentiality* to information about the model (architecture, weights) or the data used for training. *Integrity* is linked to attacks that allow the adversary manipulate the output of the underlying model, e.g. perform small perturbations to input that yield unexpected and drastic changes to the output. While *availability* is associated with inconsistency and reliability of the machine learning model. Following this, Papernot [42] also connected the security design principles of Saltzer and Schroder [53] for computer systems to machine learning systems. Since then, a variety of security-related failure models of machine learning systems have been covered [30, 29]. One particular failure mode, with relevance to this thesis and implications beyond security applications, is *test-time attacks*.

2 Test-time Attacks

Test-time attacks are based on the phenomenon that an adversary might seek to violate the integrity of a machine learning model by perturbing a benign input, constrained by some threat model, such that the output of the model for the perturbed data point is altered to the adversary's desire [9, 57]. Threat models within this application domain needs to ensure that the perturbation does not alter the *true* semantic meaning of the input. Exemplifying this for computer vision, an adversary seeks to perturb a benign image input, such that the effect of perturbation remains imperceptible, while causing the classifier under attack to misclassify the perturbed image. In the field of computer vision, threat models (the set of allowed perturbations) are express as an ϵ-sized ℓ_p-norm ball $\Delta = \{\delta \mid \|\delta\|_p \leq \epsilon\}$. Typically, adversarial perturbations for an input-output pair $(\mathbf{x}, y)$ are found by solving the following optimization problem:

$$\arg\max_{\delta \in \Delta} l(f_\theta(\mathbf{x} + \delta), y) \tag{3.2}$$

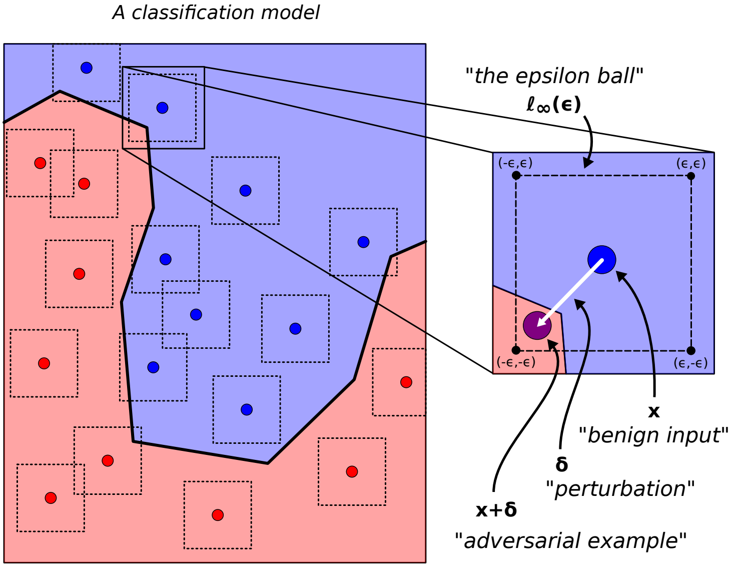


Fig. 3.1: Visualization of the decision space for a binary classification model for two classes of data points (red and blue). Each data point is surrounded by dashed squares that indicate an ℓ_∞-norm threat model of size ϵ, also referred to as an *epsilon ball*. Under the given threat model and a classification model, the objective of test-time attacks is to uncover some perturbation δ, such that the benign input $\mathbf{x}$ is misclassified (crosses the decision boundary).

Given that the loss function $l(\cdot, \cdot)$ express some numeric representation of error, adversarial perturbations can effectively be viewed as an approximation of the noise that maximizes the error during inference (i.e. violate training objective). Consider the motivating example in Figure 3.1, which illustrates an abstract classification model and two sets of data points (red and blue). It can be seen that for certain data points, the threat model $\ell_\infty(\epsilon)$ overlaps with the decision boundary of the opposing class, thereby indicating that there exists some perturbation that causes the given data point to be misclassified. In practical applications with highly-parameterized non-convex models, such as deep neural networks, providing similar proofs or guarantees through analysis of the output space is currently infeasible.

Thereby in order to measure a given model's robustness towards adversarial perturbations, a set of first-order attack algorithms [57, 14, 37] have been designed to approximate solutions to the optimization objective in Equation 3.2. Projected Gradient Descent [37] is considered state-of-the-art within the class of first-order algorithms [64].

The effect of adversarial perturbations, in the context of highly-parameterized non-convex neural networks, cause drops in accuracy by several orders of magnitude. Adversarial perturbations have demonstrated the ability to become robust to physical transformations, thus making them a potential threat to real-world systems [11, 6, 19]. A variety of defenses against these attacks have been proposed, however, only few defenses are actually improving adversarial robustness [4, 59]. An established method for improving adversarial robustness is *adversarial training*.

3 Adversarial Training

Wagner [63] describes adversarial training metaphorically as *vaccinating machine learning models*, and serves as a measure to improve robustness towards adversarial perturbations. Adversarial training is a training algorithm that attempts to solve the following robust optimization problem, which is an adaptation of Equation 3.1:

$$\underset{\theta \in \Theta}{\arg\min} \ \mathbb{E}_{(\mathbf{x},y)\sim S} \ \underset{\delta \in \Delta}{\arg\max} \ l(f_\theta(\mathbf{x}+\delta), y) \ . \tag{3.3}$$

This min-max formulation is typically solved by effectively attacking each batch of input-output pairs during training, using the described first-order methods [37, 64] prior to running common optimizers, such as SGD or ADAM [25, 49, 26]. Consequently, the models trained using this procedure demonstrates improved robustness towards adversarial perturbations [5, 16]. However, there remains a performance gap to the performance on benign input, and thus achieving adversarial robustness remains an open problem.

Part II

Papers

Paper A

Kraaler: A User-Perspective Web Crawler

Thomas Kobber Panum, René Rydhof Hansen, Jens Myrup Pedersen

In proceedings of the
3rd Network Traffic Measurement and Analysis Conference (TMA).

Abstract. *Adaption of technologies being used on the web is changing frequently, requiring applications that interact with the web to continuously change their ability to parse it. This has led most web crawlers to either inherent simplistic parsing capabilities, differentiating from web browsers, or use a web browser with high-level interactions that restricts observable information. We introduce Kraaler, an open source universal web crawler that uses the Chrome Debugging Protocol, enabling the use of the Blink browser engine for parsing, while obtaining protocol-level information. The crawler stores information in a database and on the file system and the implementation has been evaluated in a predictable environment to ensure correctness in the collected data. Additionally, it has been evaluated in a real-world scenario, demonstrating the impact of the parsing capabilities for data collection.*

© 2019 International Federation for Information Processing (IFIP).
The layout has been revised.

A.1 Introduction

Information on the web has reached magnitudes and change at a rate which are infeasible for humans to structure and manage. This has motivated the use of machines for automated solutions capable of interpreting content on the web and represent the interpretation in a structured form. These types of solutions are referred to as web crawlers and consist of applications that systematically visit web servers and parse their content [A15]. Web crawlers have been an active research topic within recent decades and were initially motivated by the need for making the web searchable [A11]. However, they are now being used in a wider variety of applications, such as gathering data sets for statistical modeling [A24, A19].

The typical retrieval process of web crawlers involves exploiting the interlinked structure of Hypertext Markup Language (HTML) documents being collected, by parsing the collected documents for hyperlinks to other documents and consider the found links as documents to be collected. This process stems from a time when web pages included fewer content types and relied less on externally linked dependencies.

However, the number of technologies and content types used on the web has increased drastically over recent decades. This has led to the end user client, the web browser, to become a complex application and include parsers for a variety of content types. The size of the code bases of the three most widely adopted parsers among web browsers (Chromium Blink [A8], WebKit [A13], Mozilla Gecko [A17]) highlight this complexity, and are respectively: 2.9M, 13.9M, and 20.7M lines of code, as of April 2019. This complexity leaves most web crawlers unable to implement parsing capabilities to the same extent, causing a deficit in method web applications are being crawled compared to the user's perspective. This deficit stems from the fact that existing web crawlers do not parse the crawled content to the same extent as a typical browser engine would. Examples of crawlers using web browsers have been seen, but their interaction with the web browsers leaves them unable to access detailed information about the underlying request behavior, e.g. usage of the HTTP protocol and subsequent requests. This leads to information being lost or unavailable for analysis and is a continuous problem as the capabilities of web browsers change over time.

In order to be able to enhance information gained throughout a crawling process, and have it represent the user's perspective, we introduce the universal web crawler, Kraaler. It is a user-perspective web crawler that uses the browser engine of Google Chrome (Chrome), Blink [A8], through the use of the Chrome Debugging Protocol (CDP) while obtaining information about parsing and HTTP usage. The contributions of our work can be summarized as:

- Demonstrate a new method for user-perspective web crawling while observing detailed protocol information.

- Present a data structure containing this information while making it efficiently available for behavioral analysis of web applications, such as phishing detection.

- Provide an open-source and extendable implementation.

The implemented crawler is evaluated by exposing it to a predictable environment and a real-world environment, respectively. Each environment provides the ability to validate the correctness of the obtained data and demonstrate the impact of the web browser's parser in a crawling setting. Lastly, examples of the applicability of obtained data for behavioral analysis of web applications is shown.

A.2 Related Work

Information hosted on web servers is accessed through the Hypertext Transfer Protocol (HTTP) or its encrypted variant Hypertext Transfer Protocol Secure (HTTPS). End users typically use these protocols on a high-level through their own client, the web browser. Machine-based approaches, in the form of web crawlers, tend to typically interact with the protocol directly. Browsers abstract from the numerous underlying requests being sent back and forth between the browser and the web server, when a user interacts with a web page. The order, and to which extent these requests are being sent, are determined by the browser's parsing component, the browser engine.

Browser engines contained within web browsers impact and define the capabilities of the applications hosted on the internet. They thereby serve both as a delimiter and enabler of technologies used for web applications and affect their respective usage. Namely, the programming language JavaScript was designed to be used for web applications and is now considered one of the most widely adopted programming languages [A22, A7].

Two common web browsers, Chrome and Mozilla Firefox (with respective market shares of $\approx 70\%$ and $\approx 10\%$ [A23]), are using their own respective browser engines: Blink (a fork of WebKit) and Gecko. These engines are able to interpret and display numerous types of content, markup languages, and programming languages. This ability has developed over time and continues to do so, as the desire for richer web applications keeps persisting.

Web crawlers are applications for systematically gathering information from web servers, and have been an active research topic for decades. The research was initiated by the Mercator web crawler, which went on to become the web crawler of the Alta Vista search engine [A11]. Following this, Kumar et al. has surveyed existing work and proposes a five type categorization of web crawlers: Preferential crawlers, hidden web crawlers, mobile crawlers, continuous crawlers, and universal crawlers [A15].

Preferential crawlers are conditioning their crawling behavior, such as restricting only gathering from a subset of sources or only gather selective information.

Hidden (or sometimes referred to as *Deep*) web crawlers focus on crawling information that cannot be obtained by just following hyperlinks. Examples of this are web applications that use dynamic input for presenting results, such as search engines that require a search query in order to present a list of results. CrawlJax is a hidden web crawler capable of crawling web applications that rely on JavaScript for creating user interactions [A6]. In order to crawl such web applications, PhantomJS is used for instrumenting a web browser to programmatically perform user actions within the browser [A1].

Mobile web crawlers constitute a subset of crawlers that use an experimental method of crawling proposed by [A10]. This method seeks out to avoid the iterative crawling behavior, of obtaining information across multiple requests for one source, by expecting remote web servers to be able to receive a data specification. The received data specification is then used to initiate a stream of actions locally on the remote server, in order to reduce bandwidth usage in the crawling process.

Continuous crawlers constitute a subset of crawlers that addresses the problem of prioritizing crawling targets, in the setting of restricted resources and crawling targets continuously changing their content.

Universal crawlers are the counterpart to the previously described crawlers, as they are designed for a broader and less specific use case. They are designed to visit any type of content or hyperlink they observe and repeat this process continuously. An example of a universal web crawler is BUbiNG, an open source high-performance distributed web crawler that performs HTML parsing and can store results on distributed storage systems [A3]. Apache Nutch is another universal web crawler that has a modular design and has been actively developed for more than a decade. The modular design of Apache Nutch has led researchers to use it as a base, and extend it to new types of web crawlers [A9].

A subset of web crawlers set out to extract exact information in a structured manner from a web application, this method is known as *web scraping*. Scrapy is a widely popular web scraping framework developed in the programming language Python [A18]. The framework requires the user to be familiar with Python in order to specify which information to extract and the following method. An alternative solution is Portia, a web scraping framework built on Scrapy requiring no competences in programming [A21]. The user clicks on information on a web page, and the framework attempts to extrapolate the underlying HTML template, defining a strategy for crawling the desired information. This extrapolation can, however, lead to incorrect identification of the underlying HTML structure, leading to incorrect or false information being extracted.

Kraaler is a universal web crawler that uses the parser of a web browser for interpreting web applications, allowing it to observe HTTP requests initiated by the HTML parser, the JavaScript interpreter, and more. A similar abstract design has been patented by Kraft et al., which describes the use a web browser for gathering dynamic content of web applications and utilize for optical character recognition for interpreting text in images [A14]. However, certain design details are undocumented and, to our knowledge, there exists no open source implementation of the described design nor a specification of the data it collects.

A.3 User-Perspective Crawling

HTTP is a protocol based upon requests that are answered by responses, and when a user enters a URL in their browser, a request of the method GET with a couple of key-value pairs in the header is sent towards the host specified in the URL. Most of the existing universal crawlers base their crawling behavior on this, so given a set of URLs, the crawler will perform GET requests towards a given URL and retrieve the body of the

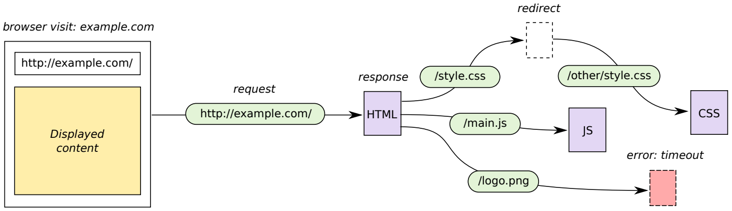


Fig. A.1: Series of subsequent requests being performed by a web browser, when visiting a page.

corresponding HTTP response. This response body, which is often assumed to contain HTML, is typically analyzed for the presence of hyperlinks or URLs that are then added to the pool of known URLs.

Modern web applications consist of multiple content types, spread across multiple URLs referenced to by an HTML document, acting as dependencies for the web application. The browser is expected to visit the HTML document first, and then a process of parsing it starts, leading the browser to perform asynchronous requests for resources that the document references. The parsing process repeats for every response that is received by the browser, creating a recursive parsing process, causing the browser to perform a series of subsequent requests from the initial request of the document.

This recursive request behavior can be seen as an interlinked dependency graph of distributed resources, rooting at an initial document, as illustrated in Figure A.1. The initial document is typically an HTML document and yields the browser to create a Document Object Model (DOM), which is an application programming interface for client-side programming languages to manipulate the received HTML and display the parsed model to the user [A12]. Concretely, this allows modern web applications to use JavaScript to manipulate the initial HTML tree of the received document. This enables additional methods of navigating and updating the information of web applications without the user having to navigate to other HTML documents [A6]. This is accomplished by allowing JavaScript to programmatically perform asynchronous HTTP requests, Asynchronous JavaScript and XML (AJAX), in order to either send or retrieve information following manipulation of the DOM [A2].

These features, in addition to other capabilities of modern browsers not described in this article, have made it increasingly difficult to ensure the features are only available in contexts that meet a minimum security level [A25]. Following this, the specification of Secure Contexts was introduced, which is a method for controlling the security level of actions by the browser on behalf of the web application's content. This means that certain aspects of the actions performed by the browser are conducted in an isolated sandbox environment, and certain actions can be restricted. As an example, a document served over a secure and encrypted connection is not allowed to reference other resources served using an insecure and unencrypted protocol. Parsing content of the web applications is an ever-changing process that evolves over time, causing modern browsers and their underlying browser engines to become applications spanning millions of lines of code. The implementation of new browsers engines for parsing content is, therefore, a costly and often infeasible process in the design of a crawler.

Historically, this left designers with the choice of either partial parsing capabilities, or to use high-level instrumentation of a web browser. High-level instrumentation of a web browser typically involves using a web driver, such as Selenium Webdriver, for programmatically controlling a subset of user-based features available in the browser [A20]. These features typically include navigating the browser to a certain URL or interacting with the JavaScript shell.

An alternative to the Selenium solution is PhantomJS, which is a headless browser allowing for more information to be extracted from the underlying browser engine [A1, A6]. In comparison, it allows for extracting request and response information from HTTP. The project has, however, been suspended, leaving the underlying engine to become obsolete from modern standards of web browsers.

In the context of Kraaler, the web browser Chrome is used as an external component for performing the HTTP requests and parsing of their respective responses during crawling. This choice stems from the fact that Chrome provides a remotely accessible application interface to its underlying browser engine, Blink, denoted Chrome Debugging Protocol (CDP) [A5]. CDP allows for reading some of the data structures present in the browser engine during runtime in a structured form, such as detailed information about requests and responses, and sending instructions to the browser. Thus enabling the use of the complex parser contained within the browser engine without missing information contained within the HTTP protocol and other types of low-level information contained within the engine, which is difficult or infeasible in other widely-used browsers.

Information in CDP is exchanged through a series of subject-based channels, e.g. the channel of networking is named `Network`, and subscribing to the same channel allows for receiving events being published on the channel. Throughout this article, events published by CDP will be following the notation of `<channel>.<event>`, so in the case of `Network.requestWillBeSent`, `Network` refers to the channel which sends `requestWillBeSent` events. These events are published to their channel as JSON objects containing information related to the event, e.g. `Network.requestWillBeSent` contains information about a request that Chrome is about to initiate. The entirety of events, across of all channels, can be seen as a structured log of the captured behavior in Chrome. Instructions pushed to channels follow the same notation, e.g. `Page.navigate` will navigate the active window to a certain URL contained in the payload. CDP defines the concept of a page, describing the situation of when the browser navigates to a URL using the address bar. As previously described, this will perform an HTTP request, for which the response is parsed and can lead to subsequent requests being executed as a background activity. Kraaler inherits the concept of page, as illustrated in Figure A.1, and defines a page to include: a series of HTTP requests and responses, the appearance of the web application in the browser, the JavaScript shell of the given web application, and other information described in Section A.5. In addition to the concept of pages, the concept of action is introduced to group a request with its respective response, or the connection error returned by the browser, when trying to perform the request.

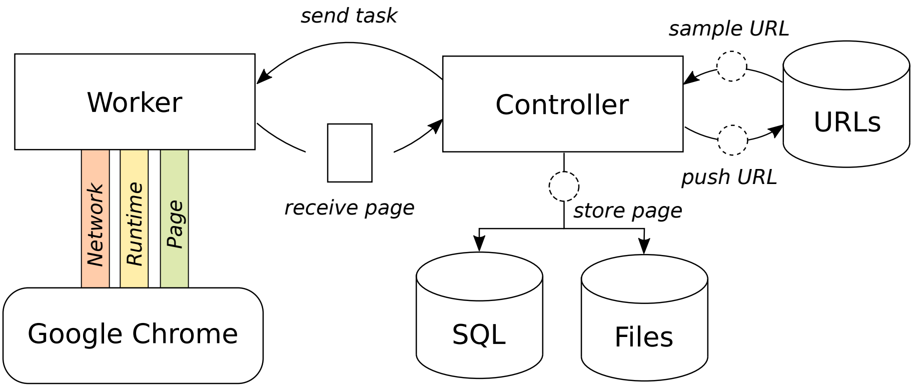


Fig. A.2: Component overview of Kraaler.

A.4 Overview

Kraaler has been implemented in the programming language of Go. Go was chosen due to its native support for parallelization primitives and its ability to compile statically-linked binaries for multiple platforms. The code is open-source with a GNU GPLv3 license and accessible in a git repository, hosted on GitHub[1]. Running the implementation depends on Chrome or Docker being available in order to either use or install Chrome.

Internally, Kraaler consists of two components, controller and worker, that are responsible for interacting and orchestrating a set of external components, as illustrated in Figure A.2. Controller is responsible for communicating and orchestrating the parallel crawling process, conducted by a pool of workers while publishing their results to the external data stores. Each worker is responsible for conducting crawling tasks and orchestrate their individual instance of Chrome.

Controller

The primary role of the controller is to maintain the parallel crawling process. The controller will continuously push tasks to the next available worker in the pool and keep the set of workers occupied with tasks until no new task is available. Tasks become available as a sampler continuously samples URLs from a set of known URLs. Initiating a crawling process thereby requires the set of known URLs to be populated with some amount of URLs. Populating this set is done by parsing a set of domains or URLs to the controller, as it is being instantiated.

A sampler is a module for containing the mechanism for prioritizing which, and when, known URLs should be transformed to tasks. In order for a sampler to conduct this prioritization, it is exposed to the current set of known URLs and timestamp of their most recent crawl, at the time of sampling. Under these conditions, users of Kraaler are capable of implementing new samplers that encapsulate a prioritization

[1] https://github.com/aau-network-security/kraaler

strategy suitable for their respective use cases. Currently, there are two samplers implemented in Kraaler, a uniform sampler, and a domain-pair sampler. As tasks are being completed, pages are returned from the pool of workers to the controller, and it starts two actions: storing the information of the page and pushing newly found URLs to the pool of known URLs.

A page's textual and numeric properties are by default stored in a relational database, while the byte-based properties are stored on the file system in a structured form and referenced in the database. Schemes and data structures of the relational database are covered in more detail in Section~A.5. The byte-based properties include response bodies and screenshots. For certain applications, storing all byte-based information can be too extensive and unnecessary. To address this, Kraaler allows for restricting information stored, through a set of modules that modify a page received from a worker before being stored. Currently, Kraaler only has one module, that deletes response bodies of a certain content type for a page's actions. This leads to the functionality of restricting byte-based information contained in the following data, as empty response bodies are not stored. An example of a use case for this module is to restrict response bodies stored to only be text-based by deleting response bodies for which their content type is not prefixed with `text`. Additional modules can be implemented by users of Kraaler, for further controlling and restricting information being stored in the resulting data set of a crawl.

For obtaining a continuous crawling process, URLs found in the response bodies are by default added to the pool of known URLs. However, if it is desired to have a discrete crawling process or filter certain URLs, it is possible to control which URLs are added to the pool by assigning a URL filter.

Worker

The worker component is responsible for controlling the life cycle of a single Chrome instance, while also interacting with its instance through CDP. Problems involved in controlling the life cycle of Chrome is covered in more detail in Section A.6. The interaction spans across three CDP channels: `Page`, `Runtime`, and `Network`.

Using the `Page` channel, a worker will use `Page.navigate` for sending instructions to navigate the browser window to a specific URL. The worker will use the event `Page.domContentEventFired` for determining when the initial document has been parsed and loaded. When this event has been received, the worker attempts to capture one or more screenshots through the instruction `Page.captureScreenshot`, for which each capture is conducted a configurable amount of seconds after the DOM is loaded.

The `Runtime` channel is used for the event `Runtime.consoleAPICalled` in order to capture console output from the JavaScript shell of the given page, e.g. debugging messages used by web developers.

Most of the information obtained by Kraaler is from the `Network` channel, which covers a variety of network events. `Network.requestWillBeSent` is an event being published when a page is about to send a HTTP request. The event covers a variety of information, but Kraaler saves the HTTP request included in the event and the initiator of the given request. Initiators defined by CDP are *parser*, *script*, *preload*, *SignedExchange*, and *other*. Kraaler inherits and expands upon this definition by introducing additional initiators, determined by request information, namely: *redirect* and *user*. The *redirect* initiator stems from the fact that Kraaler expands a chain of HTTP redirects to become a series of individual requests, while CDP assumes a request is tied to a response body. This means that if a requested resource is located on a certain URL, it will respond with one or more redirects before receiving a response body, then CDP inherit these into the same request, except that by HTTP it is a series of requests. Introduction of the *user* initiator was done to improve clarity of the generic initiator *other*, by marking user-initiated requests, such as the initial HTTP request performed by a page navigating to an URL. For gathering the response of a request, the `Network.responseReceived` event is published, containing the HTTP response for the request. The `Network.responseReceived` event just include the metadata of the response. In order to obtain the response body, the instruction `Network.getResponseBody` is sent.

The events received across these channels are observed between the initial `Page.navigate` instruction and after the `Page.captureScreenshot` instructions has completed. Lastly, the information from these events are saved in the internal data structure of a page, being the result of a crawl.

A.5 Data Structure

Creating value from a crawling process requires information related to the crawled content to be available for post-processing. In Kraaler, this is conducted by having the controller store the pages it receives in a relational database and on the file system. The choice of relational database in Kraaler is SQLite, which provides efficient reads while the data can be contained in a single file. With data available in a single file, it eases data transfers to other computation environments.

Having two types of data stores allows for separating byte-intensive information, such as screenshots and response bodies, from contextual information. Storing of screenshots in Kraaler is structured in a configurable directory structure using `screenshots/<domain>/<id>.png`, for which `domain` is the domain of hosting the page visualized and `id` being a random unique identifier. Response bodies are stored in a separate directory with the structure of `bodies/<hash>.<ext>`, with `hash` being the SHA256 hash of the body and `ext` being a determined file extension based on the content type of the body. Storing every response body can yield to a substantial amount of data, causing data sets to become overwhelmingly large in size. Kraaler provides a set of modules that can filter or change the behavior of storing response bodies, namely a compression module and a filter module. The compression module allows for using gzip to compress every stored response body to reduce the data set size. Additional measures to reduce the size of the response bodies include using the filter module to reduce the set of saved response bodies to only a certain set of content types.

Contextual information being stored in the database can be seen in Table A.1. Entities are shown using the following notation:

Some entity $\leftarrow 0..*$ *Parent entity*

In this example the notation reads as: *Parent entity* has zero or more ($0..*$) *Some entity*. In order to make the underlying database efficient for analysis purposes, a set of database design principles from online analytical processing (OLAP) has been adapted in the design of the database scheme [A4]. In Kraaler, the OLAP snow-flake scheme is used and provides efficient storage in terms of the size used by the database and reading speeds for database operations.

A.6 Orchestration

Communication with external components, namely Chrome, is a fundamental part of the crawling process of Kraaler. The CDP service of Chrome is a central dependency that would interrupt the crawling process, in case of its absence. Ensuring the availability of the service is inherently difficult, as its presence can only be observed from an operating system perspective or by interacting with it.

Experience gained from implementing Kraaler made it clear that the service could become unavailable or unusable. This led to a set of scenarios, that were difficult to differentiate from external observations, such as: external web servers being unresponsive, web servers replying slowly, or the instance of Chrome being in an unexpected state.

These scenarios could lead to a worker becoming unable to continue crawling, and drove the design of increased fault tolerance. Measures for increasing the fault tolerance, and included in the orchestration of external components, are: adding timeouts for crawls, release of unresponsive resources, isolation of Chrome instance and designing for errors.

Timeouts are a time-based threshold measure for defining and reacting to unexpected behavior in an application. They are typically defined by having a time limit that describes the maximum allowed time a certain process is allowed to be spending for processing. For Kraaler this mechanism is used internally within each worker, timing the process responsible for sending instructions and receiving feedback from CDP. In case of a timeout, a page with an internal connection timeout error is returned to the controller.

Following the case of a timeout, the resources of the asynchronous process, that was unable to complete on time, is not guaranteed to be freed and can be locking resources. In Kraaler, we explicitly ensure to inherit an idiomatic design pattern for solving this, while also restarting the instance of Chrome. This restart is done to ensure that the instance of Chrome is cleared from its potential faulty state, by returning it to the predictable initial state.

Our recommended method of running the external Chrome instances is by letting Kraaler communicate with the Docker daemon. This method allows for spawning Chrome instances in an isolated run time scope while controlling the resources available to them. The isolation ensures that the Chrome instances are unable to interfere with each other, as their run time scope is independent. Kraaler will by default con-

Table A.1: Properties of data structure

	Property	Description
Page	Browser resolution	Resolution used by the browser
	Navigated time	Unix time nanoseconds of when a page is request
	Loaded time	Unix time nanoseconds of when a page's DOM is loaded
	Terminated time	Unix time nanoseconds of when a crawl of page is complete
	Page connection error	Possible connection error of page
Console output ← 0..∗ Page	Sequence	Index of the `console.log` message for the page
	Origin	Position of JavaScript call using `console.log`
	Message	Message printed by `console.log`
Screenshot ← 1..∗ Page	Time taken	Unix time nanoseconds of when screenshot was captured
	Path	Filesystem path to screenshot file
Action ← 0..∗ Page	Parent of action	Possible previous action causing this action
	Request method	Method used for HTTP request of action
	Protocol	Protocol used for the action
	Initiator	The type of initiator initating the action
	Status code	Status code of the action's HTTP response
	Connection error	Possible connection error of the action
Host ← 1 Action	Apex domain	Domain without subdomains
	Top-level domain	Top-level domain
	IPv4	IPv4 Address of domain being resolved
	Name servers	Authoritative name servers of domain
URL ← 0..∗ Action	Scheme	Scheme used within the URL
	User	User field of URL
	Host	Host contained within the URL
	Path	Path of URL
	Fragment	Fragment used in URL
	Query	Query string of URL
Response Header ← 0..∗ Action	Key	Key field of response header
	Value	Value field of response header
Request Header ← 0..∗ Action	Key	Key field of request header
	Value	Value field of request header
Security Details ← 0..1 Action	Secure protocol	Secure protocol used by the given action
	Certificate key exchange	Type of key exchange used by action
	Certificate issuer	Issuer of the certificate used in the action
	Certificate cipher	Certificate cipher used for action
	Certificate san list	San list of certificate used by action
	Certificate subject name	Subject name of certificate used for action
	Certificate valid from	Unix time nanoseconds of validation start of cert. for action
	Certificate valid to	Unix time nanoseconds of validation end of cert. for action
Response Body ← 0..1 Action	Browser MIME type	MIME type of body, determined by the browser
	Worker MIME type	MIME type of body, determined by the worker
	Hash	SHA256 hash of the response body
	Size	Size, in bytes, of response body
	Compressed size	Gzip compressed response body size in bytes
	Path	Path to file containing response body

strain the spawned Chrome instances to 756MB, which from experience has been found sufficient for single page browsing.

Restarting the Chrome instance of a worker can be costly in terms of wasted crawling time, and should only be considered a last resort. In order to reduce the number of unnecessary restarts of the Chrome instance, a set of errors returned by CDP are provided corresponding recover mechanisms. This allows workers to cheaply recover from errors such as CDP connections timeouts, `Page.navigate` timeouts, and more. However, in the case of an error without a defined recover mechanism, the Chrome instance is restarted to ensure no present side effects.

A.7 Evaluation

Correctness is an essential attribute for a data set in order to be used for future analyses. In the setting of Kraaler, the data is collected by observing external web servers, for which their underlying design and behavior is unknown. Thereby in order to increase confidence in collected data being correctly crawled and stored, this functionality is evaluated against known web servers, for the sake of predictability. A set of automated end-to-end tests are designed, for which each individual test hosts a web server with distinct behavior. The hosted web servers are then crawled by Kraaler, following observations of changes in the database and file system. These changes are then compared against an expected change, to ensure the expected behavior of the implementation.

The set of test cases does, however, not validate the value of the browser engine's parsing capabilities in a real-world setting, as the tested web application might not be representative of that population. In order to validate the value, Kraaler was set out to crawl a uniform sample of Alexa Top 1M, while being restricted from pushing new URLs to the pool of URLs. This crawl was conducted on a single machine running eight worker instances and resulted in 8156 pages and 331561 actions. The influence of the browser engine's parsing capabilities, in terms of HTTP requests being initiated by the browser engine, can be seen in Table A.2. User-oriented initiations, i.e. those started by `Page.navigate`, are filtered out to focus only on the ones conducted by the browser engine.

Table A.2: Overview of non-user initiators of actions from Alexa Top 1M page crawls.

Initiator	Page actions	Body size (kB)
	μ (σ)	μ (σ)
parser ($n = 266819$)	34.34 (25.31)	41.04 (125.77)
script ($n = 46131$)	5.94 (6.39)	46.77 (96.53)
redirect ($n = 4512$)	0.58 (0.96)	107.00 (161.40)
other ($n = 2399$)	0.31 (1.62)	24.96 (83.15)

In comparison to a naive crawler, one which just fetches a given response body without parsing it, a substantial amount of requests would not be conducted. The initiators, *parser* ($\mu = 34.34$) and *script* ($\mu = 5.94$), account for a significant amount of additional requests compared to a naive crawler. These initiated requests are a representation of the sum of the parsing capabilities of the browser engine, and can be expressed as an upper bound measure for requests initiated by parsing for user-perspective web crawling. The total size of the response bodies for these subsequent HTTP requests represent 95.8% of the total amount of bytes for response bodies in the crawl, leaving the root documents to 4.2%.

Browsing the size of subsequent response bodies of pages, in relation to their content type, can indicate the amount of information being carried by certain technologies. The amount of information being carried by certain types of technologies can suggest the importance of certain language parsers in a crawler setting. This information, carried across pages, for the ten most frequently used MIME types across pages, is illustrated using a cumulative distribution function in Figure A.3. JavaScript with its three MIME types (application/x-javascript, application/javascript, text/javascript) is responsible for a large degree of the bytes across the pages crawled. We suspect it might be due to popular domains using complex user interfaces and relies on JavaScript-based front-end frameworks, which take up a certain byte volume.

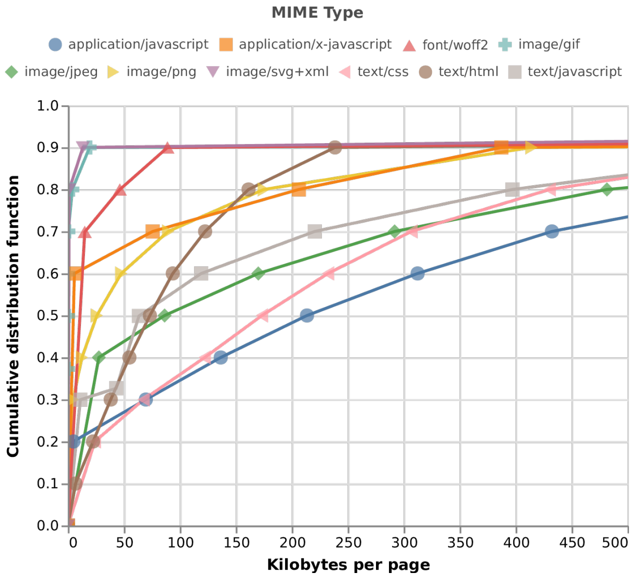


Fig. A.3: Cumulative distribution function of bytes per page for the ten most frequent content types.

The usage of technologies, in terms of request frequency and byte volume, vary significantly, e.g. *parser* actions $\sigma = 25.31$, and *parser* bytes $\sigma = 125.77$. This might suggest that pages rely on a distinct set of linked dependencies, due to the present variance in amount actions and their respective byte size for pages. This variance could also prove useful for identification of segments that carry meaning for a given problem domain that attempts to classify web applications. As an example, identification of web applications hosting malicious activities could be accomplished under the assumption that malicious web applications differentiate in their interlinked structure and usage of certain dependencies. However, this example and other applicabilities of the data set are for future research to examine, in addition to exploring the protocol-based information also contained within the data set.

A.8 Conclusion

In this article, we have presented the design of a crawler, named Kraaler, that uses the Chrome Debugging Protocol for parallelized crawling, using a modern browser engine, while obtaining detailed information from the HTTP protocol usage. Follow this, a design for storing this detailed information was covered, allowing the stored data efficiently read and available for data analysis.

The challenges of interacting with external components, such as a web browser, was presented in addition to the measures we have taken in order to solve them and provide orchestration. Throughout the implementation of our solution, the methods used for determining its correctness has been described.

Following the impact of the obtained parsing capabilities, from the browser engine, for the request frequency and size of the information collected throughout a crawling process. Demonstrating that the HTML parser ($\mu = 34.34$) and the JavaScript interpreter ($\mu = 5.94$) accounted for a significant amount subsequent HTTP requests for a page visit, for a subset of crawled Alexa Top 1M domains. The size of these response bodies, in terms of bytes, accounted for 95.8% of the total size of response bodies that was crawled.

We suggest that data sets collected using Kraaler could potentially be used for a variety of applications that seek to conduct statistical analysis of web applications.

A.9 Acknowledgements

With the Chrome Debugging Protocol being essential for this work, our gratitude towards the Chromium Developer Team cannot be understated. A great appreciation should be sent to Mathias Frederiksson, for his wonderful CDP Go package. I (Thomas) would like to thank my first-born daughter, Joanna, for the ability to be born in a timely fashion to cheer for her dad throughout his long working days, required to complete this article. Likewise, her caring mother, Maria, for indulgently accept my absence.

References

[A1] Ariya Hidayat. *PhantomJS - Scriptable Headless Browser*. http://phantomjs.org/. 2010.

[A2] Julian Aubourg et al. *XMLHttpRequest Level 1*. WD not longer in development. https://www.w3.org/TR/2016/NOTE-XMLHttpRequest-20161006/. W3C, Oct. 2016.

[A3] Paolo Boldi et al. "BUbiNG". In: *Proceedings of the 23rd International Conference on World Wide Web - WWW '14 Companion*. 2014, nil. DOI: 10.1145/2567948.2577304. URL: https://doi.org/10.1145/2567948.2577304.

[A4] Surajit Chaudhuri and Umeshwar Dayal. "An Overview of Data Warehousing and OLAP Technology". In: *SIGMOD Rec.* 26.1 (Mar. 1997), pp. 65–74. ISSN: 0163-5808. DOI: 10.1145/248603.248616. URL: http://doi.acm.org/10.1145/248603.248616.

[A5] *Chrome Debugging Protocol*. https://chromedevtools.github.io/devtools-protocol/. [Online; accessed 28-March-2019]. 2019.

[A6] Arie van Deursen, Ali Mesbah, and Alex Nederlof. "Crawl-Based Analysis of Web Applications: Prospects and Challenges". In: *Science of Computer Programming* 97.nil (2015), pp. 173–180. DOI: 10.1016/j.scico.2014.09.005. URL: https://doi.org/10.1016/j.scico.2014.09.005.

[A7] GitHub. *GitHub Octoverse*. https://octoverse.github.com/projects Last accessed on 2019-13-04. 2018.

[A8] Google. *Chromium Blink*. https://chromium.googlesource.com/chromium/blink. 2013.

[A9] Clement de Groc. "Babouk: Focused Web Crawling for Corpus Compilation and Automatic Terminology Extraction". In: *2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*. Aug. 2011, nil. DOI: 10.1109/wi-iat.2011.253. URL: https://doi.org/10.1109/wi-iat.2011.253.

[A10] Joachim Hammer and Jan Fiedler. "Using Mobile Crawlers to Search the Web Efficiently". In: *International Journal of Computer and Information Science* 1 (2000), pp. 36–58.

[A11] Allan Heydon and Marc Najork. "Mercator: A scalable, extensible Web crawler". In: *World Wide Web* 2.4 (1999), pp. 219–229. DOI: 10.1023/a:1019213109274. URL: https://doi.org/10.1023/a:1019213109274.

[A12] Ian Jacobs et al. *Document Object Model (DOM) Level 1*. W3C Recommendation. http://www.w3.org/TR/1998/REC-DOM-Level-1-19981001/. W3C, Oct. 1998.

[A13] KDE. *WebKit*. https://svn.webkit.org/repository/webkit/. 1998.

[A14] Reiner Kraft and Jussi P. Myllymaki. "System and method for enhanced browser-based web crawling". US7519902B1. 2000.

[A15] Manish Kumar, Rajesh Bhatia, and Dhavleesh Rattan. “A Survey of Web Crawlers for Information Retrieval”. In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* (2017). DOI: `10.1002/widm.1218`. URL: `https://doi.org/10.1002/widm.1218`.

[A16] Elmer EH Lastdrager. “Achieving a Consensual Definition of Phishing Based on a Systematic Review of the Literature”. In: *Crime Science* 3.1 (2014), p. 9. DOI: `10.1186/s40163-014-0009-y`. URL: `https://doi.org/10.1186/s40163-014-0009-y`.

[A17] Mozilla. *Mozilla Gecko*. https://hg.mozilla.org/mozilla-central/. 1998.

[A18] Mydeco. *Scrapy: a fast high-level web crawling & scraping framework for Python*. https://scrapy.org/ Last accessed on 2019-13-04. 2008.

[A19] Alec Radford et al. “Language Models are Unsupervised Multitask Learners”. In: (2019).

[A20] Sagar Shivaji Salunke. *Selenium Webdriver in Java: Learn With Examples*. 1st. USA: CreateSpace Independent Publishing Platform, 2014. ISBN: 1495450201, 9781495450204.

[A21] Scrapinghub. *Portia: Visual scraping for Scrapy*. https://github.com/scrapinghub/portia Last accessed on 2019-13-04. 2014.

[A22] StackOverflow. *Developer Survey Results 2018*. https://insights.stackoverflow.com/survey/2018 Last accessed on 2019-13-04. 2018.

[A23] Statista. *Global market share held by leading desktop internet browsers from January 2015 to December 2018*. https://www.statista.com/statistics/544400/market-share-of-internet-browsers-desktop/ Last accessed on 2019-13-04. 2018.

[A24] Marco Vieira, Nuno Antunes, and Henrique Madeira. “Using web security scanners to detect vulnerabilities in web services”. In: *2009 IEEE/IFIP International Conference on Dependable Systems & Networks*. June 2009, nil. DOI: `10.1109/dsn.2009.5270294`. URL: `https://doi.org/10.1109/dsn.2009.5270294`.

[A25] Mike West. *Secure Contexts*. Candidate Recommendation. https://www.w3.org/TR/2016/CR-secure-contexts-20160915/. W3C, Sept. 2016.

Paper B

CAUSE: Learning Granger Causality from Event Sequences using Attribution Methods

Wei Zhang, Thomas Kobber Panum, Somesh Jha, Prasad Chalasani, David Page

In proceedings of the
37th International Conference on Machine Learning (ICML 2020).

Abstract. *We study the problem of learning Granger causality between event types from asynchronous, interdependent, multi-type event sequences. Existing work suffers from either limited model flexibility or poor model explainability and thus fails to uncover Granger causality across a wide variety of event sequences with diverse event interdependency. To address these weaknesses, we propose CAUSE (Causality from AttribUtions on Sequence of Events), a novel framework for the studied task. The key idea of CAUSE is to first implicitly capture the underlying event interdependency by fitting a neural point process, and then extract from the process a Granger causality statistic using an axiomatic attribution method. Across multiple datasets riddled with diverse event interdependency, we demonstrate that CAUSE achieves superior performance on correctly inferring the inter-type Granger causality over a range of state-of-the-art methods.*

The layout has been revised.

B.1 Introduction

Many real-world processes generate a massive number of asynchronous and interdependent events in real time. Examples include the diagnosis and drug prescription histories of patients in electronic health records, the posting and responding behaviors of users on social media, and the purchase and selling orders executed by traders in stock markets, among others. Such data can be generally viewed as *multi-type event sequences*, in which each event consists of both a timestamp and a type label, indicating when and what the event is, respectively.

In this paper, we focus on the fundamental problem of uncovering causal structure among event types from multi-type event sequence data. Since the question of "true causality" is deeply philosophical [B29], and there are still massive debates on its definition [B28, B23], we consider a weaker notion of causality based on predictability—Granger causality. While Granger causality was initially used for studying the dependence structure for multivariate time series [B17, B10], it has also been extended to multi-type event sequences [B12]. Intuitively, for event sequence data, an event type is said to be (strongly) *Granger causal* for another event type if the inclusion of historical events of the former type leads to better predictions of future events of the latter type.

Due to their asynchronous nature, in the literature, multi-type event sequences are often modeled by multivariate point process (MPP), a class of stochastic processes that characterize the random generation of points on the real line. Existing point process models for inferring inter-type Granger causality from multi-type event sequences appear to be limited to a particular case of MPPs—Hawkes process [B16, B37, B1], which assumes past events can only independently and additively *excite* the occurrence of future events according to a collection of pairwise kernel functions. While these Hawkes process-based models are very interpretable and many include favorable statistical properties, the strong parametric assumptions inherent in Hawkes processes render these models unsuitable for many real-world event sequences with potentially abundant *inhibitive* effects or event *interactions*. For example, maintenance events should reduce the chances of a system breaking down, and a patient who takes multiple medicines at the same time is more likely to experience unexpected adverse events.

Regarding event sequence modeling in general, a new class of MPPs, loosely referred to as neural point processes (NPPs), has recently emerged in the literature [B14, B36, B27, B35]. NPPs use deep (mostly recurrent) neural networks to capture complex event dependencies, and thus excel at predicting future events due to their model flexibility. However, NPPs are uninterpretable and unable to provide insight into the Granger causality between event types.

To address this tension between model explainability and model flexibility in existing point process models, we propose CAUSE (<u>C</u>ausality from <u>A</u>ttrib<u>U</u>tions on <u>S</u>equence of <u>E</u>vents), a framework for obtaining Granger causality from multi-type event sequences using information captured by a highly predictive NPP model. At the core of CAUSE are two steps: first, it trains a flexible NPP model to capture the complex event interdependency, then it computes a novel Granger causality statistic by inspecting the trained NPP with an axiomatic attribution method. In this way, CAUSE is the first technique that brings model-agnostic explainability to NPPs and endows NPPs with the ability to discover Granger causality from multi-type event sequences exhibiting various types of event interdependencies.

Contributions. Our contributions are:

- We bring model-agnostic explainability to NPPs and propose CAUSE, a novel framework for learning Granger causality from multi-type event sequences exhibiting various types of event interdependency.
- We design a two-level batching algorithm that enables efficient computation of Granger causality scalable to millions of events.
- We evaluate CAUSE on both synthetic and real-world datasets riddled with diverse event interdependency. Our experiments demonstrate that CAUSE outperforms other state-of-the-art methods.

Reproducibility. We publish our data and our code at:

`https://github.com/razhangwei/CAUSE`.

B.2 Background

In this section, we first establish some notation and then briefly introduce the background for several highly relevant topics.

Notations

Suppose there are S subjects and each subject s is associated with a multi-type event sequence $\{(t_i^s, k_i^s)\}_{i=1}^{n_s}$, where $t_i^s \in \mathbb{R}_+$ is the timestamp of the i-th event of the s-th sequence, $k_i^s \in [K]$ is the corresponding event type, and n_s is the sequence length. We denote by $\mathbf{z}_i^s \in \{0,1\}^K$ the one-hot representation of each event type k_i^s, and use $[n]$ to represent the set $\{1, \ldots, n\}$ for any positive integer n. To avoid clutter, we omit the subscript/superscript of index s when we are discussing a single event sequence and no confusion arises.

Multivariate Point Process

Multivariate point processes (MPPs) [B11] are a particular class of stochastic processes that characterize the dynamics of discrete events of multiple types in continuous time. The most common way to define an MPP is through a set of *conditional intensity functions (CIFs)*, one for each event type. Specifically, let $N_k(t) \triangleq \sum_{i=1}^{\infty} \mathbb{1}(t_i \leq t \wedge k_i = k)$ be the number of events of type k that have occurred up to t, and let $\mathcal{H}(t) \triangleq \{(t_i, k_i) | t_i < t\}$ be the *history* of all types of events before t. The CIF for event type k

is defined as the expected instantaneous event occurrence rate conditioned on history, i.e.,

$$\lambda_k^*(t) \triangleq \lim_{\Delta t \downarrow 0} \frac{\mathbb{E}[N_k(t+\Delta t) - N_k(t)|\mathcal{H}(t)]}{\Delta t},$$

where the use of the asterisk is a notational convention to emphasize that intensity is conditioned upon $\mathcal{H}(t)$.

Different parameterizations of CIFs lead to different MPPs. One classic example of MPP is the multivariate Hawkes process [B19, B20], which assumes each $\lambda_k^*(t)$ to be of the following form:

$$\lambda_k^*(t) = \mu_k + \sum_{i:t_i<t} \phi_{k,k_i}(t - t_i), \tag{B.1}$$

where $\mu_k \in \mathbb{R}_+$ is the baseline rate for event type k, and $\phi_{k,k'}(\cdot)$ for any $k, k' \in [K]$ is a non-negative-valued function (usually referred to as *kernel function*) that characterizes the excitation effect of event type k' on type k.

More recently, a class of MPPs loosely referred to as *neural point processes* have emerged in the literature [B14, B36, B27, B35]. These models parameterize CIFs with deep neural networks and generally follow an encoder-decoder design: an *encoder* successively embeds the event history $\{(t_j, k_j)\}_{j=1}^{i}$ into a vector $\mathbf{h}_i \in \mathbb{R}^{N_h}$ for each i, and a *decoder* then predicts with $\mathbf{h}_i$ the future CIFs $\lambda_k^*(t)$ after time t_i (until the next event is generated).

Most MPPs are trained by minimizing the negative log-likelihood (NLL):

$$\sum_{s=1}^{S}\sum_{i=1}^{n_s}\left[-\log \lambda_{k_i^s}^{*s}(t_i^s) + \sum_{k=1}^{K}\int_{t_{i-1}^s}^{t_i^s} \lambda_k^{*s}(t')dt'\right], \tag{B.2}$$

where $\lambda_k^{*s}(t) \triangleq \lambda_k^*(t|\mathcal{H}_s(t))$ is the CIF of the s-th sequence for the type k. In (B.2), the first term corresponds to the NLL of an event of type k_i^s being observed at t_i^s for the s-th sequence, whereas the second term is the NLL of the observation that no events of any type occurred during the window (t_{i-1}^s, t_i^s). When there are no closed-form expressions for the integrals $\int_{t_{i-1}^s}^{t_i^s} \lambda_k^{*s}(t')dt'$, Monte-Carlo approximation needs to be used to approximate either the integrals themselves or their gradients with respect to the parameters. However, these approximation techniques are inefficient and generally suffer from large variances, resulting in low convergence rate.

Granger Causality for Multi-Type Event Sequences

The Granger causality definition for multi-type event sequences is established based on point process theory [B11]. To proceed formally, for any $\mathcal{K} \subseteq [K]$, we denote by $\mathcal{H}_{\mathcal{K}}(t)$ the natural filtration expanded by the sub-process $\{N_k(t)\}_{k\in\mathcal{K}}$; that is, the sequence of smallest σ-algebra expanded by the past event history of any type $k \in \mathcal{K}$ and $t \in \mathbb{R}_+$, i.e., $\mathcal{H}_{\mathcal{K}}(t) = \sigma(\{N_k(s)|k \in \mathcal{K}, s < t\})$.[1] We further write $\mathcal{H}_{-k}(t) = \mathcal{H}_{[K]\setminus\{k\}}(t)$ for any $k \in [K]$.

[1] Here, we abuse our previous notation $\mathcal{H}(t)$ that denotes the set of events that occurred prior to t. Appendix B.IV includes a primier on measure and probability theory for readers who are less familiar with some concepts in this subsection.

Definition 1. [B16] For a K-dimensional MPP, event type k is *Granger non-causal* for event type k' if $\lambda^*_{k'}(t)$ is $\mathcal{H}_{-k}(t)$-measurable for all t.

The above definition amounts to saying that a type k is Granger non-causal for another type k' if, given the history of events other than type k, historical events of type k do not further contribute to future $\lambda^*_{k'}(t)$ at any time. Otherwise, type k is said to be *Granger causal* for type k.

Uncovering Granger causality from event sequences generally is a very challenging task, as the underlying MPP may have rather complex CIFs with abundant event interaction and non-excitative effect. As a result, existing work tends to restrict consideration to certain classes of parametric MPPs, such as Hawkes processes [B16, B37, B1]. Specifically, for multivariate Hawkes process, it is straightforward from (B.1) that a type k is Granger non-causal for another type k' if and only if the corresponding kernel function $\phi_{k',k}(\cdot) = 0$.

Attribution Methods

We view an *attribution method* for black-box functions as another "black box", which takes in a function, an input, and a baseline, and outputs a set of meaningful attribution scores, one per feature. The following is a formal definition of attribution method.

Definition 2 (Attribution Method). Suppose $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^d$ is a d-dimensional input and $\underline{\mathbf{x}} \in \mathcal{X}$ a suitable baseline. Let $\mathcal{F}_d$ be a class of functions from $\mathcal{X}$ to $\mathbb{R}$. A functional $A : \mathcal{F}_d \times \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}^d$ is called an *attribution method* for $\mathcal{F}_d$ if $A_i(f, \mathbf{x}, \underline{\mathbf{x}})$ measures the contribution of x_i to the prediction $f(\mathbf{x})$ relative to $\underline{\mathbf{x}}$ for any $f \in \mathcal{F}_d$, $\mathbf{x}, \underline{\mathbf{x}} \in \mathcal{X}$, and $i \in [d]$.

Since it is very challenging (and often subjective) to compare different attribution methods, Sundararajan, Taly, and Yan [B33] argue that attribution methods should ideally satisfy a number of axioms (i.e., desirable properties), which we re-state in Definition 3.

Definition 3. An attribution method A is said to satisfy the axiom of:

1. *Linearity*, if for any $f, g \in \mathcal{F}_d$, $\mathbf{x}, \underline{\mathbf{x}} \in \mathcal{X}$, and $c \in \mathbb{R}$,
$$\begin{aligned} &A(f, \mathbf{x}, \underline{\mathbf{x}}) + A(g, \mathbf{x}, \underline{\mathbf{x}}) = A(f + g, \mathbf{x}, \underline{\mathbf{x}}), \\ &A(cf, \mathbf{x}, \underline{\mathbf{x}}) = c \cdot A(f, \mathbf{x}, \underline{\mathbf{x}}). \end{aligned} \tag{A1}$$

2. *Completeness/Efficiency*, if for any $f \in \mathcal{F}_d$ and $\mathbf{x}, \underline{\mathbf{x}} \in \mathcal{X}$,
$$f(\mathbf{x}) - f(\underline{\mathbf{x}}) = \sum_{i=1}^{d} A_i(f, \mathbf{x}, \underline{\mathbf{x}}). \tag{A2}$$

3. *Null player*, if for any $f \in \mathcal{F}_d$ such that f does not mathematically depend on a dimension i,
$$A_i(f, \mathbf{x}, \underline{\mathbf{x}}) = 0, \tag{A3}$$
for all $\mathbf{x}, \underline{\mathbf{x}} \in \mathcal{X}$.

4. *Implementation invariance*, if for any $\mathbf{x}, \underline{\mathbf{x}} \in \mathcal{X}$, and any $f, g \in \mathcal{F}_d$ such that $f(\mathbf{x}') = g(\mathbf{x}')$ for all $\mathbf{x}' \in \mathcal{X}$,

$$A(f, \mathbf{x}, \underline{\mathbf{x}}) = A(g, \mathbf{x}, \underline{\mathbf{x}}). \tag{A4}$$

Besides these four axioms, we also identify two other useful properties of attribution methods, which are less explicitly mentioned in the literature. We state these two properties in Definition 4.

Definition 4. An attribution method A is said to satisfy

1. *Fidelity-to-control*, if for any $f \in \mathcal{F}_d$, $\mathbf{x}, \underline{\mathbf{x}} \in \mathcal{X}$, and $i \in [d]$,

$$x_i = \underline{x}_i \quad \Rightarrow \quad A_i(f, \mathbf{x}, \underline{\mathbf{x}}) = 0. \tag{P1}$$

2. *Batchability*, if for any $f \in \mathcal{F}_d$ and any n input/baseline pairs $\{(\mathbf{x}_i, \underline{\mathbf{x}}_i)\}_{i \in [n]}$, there exists a function $F : \mathcal{X}^n \mapsto \mathbb{R}$ such that

$$A(F, \mathbf{X}, \underline{\mathbf{X}}) = [A(f, \mathbf{x}_1, \underline{\mathbf{x}}_1), \ldots, A(f, \mathbf{x}_n, \underline{\mathbf{x}}_n)], \tag{P2}$$

where $\mathbf{X} \triangleq [\mathbf{x}_1, \ldots, \mathbf{x}_n]$ and $\underline{\mathbf{X}} \triangleq [\underline{\mathbf{x}}_1, \ldots, \underline{\mathbf{x}}_n]$.

Many popular attribution methods satisfy most of these six properties, as we show in the Proposition 1 and 2.

Proposition 1. *Integrated Gradients [B33] satisfies all four axioms (A1–A4) and two properties (P1–P2), and DeepLIFT [B32] satisfies all but the implementation invariance (A4). In particular, a choice of F for both methods satisfying batchability* (P2) *is* $F(\mathbf{X}) \triangleq \sum_{i=1}^{n} f(\mathbf{x}_i)$.

Proposition 2. *For any* $U \subseteq [d]$, *let* $\bar{U} \triangleq [d] \setminus U$ *and define* $\mathbf{x}_U \sqcup \underline{\mathbf{x}}_{\bar{U}}$ *to be the spliced data point between* $\mathbf{x}$ *and* $\underline{\mathbf{x}}$ *such that for any* $i \in [d]$

$$[\mathbf{x}_U \sqcup \underline{\mathbf{x}}_{\bar{U}}]_i \triangleq \begin{cases} \mathbf{x}_i & i \in U, \\ \underline{\mathbf{x}}_i & i \in \bar{U}. \end{cases} \tag{B.3}$$

Then Shapley values [B31] with a value function $v_{f,\mathbf{x},\underline{\mathbf{x}}}(U) \triangleq f(\mathbf{x}_U \sqcup \underline{\mathbf{x}}_{\bar{U}})$ *satisfies all four axioms (A1–A4) and the fidelity-to-control (P1).*

Proof. We include proofs for both propositions in Appendix B.II. □

B.3 Method

In this section, we formally present CAUSE, a novel framework for learning Granger causality from multi-type event sequences. Our framework consists of two steps: first, it trains a neural point process (NPP) to fit the underlying event sequence data; then it inspects the predictions of the trained NPP to compute a Granger causality statistic with some "well-behaved" attribution method $A(\cdot)$, which we assume satisfies the following properties: linearity (A1), completeness (A2), null player (A3), fidelity-to-control (P1), and batchability (P2).

In what follows, we first describe the architecture of the used NPP in Section B.3. Then we elaborate the intuition and the definition of our Granger causality statistic in Section B.3. Section B.3 explains the computational challenges and presents a highly efficient algorithm for computing such statistic. We conclude this section by discussing the choice of attribution methods for CAUSE in Section 9.

A Semi-Parametric Neural Point Process

The design of our NPP follows the general encoder-decoder architecture of existing NPPs (Section B.2), but we innovate the decoder part to enable both modeling flexibility and computational feasibility.

Encoder. First, we convert each event i into an embedding vector $\mathbf{v}_i$ that summarizes both the temporal and the type information for that event, as follows:

$$\mathbf{v}_i = [\vartheta(t_i - t_{i-1}); \mathbf{V}^T \mathbf{z}_i], \tag{B.4}$$

where $\vartheta(\cdot)$ is a pre-specified function that transforms the elapsed time into one or more temporal features (simply chosen to be identity function in our experiments), $\mathbf{V}$ is the embedding matrix for event types, and recall that $\mathbf{z}_i$ is the one-hot encoding of the even type k_i.

We then obtain the embedding of a history from event embedding sequences by

$$\mathbf{h}_i = \mathrm{Enc}(\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_i), \tag{B.5}$$

where $\mathrm{Enc}(\cdot)$ is a sequence encoder and chosen to be a Gated Recurrent Unit (GRU) [B9] in our experiments.

Decoder. An ideal decoder should fullfill the following two desiderata: (a) it should be *flexible enough* to produce from $\mathbf{h}_i$ a wide variety of $\lambda_k^*(t)$ with complex time-varying patterns; and (b) it should also be *computationally manageable*, particularly in terms of computing the cumulative intensity $\int_{t_i}^{t_{i+1}} \lambda_k^*(t')dt'$, a key term in the log-likelihood-based training given in (B.2) *and* the definition of our Granger causality statistic in the subsequent subsections.

We propose a novel semi-parametric decoder that enjoys both the flexible modeling of CIF and computational feasibility. Specifically, for each $i \in [n]$, we define the CIF

$\lambda_k^*(t)$ on $(t_i, t_{i+1}]$ to be a weighted sum of a set of basis functions, as follows:

$$\lambda_k^*(t) = \sum_{r=1}^{R} \alpha_{k,r}(\mathbf{h}_i)\psi_r(t - t_i), \tag{B.6}$$

where $\{\psi_r(\cdot)\}_{r=1}^R$ is a set of pre-specified positive-valued basis functions, and $\boldsymbol{\alpha} : \mathbb{R}^{N_h} \mapsto \mathbb{R}_+^{K \times R}$ is a feedforward neural network that computes R positive weights for each of the K event types. In this way, by choosing $\{\psi_r(\cdot)\}_{r=1}^R$ to be a rich-enough function family, the CIFs defined in (B.6) are able to express a wide variety of time-varying patterns. More importantly, since the parts relevant to neural networks—$\boldsymbol{\alpha}(\cdot)$ and $\mathrm{Enc}(\cdot)$—are separated from the basis functions, we can evaluate the integral $\int_{t_i}^{t_{i+1}} \lambda_k^*(t')dt'$ analytically, as follows:

$$\int_{t_i}^{t_{i+1}} \lambda_k^*(t')dt' = \sum_{r=1}^{R} \alpha_{k,r}(\mathbf{h}_i)\Psi_r(t_{i+1} - t_i), \tag{B.7}$$

where $\Psi_r(\Delta t) \triangleq \int_0^{\Delta t} \psi_r(t)\, dt$ is generally available for many parametric basis functions.

We choose the basis functions $\{\psi_r(\cdot)\}_{r=1}^R$ to be the densities of a Gaussian family with increasing means and variances. This design of basis functions reflects a reasonable inductive bias that the CIFs should vary more smoothly as the time increases. The details are given in Appendix B.II.

From Event Contributions to a Granger Causality Statistic

Now that we have trained a flexible NPP that can successively update the history embedding after each event i occurrence and then predict the future CIFs $\lambda_k^*(t)$ after t_i until t_{i+1}; we would like to ask: *can we quantify the contribution of each past event to each prediction?* Since in our case $\lambda_k^*(t)$'s are instantiated by two potentially highly nonlinear neural networks (i.e., $\mathrm{Enc}(\cdot)$ and $\boldsymbol{\alpha}(\cdot)$), it is not as straightforward to obtain the past event's contribution to current event occurrence as in the case of some parametric MPPs (e.g., Hawkes processes).

A natural idea for the aforementioned question would be applying some attribution method to $\lambda_k^*(t)$'s. To do so, however, there are two challenges. First, the predictions in our case are time-varying functions rather than static quantities (e.g., the probability of a class, as commonly seen in existing applications of attribution methods); thus it is unclear which target should be attributed. Second, as the input to $\lambda_k^*(t)$'s are multi-type event sequences with asynchronous timestamps, it is also unclear which baseline should be used.

We tackle the first challenge by setting the *cumulative intensity* $\int_{t_i}^{t_{i+1}} \lambda_k^*(t')\, dt'$ to be the attribution target. This is not only because the cumulative intensity reflects the overall effect of $\lambda_k^*(t')$ on $(t_i, t_{i+1}]$, but also because it has a clear meaning in the context of point processes: it is the rate of the Poisson distribution that the number of events of type k on $(t_i, t_{i+1}]$ satisfies. More importantly, since the cumulative intensity has a closed form as in (B.7), its gradients with respect to its input can be computed both precisely and efficiently. Note that by adopting this target, the input now includes not only $\{(t_i, \mathbf{z}_j)\}_{j \le i}$ but also t_{i+1}; thus we define $\mathbf{x}_i \triangleq [t_1, \mathbf{z}_1, \dots, t_i, \mathbf{z}_i, t_{i+1}]$ and write the target $\int_{t_i}^{t_{i+1}} \lambda_k^*(t')\, dt'$ as $f_k(\mathbf{x}_i)$.

As for the second challenge, we choose the baseline of an input $\mathbf{x}_i$ to be $\underline{\mathbf{x}}_i \triangleq [t_1, \mathbf{0}, \dots, t_{i-1}, \mathbf{0}, t_{i+1}]$; that is, the one-hot encodings of all observed event types are replaced with zero vectors. Since $\mathbf{x}_i$ and $\underline{\mathbf{x}}_i$ only differ in the dimensions corresponding to the event types, i.e., $\{z_{j,k_j}\}_{j \le i}$, by the fidelity-to-control (P1), then only these dimensions will have non-zero attributions. With completeness (A2), it further implies that for every type k

$$f_k(\mathbf{x}_i) - f_k(\underline{\mathbf{x}}_i) = \sum_{j=1}^{i} A_j(f_k, \mathbf{x}_i, \underline{\mathbf{x}}_i), \tag{B.8}$$

where $A_j(f_k, \mathbf{x}_i, \underline{\mathbf{x}}_i)$ is the attribution to z_{j,k_j}. Thus, the term $A_j(f_k, \mathbf{x}_i, \underline{\mathbf{x}}_i)$ can be viewed as the *event contribution* of the j-th event to the cumulative intensity prediction $f_k(\mathbf{x}_i)$ relative to the baseline $f_k(\underline{\mathbf{x}}_i)$. Besides, event *timestamps* are identical in $\mathbf{x}_i$ and $\underline{\mathbf{x}}_i$, thus this contribution comes only from the event *type* and denotes how type k_j contributes to the prediction of type k for a specific event history $\mathbf{x}_i$.

A Granger Causality Statistic. We have established $A_j(f_k, \mathbf{x}_i, \underline{\mathbf{x}}_i)$'s as the past events' contribution to the cumulative intensity $f_k(\mathbf{x}_i)$ on interval $(t_i, t_{i+1}]$. A further question is: *can we infer from these event contributions for individual predictions the population-level Granger causality among event types?*

To answer this question, we define a novel statistic indicating the Granger causality for type k' to type k as follows:

$$Y_{k,k'} \triangleq \frac{\sum_{s=1}^{S} \sum_{i=1}^{n_s} \sum_{j=1}^{i} \mathbb{I}(k_j^s = k') A_j(f_k, \mathbf{x}_i^s, \underline{\mathbf{x}}_i^s)}{\sum_{s=1}^{S} \sum_{j=1}^{n_s} \mathbb{I}(k_j^s = k')}. \tag{B.9}$$

Here the numerator aggregates the event contributions for all event occurrences over the whole dataset, and denominator accounts for the fact that some event types may occur far more frequently than other types, which can lead to unreasonably large scores if used without such normalization. Note that an event contribution $A_j(f_k, \mathbf{x}_i^s, \underline{\mathbf{x}}_i^s)$ may be negative when the event j exerts an inhibitive effect; thus $Y_{k,k'}$ can also be negative and characterize the Granger causality from type k' to type k even when the influence is inhibitive.

Attribution Regularization. One caveat in (B.8) and (B.9) is that our chosen baselines $\underline{\mathbf{x}}_i$ have never appeared in the training procedure, thus the value of $f(\underline{\mathbf{x}}_i)$ may be meaningless or even misleading. Ideally, we would like $f_k(\underline{\mathbf{x}}_i)$ to be the cumulative intensity of type k given that history prior to t_i consists of "null" events at $t_1, t_2, \dots, t_i$. Thus a natural prior reflecting this idea is to make $f_k(\underline{\mathbf{x}}_i)$ nearly zero for any handcrafted baseline $\underline{\mathbf{x}}_i$. Such an "invariance" property on f can be achieved by adding an

Algorithm 1: Computation of the Granger causality statistic.

```
Input: Event sequences $\{\{(t_i^s, k_i^s)\}_{i\in[n_s]}\}_{s\in[S]}$, an attribution method $A(\cdot)$,
       and a trained NPP
Output: Granger causality statistic $\mathbf{Y}$.
Initialize $\tilde{\mathbf{Y}} = \mathbf{0}, \mathcal{I} = [S]$ ;
while $|\mathcal{I}| > 0$ do
    Sample a batch of sequence indices $\mathcal{B} \subset \mathcal{I}$ ;
    for $k = 1, \ldots, K$ do
        Compute $\mathbf{C} = A(\sum_{s\in\mathcal{B}} \sum_{i=1}^{n_s} F_{k,i}^s, \mathbf{X}, \underline{\mathbf{X}})$;
        for $k' = 1, \ldots, K$ do
            $\tilde{Y}_{k,k'} \mathrel{+}= \sum_{s\in\mathcal{B}} \sum_{j=1}^{n_s} \mathbb{I}(k_j^s = k') C_j^s$
    $\mathcal{I} \leftarrow \mathcal{I} \setminus \mathcal{B}$;
Compute $Y_{k,k'} = \tilde{Y}_{k,k'} / \sum_{s=1}^{S} \sum_{j=1}^{n} \mathbb{I}(k_j^s = k'), \forall k, k' \in [K]$.
```

auxiliary l_1 regularization for each $\mathbf{x}_i$ in the NLL given in (B.2), leading to a training objective

$$\sum_{s=1}^{S} \sum_{i=1}^{n_s} \Big\{ \underbrace{-\log \lambda_{k_i^s}^*(t_i^s) + \sum_{k=1}^{K} f_k(\mathbf{x}_{i-1}^s)}_{\text{negative log-likelihood}} + \underbrace{\sum_{k=1}^{K} \eta f_k(\underline{\mathbf{x}}_{i-1}^s)}_{\text{regularization}} \Big\}, \tag{B.10}$$

where η is a hyperparameter.

Computing the Granger Causality Statistic

While (B.9) defines $Y_{k,k'}$'s analytically, it is rather challenging to compute them. This is because a naive implementation would require applying $A(\cdot)$ at each event occurrence, which is computationally prohibitive for a dataset of millions of events. Note that the normalization in (B.9) can be easily calculated; so if we write $\tilde{Y}_{k,k'}^s \triangleq \sum_{i=1}^{n_s} \sum_{j=1}^{i} \mathbb{I}(k_j^s = k') A_j(f_k, \mathbf{x}_i^s, \underline{\mathbf{x}}_i^s)$, the problem is reduced to how to efficiently compute $\sum_{s=1}^{S} \tilde{Y}_{k,k'}^s$.

We propose an efficient algorithm to compute $\sum_{s=1}^{S} \tilde{Y}_{k,k'}^s$, which is summarized in Algorithm 1. At the core of our algorithm are two levels of batching: (a) *intra-sequence batching*, which allow the computation of $\tilde{Y}_{k,k'}^s$ with only one call of $A(\cdot)$; and (b) *inter-sequence batching*, which enables batch computation of $\{Y_{k,k'}^s\}_{s\in\mathcal{B}}$ for a mini-batch of event sequences indexed by $\mathcal{B}$. We explain the details of these two levels of batching as follows.

Intra-Sequence Batching. As this part only deals with a particular event sequence, to simplify the notation, we omit the sequence index s for now. Note that $\mathbf{x}_1 \prec \mathbf{x}_2 \prec \cdots \prec \mathbf{x}_n$ and due to the recurrent nature of f, all $f(\mathbf{x}_i)$ for $i \in [n]$ can be computed in a single forward pass with the shared input $\mathbf{x}_n$. Denote by $F = \{F_{k,i}(\cdot)\}_{k\in[K],i\in[n]}$ a matrix-valued function such that $F_{k,i}(\mathbf{x}_n) = f_k(\mathbf{x}_i)$ for any $k \in [K], i \in [n]$.

The equivalence between f and F means that,

$$A_j(f_k, \mathbf{x}_i, \underline{\mathbf{x}}_i) = A_j(F_{k,i}, \mathbf{x}_n, \underline{\mathbf{x}}_n),$$

which further implies that we can rewrite $\tilde{Y}^s_{k,k'}$ as a weighted sum of attribution scores for the same input $\mathbf{x}_n$ and baseline $\underline{\mathbf{x}}_n$. Since we are not interested in computing the individual attribution scores but their sum, we can leverage the linearity property (A1) to compute the attribution scores directly for the sum, as shown in the following proposition.

Proposition 3. *For an attribution method $A(\cdot)$ with the linearity* (A1) *and the null player* (A3), *it holds that*

$$\tilde{Y}^s_{k,k'} = \sum_{j=1}^{n_s} \mathbb{I}(k^s_j = k') A_j \left(\sum_{i=1}^{n_s} F_{i,k_i}, \mathbf{x}^s_n, \underline{\mathbf{x}}^s_n \right). \tag{B.11}$$

Proof. The proof is in Appendix B.II. □

Inter-Sequence Batching. We now discuss how to efficiently compute $\{Y^s_{k,k'}\}_{s\in\mathcal{B}}$ for a mini-batch of event sequences indexed by $\mathcal{B}$. The key idea for a significant computational speed-up here is that if $A(\cdot)$ satisfies batchability (P2), we can then batch the computation of different sequences with a single call of $A(\cdot)$.

To simplify the discussion, we assume without loss of generality that $\mathcal{B} = \{1, \ldots, |\mathcal{B}|\}$ and $n_s \equiv n$ for all $s \in \mathcal{B}$. Let $\mathbf{X} = [\mathbf{x}_s]_{s\in\mathcal{B}}$ and analogously the corresponding baselines $\underline{\mathbf{X}}$. We further override our previous notation and denote by $F = \{F^s_{k,i}(\cdot)\}_{s\in[S],k\in[K],i\in[n]}$ a new tensor-valued function such as that $F^s_{k,i}(\mathbf{X}) = f_k(\mathbf{x}^s_i)$. Then with Proposition 1, we have that

$$A(\sum_{s\in\mathcal{B}} \sum_{i=1}^{n_s} F^s_{k,i}, \mathbf{X}, \underline{\mathbf{X}}) = \left[A_j(\sum_{i=1}^{n_s} F_{i,k_i}, \mathbf{x}^s_n, \underline{\mathbf{x}}^s_n) \right]_{\substack{s\in\mathcal{B} \\ j\in[n]}}.$$

Time Complexity Analysis. With our two-level batching scheme, Algorithm 1 only requires $O(SK/B)$ invocations of $A(\cdot)$, a significant reduction from the $O(SNK)$ invocations required by a naive implementation that directly calculates $Y_{k,k'}$'s, where N is the average sequence length. Since modern computation hardware (such as GPUs) enables calling $A(\cdot)$ with a batch of inputs being almost as fast as calling it with a single input, our algorithm can achieve up to *three orders-of-magnitude speedup* over a naive implementation on datasets with relatively large N and B. (See Section B.4 for empirical evaluations.)

Choice of Attribution Methods

In our experiments, we choose the attribution method $A(\cdot)$ to be the Integrated Gradients, which is defined as follow:

$$\mathrm{IG}(f, \mathbf{x}, \underline{\mathbf{x}}) \triangleq (\mathbf{x} - \underline{\mathbf{x}}) \odot \int_0^1 \left. \frac{\partial f(\tilde{\mathbf{x}})}{\partial \tilde{\mathbf{x}}} \right|_{\tilde{\mathbf{x}}=\underline{\mathbf{x}}+\alpha(\mathbf{x}-\underline{\mathbf{x}})} d\alpha, \tag{B.12}$$

where $\odot$ is the Hadamard product. Nevertheless, CAUSE does not depend on a specific attribution method but a set of properties that we have stated upfront; this means that any other attribution methods that satisfy these properties (e.g., DeepLIFT) should be applicable to CAUSE. Also note that batchability (P2) is only used in the inter-sequence batching for speeding up the computation; thus, if efficiency is less of a concern, or the computation of attributions for different inputs can be accelerated in alternative ways,[2] attribution methods that only violate batchability, such as Shapley values, should also be applicable.

B.4 Experiments

In this section, we present the experiments that are designed to evaluate CAUSE and answer the following three questions:

- **Goodness-of-Fit**: How good is CAUSE at fitting multi-type event sequences?
- **Causality Discovery**: How accurate is CAUSE at discovering Granger causality between event types?
- **Scalability**: How scalable is CAUSE?

The experimental results on five datasets show that CAUSE (a) outperforms state-of-the-art methods in both fitting and discovering Granger causality from event sequences of diverse event interdependency, (b) can identify Granger causality on real-world datasets that agrees with human intuition, and (c) can compute the Granger causality statistic three orders-of-magnitude faster due to our optimization.

Experimental Setup

Datasets. We designed three synthetic datasets to reflect various types of event interactions and temporal effects.

- **Excitation**: This dataset was generated by a multivariate Hawkes process, whose CIFs are defined in (B.1). The exponential decay kernels were used, and a weighted ground-truth causality matrix was constructed with the ℓ_1 norms of the kernel functions $\phi_{k,k'}(\cdot)$.
- **Inhibition**: This dataset was generated by a multivariate self-correcting process [B24], whose CIFs are of the form: $\lambda_k^*(t) = \exp(\alpha_k t + \sum_{i:t_i<t} w_{k,k_i})$, where $a_k > 0$ and $w_{k,k'} \leq 0$. A weighted ground-truth causality matrix was formed with the pairwise weights $w_{k,k'}$.
- **Synergy**: Generated by a proximal graphical event model (PGEM) [B8], this dataset contains synergistic effects between a pair of event types to a third event type. A binary ground-truth causality matrix was constructed from the dependency graph of the PGEM.

[2] In fact, for almost all attribution methods, the attribution for different inputs is embarrassingly parallelizable.

We also included two real-world datasets used in existing literature.

- **IPTV** [B26]: Each sequence records the history of TV watching behavior of a user, and the event types are the TV program categories. This dataset, however, does not contain ground-truth causality between TV program categories.
- **MemeTracker (MT)**:[3] Each sequence represents how a phrase or quote appeared on various online websites over time during the period of August 2008 to April 2009, and the event types are the domains of the top websites. Like previous studies [B1, B35], a weighted ground-truth causality matrix was *approximated* by whether one site contains any URLs linking to another site.

The parameter settings for synthetic datasets, the preprocessing steps for real-world datasets, and the dataset statistics are detailed in Appendix B.III.

Methods for Comparsison. We compared our method to the following baselines:

- **HExp**: Hawkes process with fixed exponential kernels.
- **HSG** and **NHPC**: Hawkes process with sum of Gaussian kernels [B37] and nonparametric Hawkes process cumulant matching [B1]. These two methods represent the state-of-the-art parametric and nonparametric methods for learning Granger causality for Hawkes process, respectively.
- **RPPN**: Recurrent point process network [B35], to the best of our knowledge, the only NPP that can provide summary statistics for Granger causality, which is enabled by its use of an attention mechanism.

The implementation details and hyperparameter configurations for CAUSE and various baselines are provided in Appendix B.III

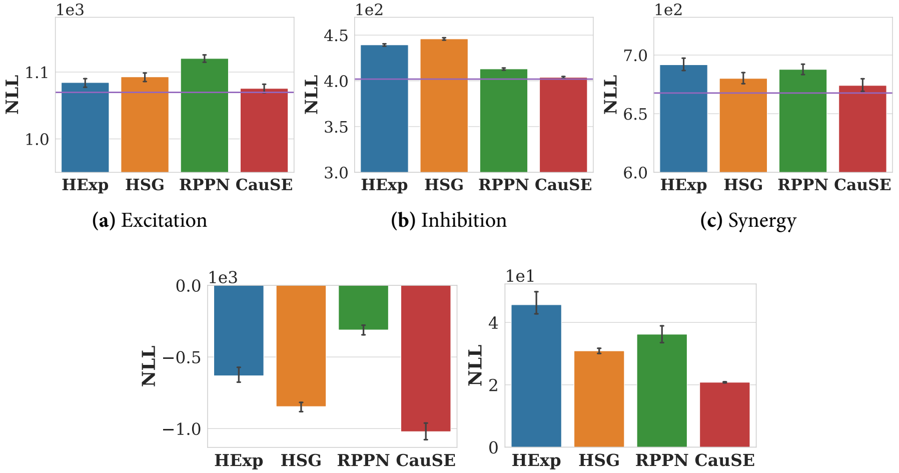
Evaluation Metrics. The hold-out negative log-likelihood (**NLL**) was used for evaluating the goodness-of-fit of each method on various datasets, and the **Kendall's τ** coefficient and the area under the ROC curve (**AUC**) were used for evaluating the estimated Granger causality matrix against the ground truth. Non-binary ground-truth causality matrices were binarized at zero in the evaluation of AUCs. We performed five-fold cross-validation and report the average results.

Detailed Results

Goodness-of-fit

We start by examining the goodness-of-fit of various methods on various datasets, since if a method fails to fit the data, it is unlikely to detect the true Granger causality between event types. As shown in Figure B.1, CAUSE attains smaller NLLs than all baselines on

[3] `https://www.memetracker.org/data.html`



(a) Excitation **(b)** Inhibition **(c)** Synergy

(d) IPTV **(e)** MT

Fig. B.1: Hold-out NLLs of various methods, where horizontal lines denote the ground-truth NLLs. CAUSE attains the best NLLs on all datasets.

Table B.1: Results for Granger causality discovery on the four datasets with ground-truth causality available. The best and the second best results on each dataset are emboldened and italicized, respectively.

	Excitation		**Inhibition**		**Synergy**		**MT**	
Method	AUC	Kendall's τ	AUC	Kendall's τ	AUC	Kendall's τ	AUC	Kendall's τ
HExp	0.858±0.004	0.453±0.005	*0.546±0.002*	*0.102±0.002*	0.872±0.058	0.251±0.039	0.404±0.009	-0.061±0.005
HSG	**0.997±0.001**	**0.635±0.002**	0.490±0.002	-0.013±0.002	0.876±0.007	0.254±0.039	*0.539±0.008*	*0.024±0.005*
NPHC	0.782±0.007	0.337±0.010	0.400±0.054	-0.138±0.067	0.741±0.129	0.163±0.087	N/A	N/A
RPPN	0.595±0.010	0.136±0.012	0.448±0.003	-0.066±0.002	*0.891±0.043*	*0.264±0.029*	0.492±0.004	-0.005±0.002
CAUSE	*0.920±0.012*	*0.533±0.013*	**0.921±0.021**	**0.532±0.021**	**0.991±0.004**	**0.331±0.003**	**0.623±0.012**	**0.075±0.007**

all datasets, suggesting that CAUSE consistently has a better fit than all baselines. Notably, on all three synthetic datasets, the NLLs of CAUSE nearly match those computed by the ground-truth models. These results confirm the flexibility of CAUSE in learning the various types of event interactions and temporal effects.

Causality Discovery

We now examined the performance of CAUSE on Granger causality discovery, both quantitatively and qualitatively.

Quantitative Analaysis. Table B.1 shows values of AUC and Kendall's τ of various methods on the four datasets that have ground-truth causality. The results in the table support the following conclusions.

First, CAUSE performs the best overall and is most robust to various types of event interactions. It not only significantly outperforms all baselines on three of the four datasets (i.e., Inhibition, Synergy, and MT), but also achieves a close-second on Excitation, in which events were generated by a Hawkes process, and CAUSE is supposed to have a disadvantage relative to Hawkes process-based baselines.

Second, once the underlying data generation process violates the assumptions of Hawkes process and exhibits complex event interactions other than excitation, Hawkes process-based methods tend to perform poorly, as seen from Inhibition and Synergy.

Finally, despite both being NPP-based methods, RPPN performs significantly worse than CAUSE on all datasets. We suspect that this underperformance is caused by two issues in RPPN's construction of the Granger causality statistics with the attention weights. First, RPPN restricts all attention weights to be positive, thus cannot distinguish between excitative and inhibitive effects. Second, attention mechanism may not correctly attribute the model's prediction to its inputs, as shown in several recent studies [B25, B30].

Qualitative Analysis. Figure B.2 shows the heat map for the Granger causality matrix of IPTV dataset estimated by CAUSE. Almost all diagonal entries have large positive values, indicating that users, on average, exhibit strong tendencies to watch the TV programs of the same category. Several positive associations between different TV program categories are also observed, such as from military, laws, finance, and education to news, and from kids and music to drama. These results agree with common sense and are consistent with the findings of an existing study with HSG [B37]. Our method also suggests several meaningful negative associations, including ads to drama and education to entertainment; such negative associations, however, can never be detected by models that only consider the excitations between events, such as HSG.

Scalability

Finally, we investigate the scalability of CAUSE in computing the Granger causality statistic by Algorithm 1. Figure B.3 shows how much speedup Algorithm 1 achieves over a naive implementation with different sequence lengths and batch sizes. The results demonstrate that with batch size and average sequence length both being relatively large (i.e., greater or equal to 16 and 100, respectively), our algorithm can achieve over *three orders-of-magnitude speedup* relative to a native implementation. Furthermore, the speedup scales almost linearly with sequence length and batch size when they do not exceed 150 and 16, respectively, which is consistent with our analysis in Section B.3. Beyond this regime, only a sublinear relationship between the speedup and batch size or sequence length is observed, which is because the GPU we tested on was reaching its maximum utilization.

We have presented CAUSE, a novel framework for learning Granger causality between event types from multi-type event sequences. At the core of CAUSE are two steps: first, it trains a flexible NPP model to capture the complex event interdependency, then it computes a novel Granger causality statistic by inspecting the trained model with an axiomatic attribution method. A two-level batching algorithm is derived to compute the statistic efficiently. We evaluate CAUSE on both synthetic and real-world datasets abundant with diverse event interactions and show the effectiveness of CAUSE on identifying the Granger causality between event types.

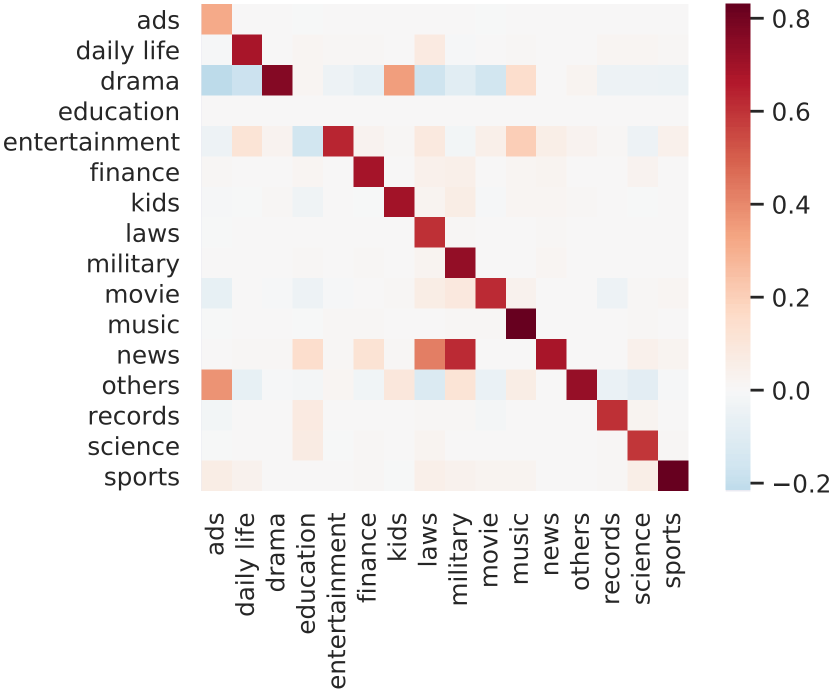


Fig. B.2: Visualization of the estimated Granger causality statistic matrices on IPTV.

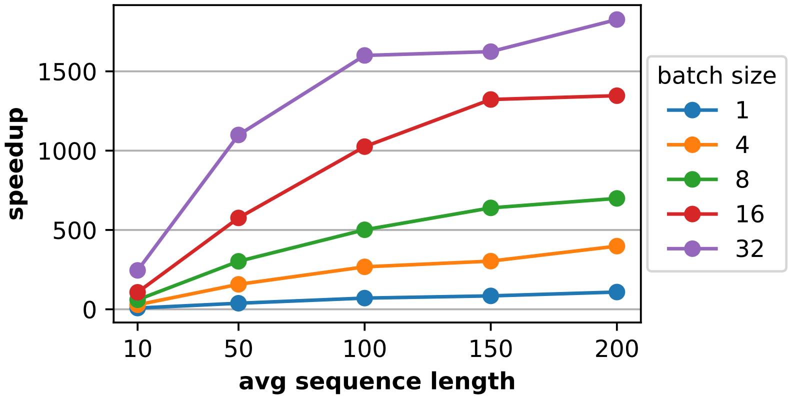


Fig. B.3: The speedup achieved by Algorithm 1 relative to a naive implementation with different average sequence lengths and batch sizes.

Appendix

B.I Additional Related Work

Event Sequence Modeling With the increasing availability of multi-type event sequences, there has been considerable interest in modeling such data for both prediction and inference. The majority of prior research in this direction has been based on the theory of point processes [B11], a particular class of stochastic processes that characterize the distribution of random points on the real line. Notably, Hawkes process [B19, B20], a special class of point process, has been widely investigated, partly due to its ability to capture mutual excitation among events and its mathematical tractability. However, Hawkes process assumes that past events can only independently and additively influence the occurrence of future events, and that influence can only be excitative; these inherent limitations have restricted its modeling flexibility and render it unable to capture complex event interaction in real-world data.

As such, other more flexible models have been proposed, including the *piecewise-constant conditional intensity model* (PCIM) [B18] and its variants [B34, B8], and more recently a class of models loosely referred to as *neural point processes* (NPPs) [B14, B36, B27, B35]. These models, particularly NPPs, generally enjoy better predictive performance than parametric point processes, since they use more expressive machine learning models (e.g., decision trees, random forests, or recurrent neural networks) to sequentially compute the conditional intensity until next event is generated. A significant weakness of these models, however, is that they are generally uninterpretable and thus unable to provide summary statistics for determining the Granger causality among event types.

Granger Causality Discovery In his seminal paper, Granger [B17] first proposed the concept of Granger causality for time series data. Many approaches have been proposed for uncovering Granger causality for multivariate time series, including the Hiemstra-Jones test [B21] and its improved variant [B13], Lasso-Granger method [B3], and approaches based on information-theoretic measures [B22]. However, as these methods are designed for the synchronous multivariate time series, they are not directly applicable to asynchronous multi-type event sequence data, since otherwise one has to discretize the continuous observation window.

Didelez [B12] first established the Granger causality for event types in event sequences under the framework of marked point processes. Later, Eichler, Dahlhaus, and Dueck [B16] shows that Granger causality for Hawkes processes is entirely encoded in the excitation kernel functions (also called impact function). To our best knowledge, existing research for Granger causality discovery from event sequences appears to be limited to the case of Hawkes process [B16, B37, B1], possibly because of this direct link between the process parameterization and Granger causality.

Prediction Attribution for Black-Box Models Prediction attribution, the task of assigning to each input feature a score for representing the feature's contribution to model prediction, has been attracting considerable interest in the field due to its ability to provide insight into predictions made by black-box models such as neural networks. While various approaches have been proposed, there are two prominent groups of approaches: perturbation-based and gradient-based approaches. Perturbation approaches [B39] typically comprise, first, removing, masking, or altering a feature, and then measuring the attribution score of that feature by the change of the model's output. While perturbation-based methods are simple, intuitive, and applicable to almost all black-box models, the quality of the resultant scores is often sensitive to how the perturbation is performed. Moreover, as these methods scale linearly with the number of input features, they become computationally unaffordable for high-dimensional inputs.

In contrast, backpropagation-based methods construct the attribution scores based on the estimation of local gradients of the model around the input instance with backpropagation. The ordinary gradients, however, could suffer from a "saturation" problem for neural networks with activation functions that contain constant-valued regions (e.g., rectifier linear unit (ReLUs)); that is, the gradient coming into a ReLU during the backward pass is zero'd out if the input to the ReLU during the forward pass is in a constant region. One valid solution to this issue is to replace gradients with discrete gradients and use a modified form of backpropagation to compose discrete gradients into attributions, such as layer-wise relevance propagation (LRP) [B4] and DeepLIFT [B32]. Another solution, proposed by Integrated Gradient (IG) [B33], is to use the line integral of the gradients along the path from the input to a chosen baseline. Sundararajan, Taly, and Yan [B33] show that IG satisfies many desirable properties, as detailed in Proposition 1.

It is worth mentioning that much existing work often uses the intermediate results, produced by certain intelligible neural network architecture, as the attribution scores for an input. A most notable example of such an idea is the use of attention weights induced by some attention mechanism as the importance of the input [B6, B38]. Recently, however, there are growing concerns on the validity of attention weights being used as the explanation of neural networks [B25, B30]. In particular, Jain and Wallace [B25] show that across a variety of NLP tasks, the learned attention weights are frequently uncorrelated with feature importance produced by gradient-based prediction attribution methods, and random permutation of attention weights can nonetheless yield equivalent predictions.

B.II Additional Technical Details

Proof of Proposition 1

Proof. That both IG and DeepLIFT satisfy A1–A4 has been established in [B33]. P1 is straightforward from the definion of either method. Thus, we only prove that both methods satisfy batchability (P2) with $F(\mathbf{X}) \triangleq \sum_{i=1}^{n} f(\mathbf{x}_i)$.

To prove that IG satisfies batchability, we first rewrite the $\text{IG}(F, \mathbf{X}, \underline{\mathbf{X}})$ as follows:

$$\begin{aligned}
\text{IG}(F, \mathbf{X}, \underline{\mathbf{X}}) &= (\mathbf{X} - \underline{\mathbf{X}}) \odot \int_0^1 \nabla_{\mathbf{X}} F \left[\underline{\mathbf{X}} + \alpha(\mathbf{X} - \underline{\mathbf{X}})\right] d\alpha \\
&= (\mathbf{X} - \underline{\mathbf{X}}) \odot \int_0^1 \sum_{i=1}^{n} \nabla_{\mathbf{X}} f \left[\underline{\mathbf{x}}_i + \alpha(\mathbf{x} - \underline{\mathbf{x}}_i)\right] d\alpha \\
&= (\mathbf{X} - \underline{\mathbf{X}}) \odot \int_0^1 \sum_{i=1}^{n} \{\nabla_{\mathbf{x}_i} f \left[\underline{\mathbf{x}}_i + \alpha(\mathbf{x} - \underline{\mathbf{x}}_i)\right]\} \mathbf{e}_i^T \, d\alpha \\
&= (\mathbf{X} - \underline{\mathbf{X}}) \odot \left[\int_0^1 \nabla_{\mathbf{x}_i} f \left[\underline{\mathbf{x}}_i + \alpha(\mathbf{x} - \underline{\mathbf{x}}_i)\right] d\alpha\right]_{i=1,\ldots,m},
\end{aligned}$$

where the second step is due to that summation and gradients are swapable, and the third step is because the gradients of different terms are separable. Thus, we have

$$[\text{IG}(F, \mathbf{X}, \underline{\mathbf{X}})]_{:,i} = (\mathbf{x}_i - \underline{\mathbf{x}}_i) \odot \int_0^1 \nabla_{\mathbf{x}_i} f \left[\underline{\mathbf{x}}_i + \alpha(\mathbf{x} - \underline{\mathbf{x}}_i)\right] d\alpha = \text{IG}(f, \mathbf{x}_i, \underline{\mathbf{x}}_i), \tag{B.13}$$

which establishes the formula.

The proof of DeepLIFT satisfying batchability can be established in a similar way as IG. The key part, shown in the Proposition 2 of [B2], is that the attribution scores produced by DeepLIFT for a neural-network-like function f, an input $\underline{\mathbf{x}}$, and a baseline $\underline{\mathbf{x}}$, i.e., $\text{DeepLIFT}(f, \mathbf{x}, \underline{\mathbf{x}})$ can be viewed as the Hadamard product between $\mathbf{x} - \underline{\mathbf{x}}$ and a modified gradient of f at all its internal nonlinear layers. Since the last layer of F is a simple linear addition of all $f(\mathbf{x}_i)$'s, the modified gradient of F for input $\mathbf{x}_i$ is the same as the one of f for $\mathbf{x}_i$. Thus, we have

$$[\text{DeepLIFT}(F, \mathbf{X}, \underline{\mathbf{X}})]_{:,i} = \text{DeepLIFT}(f, \mathbf{x}_i, \underline{\mathbf{x}}_i). \tag{B.14}$$

□

Proof of Proposition 2

We first briefly review Shapley values. Suppose there is a team of d players working together to earn a certain amount of value. The value that every coalition $U \subseteq [d]$ achieves is $v(U)$, where $v : 2^d \mapsto \mathbb{R}$ is a value function. Shapley values, proposed by Shapley [B31], provide a well-motivated way to decide how the total earning $v([d])$ should be distributed among such d players. Specifically, the Shapley value for each player $i \in [d]$ is defined as

$$\phi_v(i) = \sum_{U \subseteq [d] \setminus \{i\}} \frac{(|U|!(d - |U| - 1)!)}{d!} \left[v(U \cup \{i\}) - v(U)\right]. \tag{B.15}$$

For any target function $f \in \mathcal{F}_d$, input $\mathbf{x} \in \mathcal{X}$, and baseline $\underline{\mathbf{x}} \in \mathcal{X}$, we define a value function $v_{f,\mathbf{x},\underline{\mathbf{x}}}(U) \triangleq f(\mathbf{x}_U \sqcup \mathbf{x}_{\bar{U}})$ for any $U \in [d]$, where $\mathbf{x}_U \sqcup \mathbf{x}_{\bar{U}}$ is the spliced data point between $\mathbf{x}$ and $\underline{\mathbf{x}}$, defined in (B.3). Then the Shapley values $[\phi_{v_{f,\mathbf{x},\underline{\mathbf{x}}}}(i)]_{i\in[d]}$ can be viewed as an attribution method that provides the attribute scores for any f, $\mathbf{x}$, and $\underline{\mathbf{x}}$.

Now we prove that this attribution method based on Shapley valeus satisfies all four axioms (A1–A4) and the fidelity-to-control (P1), as stated in Proposition 2.

Proof. First, it is clear from the definition of Shapley values in (B.15) that $\phi_{v_{f,\mathbf{x},\underline{\mathbf{x}}}}(\cdot)$ satisfies linearity (A1) and implementation variance (A4). Since Shapley [B31] shows that for any value function v, the Shapley values $\phi_v(\cdot)$ satisfies that

$$\phi_v([d]) - \phi_v(\emptyset) = \sum_{i=1}^{d} \phi_v(i),$$

substituting our definition of the value function $\phi_{v_{f,\mathbf{x},\underline{\mathbf{x}}}}(\cdot)$ into the above equation yields

$$f(\mathbf{x}) - f(\underline{\mathbf{x}}) = \sum_{i=1}^{d} \phi_{v_{f,\mathbf{x},\underline{\mathbf{x}}}}(i),$$

which establishes the completeness (A2). For any $i \in [d]$ and $U \subseteq [d] \setminus \{i\}$, we have

$$v_{f,\mathbf{x},\underline{\mathbf{x}}}(U \cup \{i\}) - v_{f,\mathbf{x},\underline{\mathbf{x}}}(U) = f(\mathbf{x}_{U\cup\{i\}} \sqcup \mathbf{x}_{\bar{U}\setminus\{i\}}) - f(\mathbf{x}_U \sqcup \mathbf{x}_{\bar{U}})$$

Note that $\mathbf{x}_{U\cup\{i\}} \sqcup \mathbf{x}_{\bar{U}\setminus\{i\}}$ and $\mathbf{x}_U \sqcup \mathbf{x}_{\bar{U}}$ only potentially differ on the i-th dimension. If f does not depend on the i-th dimension of its input or $x_i = \underline{x}_i$ (which implies $\mathbf{x}_{U\cup\{i\}} \sqcup \mathbf{x}_{\bar{U}\setminus\{i\}} = \mathbf{x}_U \sqcup \mathbf{x}_{\bar{U}}$), then $f(\mathbf{x}_{U\cup\{i\}} \sqcup \mathbf{x}_{\bar{U}\setminus\{i\}}) = f(\mathbf{x}_U \sqcup \mathbf{x}_{\bar{U}})$ and thereby $\phi_{v_{f,\mathbf{x},\underline{\mathbf{x}}}}(i) = 0$. Thus, $\phi_{v_{f,\mathbf{x},\underline{\mathbf{x}}}}(\cdot)$ satisfies null player (A3) and fidelity-to-control (P1). □

Dyadic Gaussian Basis

Inspired by the dyadic interval bases used by Bao et al. [B7], we choose the basis functions $\{\psi_r(\cdot)\}_{r=1}^{R}$ to be the densities for a Gaussian family $\{\mathcal{N}(\mu_r, \sigma_r^2)\}_{r=1}^{R}$, which we term *dyadic Gaussian basis*. The means of dyadic Gaussian basis are given by

$$\mu_r = \begin{cases} 0, & r = 1, \\ L/2^{R-r}, & r = 2, \dots, R, \end{cases} \tag{B.16}$$

and the standard deviations by $\sigma_r = \max(\mu_r/3, \mu_2/3)$ for $r \in [R]$. This design of basis functions reflects a reasonable inductive bias that the CIFs should vary more smoothly as the time increases. As shown in Figure B.II.1 for an example of $L = 100$ and $R = 5$, the first a few bases, due to their small means and variances, capture the short-term effects, whereas the last several characterize the mid/long-term effects.

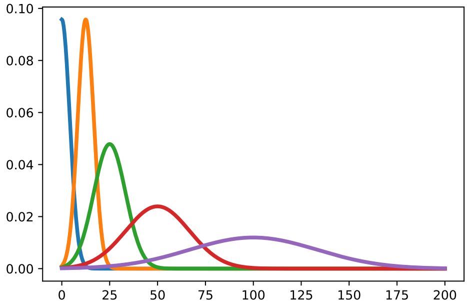


Fig. B.II.1: An example of dyadic Gaussian bases for $L = 100$ and $R = 5$. The first a few bases, due to their small means and variances, can capture the short-term effects, whereas the last several characterize the mid/long-term effects.

Proof of Proposition 3

Proof. We omit the index s in this proof for brevity. First, we rewrite $\tilde{Y}_{k,k'}$ as

$$\begin{aligned}\tilde{Y}_{k,k'} &= \sum_{i=1}^{n}\sum_{j=1}^{i}\mathbb{I}(k_j = k')A_j(f_k, \mathbf{x}_i, \underline{\mathbf{x}}_i)\\ &= \sum_{j=1}^{n}\mathbb{I}(k_j = k')\left[\sum_{i=j}^{n}A_j(f_k, \mathbf{x}_i, \underline{\mathbf{x}}_i)\right]\\ &= \sum_{j=1}^{n}\mathbb{I}(k_j = k')\left[\sum_{i=j}^{n}A_j(F_{k,i}, \mathbf{x}_n, \underline{\mathbf{x}}_n)\right],\end{aligned}$$

where in the step step, we replace f with F. Since $F_{k,i}$, i.e., $\int_{t_i}^{t_{i+1}}\lambda_k(t')dt$, does not depend on the events before the i-th event, with null player (A3), we have $A_j(F_{k,i}, \mathbf{x}_n, \underline{\mathbf{x}}_n) = 0$ for any $j < i$, which further implies that

$$\tilde{Y}_{k,k'} = \sum_{j=1}^{n}\mathbb{I}(k_j = k')\left[\sum_{i=1}^{n}A_j(F_{k,i}, \mathbf{x}_n, \underline{\mathbf{x}}_n)\right].$$

With linearity (A1), we have

$$\tilde{Y}_{k,k'} = \sum_{j=1}^{n}\mathbb{I}(k_j = k')A_j\left(\sum_{i=1}^{n}F_{k,i}, \mathbf{x}_n, \underline{\mathbf{x}}_n\right),$$

which establishes the formula.

□

B.III Additional Experimental Details

The settings for synthetic and real-world datasets.

We describe below the setup and preprocessing details for the five datasets that we consider in this paper. The statistics of these datasets are summarized in Table B.III.1.

- **Excitation.** This dataset was generated by a multivariate Hawkes process, whose CIFs are of the form:

  $$\lambda_k^*(t) = \mu_k + \sum_{i:t_i<t} \alpha_{k,k'}\beta_{k,k'} \exp[-\beta_{k,k'}(t-t_i)].$$

  We set $S = 1000$, $K = 10$, $n_s \sim \text{Poisson}(250)$, $\mu_k \sim \text{Uniform}(0, 0.01)$, and $\beta_{k,k} \sim \text{Exp}(0.05)$. To generate a sparse excitation weight matrix $\mathbf{A} \triangleq [\alpha_{k,k'}]_{k,k'\in[K]}$, we first selected all its diagonal entries and $M = 16$ random off-diagonal entries, then generated the values for these entries from $\text{Uniform}(0, 1)$, and finally scaled the matrix to have a spectral radius of 0.8.

- **Inhibition.** This dataset was generated by a multivariate self-correcting point process, whose CIFs take the form:

  $$\lambda_k^*(t) = \exp(\alpha_k t + \sum_{i:t_i<t} w_{k,k_i}).$$

  We chose $S = 1000$, $K = 10$, $n_s \sim \text{Poisson}(250)$, and $\alpha_k \sim \text{Uniform}(0, 0.05)$. To generated a sparse weight matrix $\mathbf{W} = [w_{k,k'}]_{k,k'\in[K]}$, we first selected all its diagonal entries and $M = 16$ random off-diagonal entries and further generated the values for these entries from $\text{Uniform}(-0.5, 0)$.

- **Synergy.** This dataset was generated by a proximal graphical event model (PGEM) [B8]. PGEM assumes that the CIF of an event type depends only on whether or not its parent event types (specified by a dependency graph) have occurred in the most recent history. We designed a local dependency structure that consists of five event types labeled as A–E. Among these event types, type E is the outcome and can be excited by the occurrence of type A, B, or C; type A and B, only when both occurred in the most recent history, would incur a large synergistic effect on type E; type C has an isolated excitative effect on type E and does not interacts with other event types; and finally, type D does not have any excitative effect and is introduced to complicate the learning task. The dependency graph, together with the corresponding time windows and intensity tables, illustrated in Figure B.III.1. To add more complexity to this dataset, we further replicated this local structure for another copy, leading to a total of $K = 10$ event types. We generated $S = 1000$ event sequences with a maximum time span of $T = 1000$.

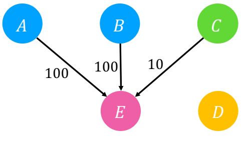
- **IPTV.** We obtained the dataset from[4]. We further normalized the timestamps into the days and splitted long event sequences so that the length of each sequence is smaller or equal to 1000.

[4] `https://github.com/HongtengXu/Hawkes-Process-Toolkit/tree/master/Data`

- **MT.** We downloaded the raw MemeTracker phrase data from[5]. We filtered the phrase data that occurred from 2008-08-01 to 2008-09-30 and from the top-100 website domains. We further normalized the timestamps into hours and filtered out those event sequences (i.e., phrase cascades) whose lengths are not in between 3 and 500.

Table B.III.1: Statistics for various datasets.

Dataset	S	K	# of events	Ground truth
Excitation	1,000	10	250,447	Weighted
Inhibition	1,000	10	250,442	Weighted
Synergy	1,000	10	178,338	Binary
IPTV	1,869	16	966,338	N/A
MT	382,014	100	3,419,399	Weighted



$$\lambda_A = \lambda_B = 0.01$$
$$\lambda_C = \lambda_D = 0.1$$
$$\lambda_E = 0.01 + \lambda_E(x_A, x_B) + \lambda_E(x_C)$$

$\lambda_E(x_A, x_B)$	$\neg x_A$	x_A
$\neg x_B$	0	0
x_B	0	0.1

	$\neg x_C$	x_C
$\lambda_E(x_C)$	0	0.03

Fig. B.III.1: The dependency graph, time windows, and intensity tables for the PGEM used in generating the Synergy dataset.

Implementation Details and Hyperparameter Configurations for Various Methods

For CAUSE, the $\mathrm{Enc}(\cdot)$ and the $\alpha(\cdot)$ were implemented by a single-layer GRU and a two-layer fully connected network with skip connections, respectively. The dimension of event type embeddings was fixed to 64, and the number of hidden units for GRU was set to be 64 for synthetic datasets and 128 for real-world datasets. The number of basis functions R and the maximum mean L were chosen by a rule of thumb such that μ_2 and μ_R are of the same scale as the 50th and the 99th percentiles of the inter-event elapsed times, respectively. The optimization was conducted by Adam with an initial learning rate of 0.001. A hold-out validation set consisting of 10% of the sequences of the training set was used for model selection; the model snapshot that attains the smallest validation loss was chosen. As events sequence lengths vary greatly on two real-world datasets, in constructing mini-batches for both training and inference, we adopted the bucketing technique to reduce the amount of wasted computation caused by padding. Finally, the line integral of IG, defined in (B.12), was approximated by 20 steps for MT and 50 steps for other datasets; a smaller number of steps, although

[5] `https://www.memetracker.org/data.html#raw`

may result in certain lose of accuracy, allows for a larger batch size and thus shorter execution time for attribution.

For the Hawkes process-based baselines—HExp, HSG, and NHPC—their implementation was provided by the package `tick` [B5]. The most relevant hyperparameters for each method were tuned by cross-validation.

As there is no publicly available codes for RPPN, we implemented it with our best effort. Its overall settings for architecture and optimization is similar to the ones for CAUSE.

Platform and Runtime

All experiments were conducted on a server with a 16-core CPU, 512G memory, and two Quadro P5000 GPUs. On the largest dataset, MT, the total runtime for CAUSE was less than 3 hours, including both training and computing the Granger causality statistic.

Qualitative Analysis on MT

Since there are too many event types in MT, instead of a heat map, we visualize the causality matrix as a graph and show in Figure B.III.2a and Figure B.III.2b the top-two communities of that graph, where the directed edges denote the estimated Granger causality between pairs of domains.[6] In Figure B.III.2a, the domain `news.google.com` centers in the middle and is pointed by many sites, which is unsurprising because Google News aggregates articles from other publishers and websites. Our method also correctly identifies other major "information-consuming" domains such as `bogleheads.org`, an active forum for investment-related Q&A. In Figure B.III.2b, the then very popular social networking website `blog.myspace.com` sits in the center of the community. Our method also identifies credible excitative relationships between the subdomains of `craigslist.org`, a mega-website that hosts classified, local advertisements.

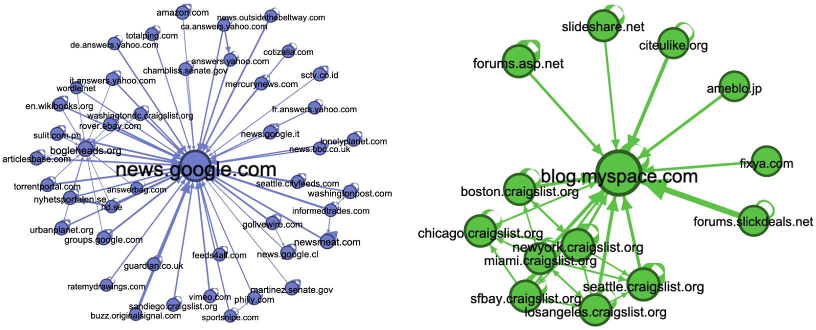
B.IV A Primer on Measure and Probability Theory

In this section, we review some of the basic definitions of in measure theory, which may help the understanding of the definition of Granger causality for multivariate point processes. Most of the content in this section were adapted based on primarily based on the Chapter 1 of [B15] and the Appendix 3 of [B11],

Let Ω be a set of "outcomes" and $\mathcal{F}$ a nonempty collection of subsets of Ω. The set $\mathcal{F}$ is σ-**algebra** of Ω, if it is closed under complement and countable unions; that is,

1. if $A \in \mathcal{F}$, then $\Omega \setminus A \in \mathcal{F}$, and
2. if $A_i \in \mathcal{F}$ is a countable sequence of sets, then $\cup_i A_i \in \mathcal{F}$.

[6] The graph visualization and community detection were performed using the software `Gephi`.



(a) MT (Community 1) **(b)** MT (Community 2)

Fig. B.III.2: The top-two communities of the estimated Granger causality statistic matrices on MT (Figure B.III.2a & B.III.2b). Better viewed on screen.

With these two conditions, it's easy to see that σ-algebra is also closed under arbitrary (possibly uncountable) intersections. From this, it follows that given a nonempty set Ω and a collection of $\mathcal{A}$ of subsets of Ω, there is a smallest σ-algebra containing $\mathcal{A}$; we denote such smallest σ-algebra by $\sigma(\mathcal{A})$. One particular σ-algebra is of particular interst—**Borel σ-algebra**; that is, the smallest σ-algebra containing all open sets in $\mathbb{R}^d$, denoted by $\mathcal{R}^d$. Specifically, let $\mathcal{S}_d$ be the empty set plus all sets of the form $(a_1, b_1] \times \cdots (a_d, b_d] \subset \mathbb{R}^d$, where $-\infty \le a_i < b_i \le \infty$, then $\mathcal{R}^d = \sigma(\mathcal{S}^d)$. The superscript d is dropped when $d = 1$.

A pair $(\Omega, \mathcal{F})$, in which Ω is a set and $\mathcal{F}$ is a σ-algebra of Ω, is called a **measurable space**. A **measure** defined on $(\Omega, \mathcal{F})$ is a nonnegative countably additive set function; that is a function $\mu : \mathcal{F} \mapsto \mathbb{R}$ with

1. $\mu(A) \ge \mu(\emptyset) = 0$ for all $A \in \mathcal{F}$, and
2. if $A_i \in \mathcal{F}$ is a countable sequence of disjoint sets, then

$$\mu(\bigcup_i A_i) = \sum_i \mu(A_i)$$

If $\mu(\Omega) = 1$, we call such a μ a **probability measure**. The triplet $(\Omega, \mathcal{F}, \mu)$ is called a **measure space**, and a **probability space** if μ is a probability measure.

Given a probability space $(\Omega, \mathcal{F}, \mu)$, a real-valued function X defined on Ω is said to be a **random variable** if for every Borel set $B \in \mathcal{R}$ we have $X^{-1}(B) \triangleq \{\omega : X(\omega) \in B\} \in \mathcal{F}$; in another words, X is **$\mathcal{F}$-measuable**. A **stochastic process** is a collection of random variables $\{X_i\}_{i \in \mathcal{I}}$ defined on a common probability space and indexed by a **index set** $\mathcal{I}$. In most cases, the index set can be positive numbers $\mathbb{N}_+$, or real line $\mathbb{R}_+$. A **filtration** is a sequence of σ-algebras, denoted by $\{\mathcal{F}_i\}_{i \in \mathcal{I}}$, if $\mathcal{F}_j \subseteq \mathcal{F}_i$ if $j \le i$ and $i, j \in \mathcal{I}$. Given a stochastic process $\{X_i\}_{i \in \mathcal{I}}$ defined on $(\Omega, \mathcal{F}, \mu)$, the **natural filtration** of $\mathcal{F}$ with respect to the process is given by

$$\mathcal{H}_i \triangleq \sigma\left(\{X_j^{-1}(B) | j \in \mathcal{I}, j \le i, B \in \mathcal{R}\}\right). \tag{B.17}$$

It is in a sense that the simplest filtration filtration available for studying the given: all information concerning the process, and only that information, is avaiable in the natural filtration. Thus, the natural filtration $\mathcal{H}_i$ can be often be viewed as the "**history**" of the subprocess $\{X_j\}_{j \leq i, j \in I}$. Note that sometimes the definition in (B.17) is simply written as $\mathcal{H}_i \triangleq \sigma\left(\{X_j | j \in \mathcal{I}, j \leq i\}\right)$.

A **point process** $\{T_i\}_{i \geq 1}$ is a real-valued stochastic process indexed on N_+ such that $T_i \leq T_{i+1}$ almost surely. Each random variable is generally viewed as the arrival timestamp of an event. For each point process, one can define a continuously indexed stochastic process associated with it called **counting process**, as $N(t) \triangleq \sum_{i=1}^{\infty} \mathbb{1}(T_i \leq t)$. From this definition, it is easily seen that every realization of a counting process is a càdlàg (i.e. right continuous with left limits) step function, and that a counting process $N(t)$ equivalently defines a point process, as one can recover the event timestamp by $T_i = \inf\{t \geq 0 : N(t) = i\}$. Due to this equivalence, the phrases point process and counting process, as well as their notation, $\{T_i\}_{i \in \mathbb{N}_+}$ and $N(t)$, are often used interchangably in the literature. A K-dimensional **multivariate point process (MPP)** is a coupling of K point/counting process $\mathbf{N}(t) = [N_1(t), N_2(t), \ldots, N_K(t)]$. A realization of a multivariate point process is a multi-type event sequence, $\{(t_i, k_i)\}_{i \in \mathbb{N}_+}$, where t_i indicates the event timestmap of the i-th event, and the k_i indicates which dimension the i-th event comes from (often interpreted as event type).

The most common way to define an MPP is through a set of **conditional intensity functions (CIFs)**, one for each event type. Specifically, let $\mathcal{H}(t) \triangleq \sigma(\{N_k(s) | k \in [K], s < t\})$ for any t be the natural filtration of MPP and let $\mathcal{H}(t-) \triangleq \lim_{s \uparrow t} \mathcal{H}(s)$ the CIF for event type k is defined as the expected instantaneous event occurrence rate conditioned on natural filtration, i.e.,

$$\lambda_k^*(t) \triangleq \lim_{\Delta t \downarrow 0} \frac{\mathbb{E}[N_k(t + \Delta t) - N_k(t) | \mathcal{H}(t)]}{\Delta t},$$

where the use of the asterisk is a notational convention to emphasize that intensity $\lambda_k^*(t)$ must be $\mathcal{H}(t)$-measurable for every t.

Finally, for any $\mathcal{K} \subseteq [K]$, denote by $\mathcal{H}_{\mathcal{K}}(t)$ the natural filtration expanded by the sub-process $\{N_k(t)\}_{k \in \mathcal{K}}$, i.e., $\mathcal{H}_{\mathcal{K}}(t) = \sigma(\{N_k(s) | k \in \mathcal{K}, s < t\})$, and further write $\mathcal{H}_{-k}(t) = \mathcal{H}_{[K] \setminus \{k\}}(t)$ for any $k \in [K]$. For a K-dimensional MPP, event type k is **Granger non-causal** for event type k' if $\lambda_{k'}^*(t)$ is $\mathcal{H}_{-k}(t)$-measurable for all t. This definition amounts to saying that a type k is Granger non-causal for another type k' if, conditioned on the history of events other than type k, the future $\lambda_{k'}^*(t)$ does not depend on the historical events of type k at any time. Otherwise, type k is said to be *Granger causal* for type k.

References

[B1] Massil Achab et al. "Uncovering causality from multivariate Hawkes integrated cumulants". In: *International Conference on Machine Learning*. PMLR. 2017, pp. 1–10.

[B2] Marco Ancona et al. "Towards better understanding of gradient-based attribution methods for deep neural networks". In: *arXiv preprint arXiv:1711.06104* (2017).

[B3] Andrew Arnold, Yan Liu, and Naoki Abe. "Temporal causal modeling with graphical granger methods". In: *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2007, pp. 66–75.

[B4] Sebastian Bach et al. "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation". In: *PLoS ONE* 10.7 (2015), pp. 1–46. ISSN: 19326203. DOI: 10.1371/journal.pone.0130140.

[B5] Emmanuel Bacry et al. "Tick: a Python library for statistical learning, with a particular emphasis on time-dependent modelling". In: *arXiv preprint arXiv:1707.03003* (2017).

[B6] Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate". In: *International Conference on Learning Representations*. 2015, pp. 1–15. arXiv: 1409.0473.

[B7] Yujia Bao et al. "Hawkes process modeling of adverse drug reactions with longitudinal observational data". In: *Machine learning for healthcare conference*. PMLR. 2017, pp. 177–190.

[B8] Debarun Bhattacharjya, Dharmashankar Subramanian, and Tian Gao. "Proximal graphical event models". In: *Advances in Neural Information Processing Systems* 31 (2018), pp. 8136–8145.

[B9] Kyunghyun Cho et al. "Learning phrase representations using RNN encoder-decoder for statistical machine translation". In: *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference* (2014), pp. 1724–1734. DOI: 10.3115/v1/d14-1179. arXiv: 1406.1078.

[B10] Rainer Dahlhaus and Michael Eichler. "Causality and graphical models in time series analysis". In: *Oxford Statistical Science Series* (2003), pp. 115–137.

[B11] Daryl J Daley and David Vere-Jones. *An Introduction to the Theory of Point Processes. Volume II: General Theory and Structure*. Springer, 2008.

[B12] Vanessa Didelez. "Graphical models for marked point processes based on local independence". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 70.1 (2008), pp. 245–264.

[B13] Cees Diks and Valentyn Panchenko. "A new statistic and practical guidelines for nonparametric Granger causality testing". In: *Journal of Economic Dynamics and Control* 30.9-10 (2006), pp. 1647–1669.

[B14] Nan Du et al. “Recurrent marked temporal point processes: Embedding event history to vector”. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2016, pp. 1555–1564.

[B15] Rick Durrett. *Probability: Theory and Examples*. 5th ed. Vol. 49. Cambridge university press, 2019.

[B16] Michael Eichler, Rainer Dahlhaus, and Johannes Dueck. “Graphical modeling for multivariate hawkes processes with nonparametric link functions”. In: *Journal of Time Series Analysis* 38.2 (2017), pp. 225–242.

[B17] Clive WJ Granger. “Investigating Causal Relations by Econometric Models and Cross-spectral Methods”. In: *Econometrica* 37.3 (Aug. 1969), p. 424. DOI: `10.2307/1912791`. URL: `http://links.jstor.org/sici?sici=0012-9682(196908)37:3%7B%5C%%7D3C424:ICRBEM%7B%5C%%7D3E2.0.CO;2-L%7B%5C%%7D5Cnhttp://www.jstor.org/stable/1912791%20http://www.jstor.org/stable/1912791?origin=crossref`.

[B18] Asela Gunawardana, Christopher Meek, and Puyang Xu. “A model for temporal dependencies in event streams”. In: *Advances in neural information processing systems* 24 (2011), pp. 1962–1970.

[B19] Alan G Hawkes. “Point spectra of some mutually exciting point processes”. In: *Journal of the Royal Statistical Society: Series B (Methodological)* 33.3 (1971), pp. 438–443.

[B20] Alan G Hawkes. “Spectra of some self-exciting and mutually exciting point processes”. In: *Biometrika* 58.1 (1971), pp. 83–90.

[B21] Craig Hiemstra and Jonathan D. Jones. “Testing for Linear and Nonlinear Granger Causality in the Stock Price-Volume Relation”. In: *The Journal of Finance* 49.5 (1994), pp. 1639–1664. ISSN: 15406261. DOI: `10.1111/j.1540-6261.1994.tb04776.x`.

[B22] Katerina Hlaváčková-Schindler et al. “Causality detection based on information-theoretic approaches in time series analysis”. In: *Physics Reports* 441.1 (2007), pp. 1–46.

[B23] Guido W Imbens and Donald B Rubin. *Causal inference in statistics, social, and biomedical sciences*. Cambridge University Press, 2015.

[B24] Valerie Isham and Mark Westcott. “A self-correcting point process”. In: *Stochastic Processes and Their Applications* 8.3 (1979), pp. 335–347.

[B25] Sarthak Jain and Byron C. Wallace. “Attention is not Explanation”. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. 2019, pp. 3543–3556. arXiv: `1902.10186`. URL: `http://arxiv.org/abs/1902.10186`.

[B26] Dixin Luo et al. “Multi-task multi-dimensional hawkes processes for modeling event sequences”. In: (2015).

[B27] Hongyuan Mei and Jason Eisner. "The Neural Hawkes Process: A Neurally Self-Modulating Multivariate Point Process". In: *NIPS*. 2017.

[B28] Judea Pearl. *Causality*. Cambridge university press, 2009.

[B29] Jonathan Schaffer. "The metaphysics of causation". In: (2003).

[B30] Sofia Serrano and Noah A. Smith. "Is Attention Interpretable?" In: *ACL*. 2019, pp. 2931–2951. DOI: 10.18653/v1/p19-1282. arXiv: arXiv:1906.03731v1.

[B31] Lloyd S Shapley. "A value for n-person games". In: *Contributions to the Theory of Games* 2.28 (1953), pp. 307–317.

[B32] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. "Learning Important Features Through Propagating Activation Differences". In: *International Conference on Machine Learning*. Apr. 2017. arXiv: 1704.02685. URL: http://arxiv.org/abs/1704.02685.

[B33] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. "Axiomatic attribution for deep networks". In: *International Conference on Machine Learning*. PMLR. 2017, pp. 3319–3328.

[B34] Jeremy C Weiss and David Page. "Forest-based point process for event prediction from electronic health records". In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer. 2013, pp. 547–562.

[B35] Shuai Xiao et al. "Learning Time Series Associated Event Sequences With Recurrent Point Process Networks". In: *IEEE Transactions on Neural Networks and Learning Systems* PP (2019), pp. 1–13. ISSN: 2162-237X. DOI: 10.1109/TNNLS.2018.2889776. URL: https://ieeexplore.ieee.org/document/8624540/.

[B36] Shuai Xiao et al. "Modeling the intensity function of point process via recurrent neural networks". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 31. 1. 2017.

[B37] Hongteng Xu, Mehrdad Farajtabar, and Hongyuan Zha. "Learning granger causality for hawkes processes". In: *International Conference on Machine Learning*. PMLR. 2016, pp. 1717–1726.

[B38] Kelvin Xu et al. "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention". In: (2015). ISSN: 19410093. DOI: 10.1109/72.279181. arXiv: 1502.03044. URL: http://arxiv.org/abs/1502.03044.

[B39] Matthew D. Zeiler and Rob Fergus. "Visualizing and understanding convolutional networks". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 8689 LNCS.PART 1 (Nov. 2014), pp. 818–833. ISSN: 16113349. DOI: 10.1007/978-3-319-10590-1_53. arXiv: 1311.2901.

Paper C

Towards Adversarial Phishing Detection

Thomas Kobber Panum, Kaspar Hageman, Jens Myrup Pedersen, René Rydhof Hansen

In proceedings of the
13th USENIX Workshop on Cyber Security Experimentation and Test (CSET 2020).

Abstract. *Over the recent decades, numerous evaluations of automated methods for detecting phishing attacks have been reporting stellar detection performances based on empirical evidence. These performances often neglect the adaptive behavior of an adversary seeking to evade detection, yielding uncertainty about their adversarial robustness. This work explores the adversarial robustness of highly influential and recent detection solutions, by assessing their common detection strategies. Following discussions of potential evasion techniques of these strategies, we present examples of techniques that enable evasion through imperceptible perturbations. In order to enable and improve future evaluations for adversarial robustness, a set of design guidelines is proposed.*

The layout has been revised.

C.1 Introduction

Protecting digital infrastructure against malicious attacks has become essential as computational systems increasingly store and exchange private information of interest. This has motivated the design of initiatives to make the systems under attack fundamentally more secure, causing adversaries to adopt attacks that circumvent these initiatives by exploiting the social behavior of users of the systems rather than the system itself. A type of these attacks, phishing has had increased frequency in recent years [C35, C1], and was described as the most widely adopted method for criminals to get unauthorized access to private networks in 2017.

Phishing attacks seek to exploit users by deceiving them, through some form of non-physical interaction, to release sensitive information for the benefit of the adversary [C19]. Since their discovery, over two decades ago, numerous research initiatives have tried to design solutions for automating identification of these attacks, in order to actively prevent them from reaching their targets [C36]. This has yielded solutions that demonstrate high accuracy for detecting these attacks, yet it has been highlighted that this performance is seemingly counter-intuitive and in contradicting to the observed volume of attacks [C30].

Following this, it was deemed that the used evaluation methodologies influenced this phenomenon, emphasizing that a very limited set of methods accounted for the adaptive behavior of adversaries in their evaluation. Thereby, these solutions could potentially be relying on attributions of phishing attacks that are exploitable, and potentially enable adversaries to find attacks that evade detection. The absence of these considerations serves as the main motivation behind this work, as stated performance might be causing a false sense of security, as evaluations are unable to reflect the true adversarial setting that a detection solution faces in practice.

Firstly, we cover related work that has addressed problems of evaluation methods for the non-adversarial setting, and address the situation of varying definitions of phishing attacks (Section C.2). Following this, we introduce a new terminology for phishing attacks, and their associated adversarial environment, that is independent of implementations and applications (Section C.3). We then introduce a set of axioms for phishing attacks, that encapsulate the functional properties of the attacks, and serve as abstract guidelines for selecting information to use for inferring attacks in a given context (Section C.4). Using the introduced terminology and axioms, we then assess two groups of existing work, highly influential and recent, by presenting four common strategies that the selected methods use for inferring attacks. The robustness of these strategies, to an adversary with an objective of creating phishing attacks that avoid detection, is then discussed and examples of perturbations that enable evasion are presented (Section C.5). Based on the knowledge obtained throughout the assessment, we present a set of design guidelines for designers of future detection solutions to adopt, in order to enhance robustness to adaptive attacks and enable evaluations of their solution (Section C.6). The contributions provided by this work can be summarized as:

- Propose a set of axioms for phishing attacks using a terminology that is independent of the application environment.

- Demonstrate and discuss the adversarial robustness of common detection strategies among highly influential and recent detection solutions.

- Following the assessment, put forward a set of design guidelines to enable and improve evaluations of adversarial robustness.

The implementation of the experiments for the conducted assessment, including reproductions of detection solutions and perturbation methods, is provided with open access at:

`https://github.com/tpanum/towards-adversarial-phishing-detection.`

C.2 Background

In 1995 criminals performed a large scale attack on users of the chat service America Online (AOL), that involved tricking users into sharing their passwords, as the criminals exploited software to impersonate staff members of AOL [C43, C37]. This incident is often associated with the origin of phishing attacks, and despite numerous initiatives to combat the attack, certain sources state that phishing attacks have never been more frequent [C17, C1, C34, C40]

Marchal et al. highlighted that this fact is counter-intuitive to the the fact that existing phishing detection solutions are reporting detection accuracy of over 99.9% [C42, C30]. They suggested that the cause might be design limitations of the methods, making them infeasible to deploy in real-world settings, or that evaluations of these methods are biased [C30].

Following this, they propose a systematic methodology with recommendations for future designers of detection solutions, that covers the topics of: data usage, evaluation metrics and temporal resilience. Within the scope of temporal resilience, they emphasize that these solutions are likely to see active attempts of evasion over time and deem that detection solutions should seek to become robust against these attempts. We refer to the ability for the solutions to resist these evasion attempts as adversarial robustness and cover it in greater detail in Section C.3. Marchal et al. finds that a limited set of methods have directly addressed their adversarial robustness, which could potentially cause them to be open to evasion in the setting with an adversary.

While phishing attacks have been studied thoroughly, a recent meta-analysis of scientific publications showed that a variety of definitions of phishing attacks exist across the literature [C23]. The cited analysis examined 536 publications containing 113 definitions, highlighting that definitions have varied globally across time and internally among research groups. Taking all of these definitions and their variations into account, Lastdrager et al. arrived at the following definition of phishing:

Definition 1 (from [C23]). Phishing is a scalable act of deception whereby impersonation is used to obtain information from a target.

Here, it is essential to clarify that scalable refers to the relative effort for an adversary to perform the attack. Lastdrager et al. states that this formulation encapsulates highly targeted attacks, often referred to as spear phishing, while disallowing face-to-face interaction and phone calls as valid measures for conducting phishing attacks. Impersonation as a measure for obtaining information can be exemplified by falsely claiming to be a policeman in order to see an identity card with sensitive information.

Despite this definition being able to express commonalities among historical definitions for various environments, it does not cover the properties of the environments hosting these attacks. We seek to establish more clarity of the problem of phishing detecting and address this gap, defining terminology capable of expressing the shared properties of these environments in conjunctions with elements of phishing detection. Following this, we decompose Definition 1 into a set of axioms of phishing attacks expressed using the established terminology. This will serve as language for clarifying the adversarial setting in which phishing detection solutions exist, and the challenges that arise when seeking adversarial robustness.

C.3 Terminology

Phishing attacks are known to exist across multiple environments, such as: instant messaging, websites, and emails. These environments share common properties that enable them to host phishing attacks. Throughout this section we seek to derive these properties, by establishing a terminology for phishing attacks, their related entities, and the interference conducted by detection solutions.

Definition 1 clearly states that the objective for the adversary conducting the phishing attack, is to obtain information. In order for this to be feasible, the environment must have some ability to exchange information across certain subjects. We denote the exchanged information as *messages* and the method of exchange as a *channel.* Each message has some *content* and a pair of subjects that reflect the *sender* and the *recipient*. We refer to an environment with these abilities as a *messaging environment*, which effectively serves as the foundation for phishing attacks to exist.

Attacks are carried by messages and are only functional when recipients receive and read them. This fact serves as the motivation for the design of phishing detection solutions, that seek to effectively filter out messages being sent that contain attacks, such that they never reach their recipient. Throughout this work we refer to these solutions as *detectors* and *detection solutions* interchangeably.

Naturally, the objective for these detectors is to categorize messages as benign or phishing with limited misclassifications. In order for a detector to categorize a message as phishing, it relies on a set of *attributes* that messages with phishing attacks are expected to have. We refer to these attributes as phishing attributes, for which each detector has a set of phishing attributes that effectively serves as the requirements for a message to be considered a phishing attack.

Let a detector be a function, that maps a message m to a set of phishing attributes from a set of candidate attributes $\mathcal{A}_\mathcal{D}$, such that $\mathcal{D}(m) \subseteq \mathcal{A}_\mathcal{D}$. The candidate set $\mathcal{A}_\mathcal{D}$ serves as a specification of features that a message must have to be considered phishing for the respective detector. A message m is considered to be a phishing attack if, and only if, the detector yields the entire set of phishing attributes from the candidate set, such that $\mathcal{D}(m) \supseteq \mathcal{A}_{\mathcal{D}}$. Note that this discrete formalization does not directly encapsulate applications relying on probabilistic approaches. However, ultimately these solutions are used for classification, effectively forcing them to be discrete functions as they predict discrete classes. See Section C.5. Importantly, this formulation intents to emphasize the scope of information used for inferring attacks, thus excluding more complex compositions of logic for simplicity purposes.

Given that detectors need to specify a candidate set of phishing attributes, the extent to which these definitions reflect the ground-truth set of attributes is uncertain, as obtaining completeness of this set is deeply philosophical. However, observations of successful attacks can serve as samples of empirical evidence of messages that contain the ground-truth set of attributes. Here we let the ability to obtain ground-truth categorizations of messages be expressed by an oracle function that maps a message m, to a set of phishing attributes from the ground-truth candidate set of phishing attributes $\mathcal{A}_{GT}$, such that $\mathcal{O}(m) \subseteq \mathcal{A}_{GT}$. Similar to detectors, a message m is a phishing attack if the oracle yields the ground-truth set of phishing attributes in its entirety.

Using this formulation, the natural objective for adversaries is to find a message m that satisfies the constraint $\mathcal{O}(m) \supseteq \mathcal{A}_{GT}$. When a detection solution is introduced into a previously undefended environment, it challenges the adversaries by requiring that the message must also be incorrectly classified by the detector. Thereby, the given message m must also satisfy $\mathcal{D}(m) \subset \mathcal{A}_{\mathcal{D}}$ to be a valid attack. We refer to this constraint as *circumvention*.

Occurrences of messages that satisfy circumvention is caused by *over-attribution*. Over-attribution is an inherit problem of the assumptions of phishing attacks adopted by a given detector. Concretely, it occurs when the set of phishing attributes used by the detector $\mathcal{A}_{\mathcal{D}}$ includes attributes that are unrelated to the ground-truth, such that $\mathcal{A}_{\mathcal{D}} \not\subseteq \mathcal{A}_{GT}$. Thereby, the detector relies on attributes of the message that are unrelated to its ability to carry an attack.

Conversely, when the detector's candidate set of attributes is a subset of the ground-truth candidate set, $\mathcal{A}_{\mathcal{D}} \subset \mathcal{A}_{GT}$, it can cause the detector to have overly defensive behaviour. We refer to this behaviour as *under-attribution*, and it can cause benign messages to falsely be considered attacks. Such situations are highly undesirable, as it can cause the detectors to become inapplicable for practical settings.

C.4 Axioms

The fact that the true attributes of phishing attacks $\mathcal{A}_{GT}$ are not directly observable, remains the core challenge for the process of designing detection solutions. Knowing this has driven the community to create numerous definitions [C23], which is undesirable for establishing common progress. As a measure to improve upon this situation, and as an attempt to establish a common perception of the problem of phishing, we propose a set of axioms. These axioms serve as abstractions of phishing attributes, which detection solutions should explicitly account for in their method of inference. We derive these axioms by examining and decomposing Definition 1.

Definition 1 initially states *Phishing is a scalable act [...]*, emphasizing that the *act* (of phishing) has to be *scalable*, giving name to the first decomposed axiom.

Axiom Scalable. Being scalable in the context of phishing attacks means that the method of carrying out the attack should be inexpensive. Importantly, this axiom does not address the volumes of attacks, and thereby the more targeted variations of email phishing, such as spear phishing, that also satisfy this axiom [C23].

Remark. Cost is context dependent, and the boundary for inexpensive is largely determined by a threat model for the respective environment. Examples of scalable attacks are phishing attacks conducted using inexpensive channels, such as email, as opposed to face-to-face communication which is considered expensive. For most practical environments, the use of a certain channel is often associated with a foreseeable and invariant cost. If such a cost is considered inexpensive, solutions acting in an environment, that uses solely such a channel, can implicitly satisfy this axiom.

Following this, additional specifications of the mentioned *act* are covered by: *[...] of deception whereby impersonation is used [...].* Here, impersonation is described as a method of deception, serving as a functional dependency of the mentioned *act*. We decompose this functional dependency of impersonation into an axiom of the same name.

Axiom Impersonating. An essential ability of phishing attacks, is the ability to deceive victims into believing that the sender's identity, of a message carrying an attack, is genuine and benign. Adversaries exploit identities across various abstractions of subject identities, varying from identities of specific subjects to mimicking a class of subjects. Exemplifying this in a context of websites, an adversary might seek directly replicate the appearance of a specific bank, e.g. Bank of America, or alternatively construct an appearance that resemble a generic identity of banks as a class.

Remark. This axiom implies that recipients of messages are to a certain degree responsible for validating the identities of senders. Additionally, their ability to do this must be imperfect in order for adversaries to exploit this axiom.

Lastly the definition states that the adversarial objective of the attack is: *[...] to obtain information from a target.* This objective suggests that the attack should induce some action that leads to the exchange of information, we capture this by the following axiom.

Axiom Inducive. As phishing attacks seek to exploit the users of a system rather than the system itself, it is necessary for the recipient of the attack message to conduct some action that allows for the attacker to fulfill his objective of obtaining information. This axiom encapsulates the fact that users must act upon interpreting the received message, that cause the adversary to obtain desired information.

C.5 Assessment of Existing Methods

Numerous initiatives from academia and industry have proposed methods for detecting phishing attacks without human intervention [C22]. These solutions have reported impressive performance measures based on historical observations of phishing attacks. It is often implicit or unknown to which extent these observations reflect the posterior measures that adversaries are likely to adopt for evading a detection solution, while maintaining functional attacks. This naturally yields uncertainty about to which extent the work accounts for evasion techniques. A study, that has analyzed lateral phishing attacks at large-scale, suggests that adversaries are willing to invest additional time into avoid being detected by their victims (opposed to a detection solution) [1] [C18].

Therefore, we seek to assess the ability of existing work to perform under these conditions, through an assessment of their adversarial robustness. For the assessment, we include methods of two categories, namely *highly influential* and *recent*. For highly influential methods, we aggregated the union of the ten most cited (or highest ranked) publications among the search results from a series of well-established search engines commonly used by the scientific community [2]. Throughout these searches we used search queries related to phishing detection [3]. Most of the publications within this group have impacted a network of succeeding solutions that either adopt a similar methodology or directly extend the given method. As a measure to explore if adversarial robustness has changed over time, and acknowledging that high citation counts favors older publications, we furthermore manually select a group of recently published methods that use novel methodologies for inference. The full list of methods selected for the assessment can be seen in Table C.5.1.

The selected methods are designed for a limited set of messaging environments, suggesting that these environments, namely the web and email, remain of highest interest. The web is an environment in which websites are exchanged across publishers and consumers, through servers and clients typically using the HTTP protocol as a channel. Email is an environment for exchange of text-based messages sent across the channels of SMTP, POP3 and IMAP. We speculate that this dominance of environments is largely caused by the large volume of phishing attacks in these environments, serving as a natural starting point for solutions seeking to detect attacks. Importantly, as suggested by the proposed axioms in Section C.4, attacks are not strictly limited to only exist in these environments.

Examining the selected methods is challenged by the fact that none of them provide a publicly accessible implementation. Additionally, reproducing an implementation of these methods is challenged by the the fact they often rely on private datasets or third-party components that are unrecoverable, such as older search engines. This inherently makes evaluations of adversarial robustness difficult, as perturbations for evasion requires access to the output of an implementation to validate their performance. Additionally, most of the methods do not explicitly state the entire set of attributes that their method relies on for classifying a message as a phishing attack. In respect to our

[1] Importantly, as more time is invested into individual attacks, fulfillment of the axiom of scalability decreases.

[2] Search engines: Web of Science, Scopus, IEEE Xplore, Google Scholar.

[3] The searches were conducted during September and October 2019 and used the queries "phishing detection" and "phishing classification".

	Influential															Recent		
	Fu et al. (2006) [C14]	Pan et al. (2006) [C32]	Zhang et al. (2007) [C45]	Fette et al. (2007) [C13]	Garera et al. (2007) [C15]	Dunlop et al (2010) [C11]	Aburrous et al. (2010) [C4]	Whittaker et al. (2010) [C42]	Le et al. (2010) [C24]	Xiang et al. (2011) [C44]	Islam et al. (2013) [C21]	Mohammed et al. (2013) [C31]	Chen et al. (2014) [C9]	Abdelhamid et al. (2014) [C2]	Chiew et al. (2015) [C26]	Mao et al. (2017) [C29]	Corona et al. (2017) [C10]	Abdelnabi et al. (2019) [C3]
VS	●												●		●	●	●	●
SM		●	●	●			●	●	●	●	●	●		●		●	●	●
RSC			●			●				●					●			
CI					●				●									
ENV	*W*	*W*	*W*	*E*	*W*	*W*	*W**	*W*	*W*	*W*	*E*	*W*	*W*	*W*	*W*	*W*	*W***	*W*

Table C.5.1: Methods selected for assessment and their identified strategies: Visual Similarity (VS), Statistical Modeling (SM), Reverse Search Credibility (RSC), and Channel Meta-information (CI). Messaging environments (*ENV*) cover email (*E*) and web (*W*), for which *W** is restricted to only e-banking websites and *W*** being phishing websites created using phishing toolkits.

notation, this effectively leaves the candidate set of phishing attributes, for a given detector, $\mathcal{A}_{\mathcal{D}}$ to be unknown. We address this by only assessing common attributes and methodologies that are thoroughly covered across multiple methods which expressed core ideas of the inference design.

These common attributes and ideas are expressed as *strategies*, for which we identified four among the selected methods: Visual Similarity (VS), Statistical Modelling (SM), Reverse Search Credibility (RSC), Channel Meta-information (CI). We seek to either show perturbations that enable evasions of the given strategy, or discuss fundamental problems that are likely to enable evasions.

Only two of the recently proposed methods [C10, C3] have been evaluated with respect to adversarial robustness. We attempt to reproduce one of the solutions and find evidence suggesting that the reported adversarial robustness is flawed. More details are contained within Section C.5.

Visual Similarity

Human perception is often the centrepiece of impersonation, known to be an axiom of phishing attacks, as it is exploited to deceive recipients into misinterpreting the identity of the sender. Certain solutions use a strategy that seeks to detect attacks by mimicking human perception of messages' visual identity and ideally is able to differentiate between messages that appear to have similar and unique visual identities. These similarity measures are then used in conjunction with appearances of known benign messages, to detect the visual similarity of future messages. If a message is similar to the set of known benign messages, while originating from a different source, then it is considered to be impersonation and for some methods this is the only attribute required to be considered a phishing attack. This strategy is expressed in the following derived attribute of phishing attacks:

Phishing Attribute 1. Sharing visual identity with an already observed benign message while originating from a different source.

Fu et al. implements this strategy by measuring the visual similarity using the Earth Mover's Distance for pixel intensity values of rendered websites [C14]. Chen et al. use Normalized Compression Distance for byte-representations of rendered websites' pixel intensities [C8, C9]. Mao et al. introduce a method that implements this attribute by comparing aggregations of a page's HTML elements, including respective CSS styles for each element, thus assuming similarities across these aggregations are identical to their rendered representations [C29, C28]. Corona et al. introduce a two-fold method for detecting attacks, for which one component uses image descriptors, in the form of Histogram of Oriented Gradients, and color histograms to measure for visual similarity among websites that host phishing attacks from phishing kits [C10].

We argue that Phishing Attribute 1 is a direct adoption of Axiom Impersonating, thereby serving as a useful attribution for inferring attacks. However, measuring the correctness of models, that seek to mimic human perception, is difficult and the inability to do so can lead to potential imperfections. These imperfections can potentially serve as an opportunity for exploitation that would enable adversaries to create attacks that circumvent detection. For demonstrative purposes, we employ a perturbation technique that yield seemingly imperceptible changes, and thereby are expected not to affect Phishing Attribute 1. However, this perturbation technique significantly changes the similarity values of NCD [C8, C9]. The technique is based on the fact that colors, in their binary representation, are completely distinct while color perceptions of humans are more fluid [C12].

Thereby, conducting color perturbations that are small in the perception space of humans still yield large distinctive changes in the binary changes. We exploit the HSL (hue, saturation, light) color space, for which changes in its continuous values reflect human perception better than similar changes conducted in the frequently used RGB (red, green, blue) color space. For the implementation of this color space, we use HSLuv [C7], and perturb images of websites by increasing saturation values by 1% and projecting the values to respective numerical limits. Let the similarity measure for two websites x_i and x_j be specified by NCD$(x_i, x_j) \to \mathbb{R} \in [0; 1]$, for which higher values reflect more similarity. Additionally, let a given appearance of a website be x, for which x' is the perturbed variant of x. The experiments showed that the perturbation technique dropped the similarity scores significantly, effectively causing NCD$(x, x') \approx 0$, for the most popular websites of the Tranco list [C25]. Given that phishing attacks often impersonate popular websites, this suggests that this perturbation technique could potentially lead to a consistent method for circumventing detection. Examples of perturbations, and the ability of the technique to influence the similarity score, can be seen in Figure C.5.1.

While our perturbation technique for NCD illustrates that specifically NCD is not adversarial robust, we argue that similar imperfections will exists for *any* method using the VS strategy until human perception have been effectively verified to be reproduced in a computational setting.

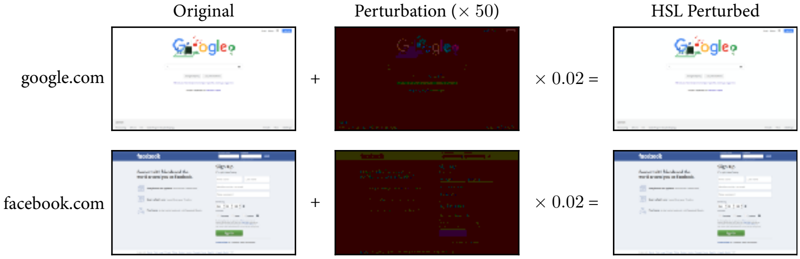


Fig. C.5.1: Visual appearances of the two most popular websites from the Tranco list [C25], being perturbed by tiny shifts in the HSL color space. For visualization purposes, the perturbations are enhanced by a multitude. These perturbations cause significant drops, NCD$(x, x') -$ NCD$(x, x) = -0.96 \pm 0.01$, in the visual similarity scores of NCD while being seemingly imperceptible.

Reverse Search Credibility

Search engines are a fundamental tool for finding and ranking information from the Internet using search queries of provided keywords. This strategy is based on the assumption that search engines only display trustworthy and credible websites in their search results. Consequently, the absence of given websites in search results can be attributed to a website being a phishing attack, expressed in the following derived attribute:

Phishing Attribute 2. Absence of a given website in the most relevant search results returned by querying search engines with a signature derived from the given website.

Zhang et al. and Xiang et al. implement this attribute by using the term frequency of infrequent words, from a corpus of websites, for creating signatures of individual websites used for querying the Google search engine [C45, C44]. Dunlop et al. expand upon this strategy by deriving the text of websites using Object Character Recognition (OCR), in order to be more robust against imperceptible text to image transformations of website content [C11]. Chiew et al. introduce a method extracting logos of websites, using them as signatures for reverse image searches [C26].

These studies demonstrate promising empirical evaluations using real-world data. However, we question their adversarial robustness as Phishing Attribute 2, suffer from some fundamental assumptions that can be exploited by an adversary. Firstly, we argue that the related attribute of the strategy does not align with any of the proposed axioms in Section C.3, thereby causing over-attribution. Secondly, the strategy puts forward strong assumptions on the algorithmic functionality of the search engines, which remain undocumented due to commercial interest, thus creating uncertainty about the functionality of crucial components for the design.

This uncertainty could potentially lead to exploits that allow for circumvention. Concretely, the solutions relying on the text documents of websites could suffer from injected rare words that would affect the extracted signature without altering the rendered interpretation for the user in the browser [C11]. Effectively this would cause the appearance of the website to remain unchanged while the signature could be altered to the desire of the adversary. Additionally, for solutions relying on OCR to extract signatures, it has been shown that OCR systems are vulnerable to imperceptible noise that gains the adversary some control of the set of recognized characters [C38].

For these reasons, we find this strategy insufficient for achieving adversarial robustness.

Channel Meta-information

Phishing attacks are carried by messages and require that these messages are exchanged in order to reach their target. The channel responsible for this exchange typically relies on user-controlled information that could potentially carry attributions of phishing attacks. This strategy is based on the assumption that attacks can be inferred purely based on meta-information of messages, and thereby independently of message content. Effectively, this restricts the set of allowed attributes to be within a certain domain of information that the channel exposes.

A common implementation of this strategy for websites, is to infer attacks solely based on similarities across the Uniform Resource Locators (URLs) [C15, C24]. Unknown URLs are then compared to a set of URLs from known benign websites, in order to infer attacks, as resemblance is a sign of impersonation, as derived in the following attribute.

Phishing Attribute 3. URLs resembling a URL from a known benign source.

Garera et al. implement this attribute by creating a statistical model that uses lexical information contained in URLs, in conjunction with Google PageRank information, in order to proactively prevent attacks prior to visiting websites [C15]. Le et al. propose a method that uses lexical features of URLs while being resistant to common obfuscation techniques used by adversaries for client-side inference of attacks [C24].

We argue that solutions adopting this strategy, using only content-independent-attributes, are prone to under-attribution. This stems from the inability to include attributions that align with Axiom Inducive. Additionally, we question to which extent Axiom Impersonating can be fully determined from content-independent attributes. However, we acknowledge that content-independent attributes can influence Axiom Impersonating.

Statistical Modeling

Deriving a set of concrete phishing attributes for a given messaging environment that prove to be useful and robust is the fundamental challenge of phishing detection. This difficulty has lead many researchers to learn these attributes using data of messages and statistical modeling. In particular, machine learning, a class of statistical modeling, has a significant presence in the selected methods. Machine learning is based on the popular approximation technique Empirical Risk Minimization [62], where the objective is to learn some probability distribution by minimizing the risk, typically represented as the weighted sum of some goodness of fit measure over the used data points. The learned probability distribution is then expressed as a model, that can be used for inference.

For phishing detection it is desired to learn some probability distribution of a given message containing a phishing attack, with respect to some set of information desired for inferring attacks. This information serves as a bound of information in which attributes of phishing can be learned by solving the stochastic optimization problem of minimizing the objective function. Naturally, the ability to approximate this probability distribution influences the chances of finding a distribution with low uncertainty to be useful for inference. Additionally, even if the ideal probability distribution is found, it might not even be useful for inference, if the information used for inference is seemingly uncorrelated with phishing attacks.

For certain models, including information that is strongly uncorrelated to the learning objective can hinder the ability of uncovering a useful probability distribution, thereby requiring manual labor for removing them prior to the learning process. However, progress over the last decade has allowed for more flexible models that are less sensitive to inclusion of uncorrelated information, such as Deep Neural Networks (DNNs). After the learning process is over for the statistical model, information that is associated with increased probability of the presence of attacks can effectively be addressed as phishing attributes of phishing attacks.

This strategy does thereby not impose the of use a specific attribute of phishing, it only expresses that attributes should be uncovered using patterns contained within data points.

Whittaker et al. use a random forest classifier that was trained using more than 3000 features in order to infer websites containing phishing attacks. Abdelnabi et al. exploit the potential of DNNs, by using a model named WhiteNet that has more than 100M trainable parameters for translating pixel intensity values of websites' appearances into a set of visual metrics.

Despite these methods showing promising results during evaluation, the ability to interpret the inference conducted by these models is difficult, thereby challenging the ability to validate and uncover the underlying phishing attributes. Validating these attributes is valuable, as the methods are prone to learn bias in high-dimensional spaces [C20]. In addition, approaches such as DNNs also suffer from a fundamental problem named adversarial examples, for which tiny perturbations of legitimate input cause unexpected large changes in the predictions of the model [C6, C41]. A perturbation is expressed as some noise $\delta \in \mathbb{R}^D$ for a given model input $x \in \mathbb{R}^D$, such that its perturbed variant is given by $x' = x + \delta$. Typically a threat model for these attacks is defined by a perturbation bound ϵ, such that for any given noise δ it must satisfy $\epsilon \leq \|\delta\|_p$ for some p-norm. To reduce this problem of these perturbations, and thereby making models more robust, it was proposed to use a training technique named adversarial training that includes perturbed input into the training process [38]. This technique has shown to reliably increase robustness.

WhiteNet uses a variation of this technique, to suit the training objective of metric learning, and improve the robustness of the model. The evaluation of the original model reports an accuracy of 65% (closest match) against adversarial examples generated using the Fast Gradient Sign Method (FGSM) [C16] ($\epsilon \leq 0.01$). This accuracy is considered, in contrast to other applications being attacked by adversarial examples [C5], relatively high in relation to the 81% accuracy for the original data. Using the adversarial training lifts this accuracy to 71% against adversarial examples.

This decreased performance led us to hypothesize that the reported high robustness could stem from two causes: the Siamese Neural Network (SSN) architecture used by WhiteNet has some inherent robust properties or the evaluation was performed incorrectly. Importantly, SSNs use a fundamentally different training procedure than typical machine learning classifiers, as the loss function takes triplets of data points as parameters.

As a measure to explore our hypotheses, we replicate the WhiteNet model using a similar data set of 37043 websites across 2449 domains, gathered using the Kraaler tool [C33]. Our implementation achieves a significantly lower accuracy of 24.6% against adversarial examples prior to adversarial training, with an increase to 30.8% after using adversarial training, as seen in Table C.5.2. During the generation of the attacks we adopted a larger batch size for attacks, as it have previously been discovered that the sampling of triplets can greatly influence the calculated loss during training [66]. Given that the loss function is also used for attacking, we hypothesize that a similar importance should be accounted for during attacks. We speculate that the increase in batch size, and thereby better sampling, enabled us to create substantially stronger at-

Model	Unperturbed	$\epsilon = 0.005$	$\epsilon = 0.01$
Traditional Training			
WhiteNet	81.0%	72.8%	62.5%
WhiteNet *(replica)*	87.8%	**30.0%**	**24.6%**
Adversarial Training			
WhiteNet	81.0%	79.0%	73.1%
WhiteNet *(replica)*	90.3%	**33.3%**	**30.8%**

Table C.5.2: Precision (closest match) for WhiteNet and our replica model across perturbations created using the FGSM attack for various threat models ϵ.

tacks. Following these results, we conclude that the reported robustness measures of WhiteNet are not representative for actual robustness towards adversarial examples.

C.6 Design Guidelines

Assessing existing detection solutions in Section C.5 highlighted problems related to the ability of evaluating adversarial robustness and attaining it, concretely: inaccessible implementations, implicit attributes, and over-attribution. As a measure to prevent future detection solutions from inheriting these problems, we propose three design guidelines: Accessible, Explicit Attributes, and Axiom Alignment.

Accessible. Adversaries are adaptive by nature, this fact should be reflected in the ability for the scientific community to be able to continuously evaluate solutions as new attacks emerge. Currently, most of the methods we have assessed do not provide widely available implementation, thus making it challenging to independently evaluate their performance. For certain methods it is infeasible to even reproduce the results, namely the data driven approaches that use private datasets in conjunction with not sharing the trained model weights.

To combat this phenomenon, we encourage that more authors make their methods easily accessible to the community. Ideally, this would be open access to the implementation, used throughout the original evaluations, or as a bare minimum ensure reproducibility. For methods relying on statistical modeling, this would include either making training data, or the found model weights, widely available. We deem that higher accessibility of solutions could contribute to the establishment of a community for perturbation techniques, that will prove useful for systematic evaluations of adversarial robustness.

Explicit Attributes. Defining phishing attacks has been shown to be difficult, causing a variety of definitions to exist [A16]. When designing a phishing detection solution, the fundamental task is to design some mechanism capable of quantifying attacks, typically based on some intuition of attacks. If the adaption of the intuition of attacks into concrete design decisions remain unclear, it can potentially disguise strong assumptions of attacks that can be violated and exploited in the adversarial setting.

We suggest that designers of detection solutions explicitly state, to the best of their ability, which information is considered as attribute(s) of a given message being a phishing attack for their respective domain. For statistical modeling, we suggest either using models for which causality can be directly studied at test-time or adopt methods for exploring the underlying attributions during inference [C27, C39]. This should be conducted to reduce the risk of the model inheriting bias from the underlying training data, causing undesired effects for the generalization performance. This guideline seeks to ensure that assumptions adopted throughout the design, namely the phishing attributes, become more explicit and thereby make the identification of potential cases of bias, over-attribution, or under-attribution, more effective.

Align with Axioms. Throughout the introduced terminology we have covered the consequences of having phishing attributes that are unaligned with true set of attributes, namely over-attribution and under-attribution. Unfortunately, the assessment highlighted that some of the common strategies adopted by the selected methods could be affected by these consequences. As a first measure for combating this phenomenon, we introduced a set of axioms of phishing attacks in Section C.4. Effectively, these axioms serve as abstract phishing attributes, independent of messaging environments, that one has to account for and transform into concrete information for inference in a given environment of application. Thereby, we suggest that designers explicitly document the relationship between their phishing attributes used for inference and the proposed axioms. Additionally, it must be ensured that the used set of phishing attributes cover the full set of axioms.

C.7 Conclusion

Detection solutions for identifying phishing attacks are continuously challenged by adversaries trying to adapt their attacks to evade detection. Across the influential and recent methods, most of these solutions do not account for this challenge in their evaluation, yielding uncertainty about their adversarial robustness. In order to clarify the conditions of this adversarial setting, we introduced a terminology that is independent of environment and application for respective methods. Based on a consensual definition of phishing, we presented three axioms of phishing attacks, that any detection solution should account for to avoid using incorrect attributes for inference. Following this, the adversarial robustness of highly influential and recent work were assessed by decomposing their methods of inference into a set of strategies. The ability to evade detection for the respective strategies was then discussed, and examples of perturbations that enabled evasions for certain methods were discovered. These findings let us to define a set of design guidelines for the community of phishing detection to adopt to both enable and improve evaluations of adversarial robustness.

References

[C1] Greg Aaron. *Phishing Acitivity Trends 1st Quarter 2018*. report. Anti-Phishing Working Group APWG, 2018. URL: https://docs.apwg.org/reports/apwg_trends_report_q1_2018.pdf.

[C2] Neda Abdelhamid, Aladdin Ayesh, and Fadi Thabtah. "Phishing detection based Associative Classification data mining". In: *Expert Systems with Applications* 41 (Oct. 2014), pp. 5948–5959. DOI: 10.1016/j.eswa.2014.03.019.

[C3] Sahar Abdelnabi, Katharina Krombholz, and Mario Fritz. *WhiteNet: Phishing Website Detection by Visual Whitelists*. 2019. arXiv: 1909.00300 [cs.CR].

[C4] Maher Aburrous et al. "Intelligent phishing detection system for e-banking using fuzzy data mining". In: *Expert systems with applications* 37.12 (2010), pp. 7913–7921.

[C5] Anish Athalye et al. *RobustML*. https://www.robust-ml.org/. Accessed: 2020-07-03. 2018.

[C6] Battista Biggio et al. "Evasion Attacks against Machine Learning at Test Time". In: *Lecture Notes in Computer Science* (2013), pp. 387–402. ISSN: 1611-3349. DOI: 10.1007/978-3-642-40994-3_25.

[C7] Alexei Boronine. *HSLuv*. https://www.hsluv.org/. Accessed: 2020-05-19. 2012.

[C8] Teh-Chung Chen, Scott Dick, and James Miller. "Detecting visually similar web pages: Application to phishing detection". In: *ACM Transactions on Internet Technology (TOIT)* 10.2 (2010), pp. 1–38.

[C9] Teh-Chung Chen et al. "An anti-phishing system employing diffused information". In: *ACM Transactions on Information and System Security (TISSEC)* 16.4 (2014), pp. 1–31.

[C10] Igino Corona et al. "DeltaPhish: Detecting Phishing Webpages in Compromised Websites". In: *CoRR* (2017). arXiv: 1707.00317.

[C11] Matthew Dunlop, Stephen Groat, and David Shelly. "Goldphish: Using images for content-based phishing analysis". In: *2010 Fifth international conference on internet monitoring and protection*. IEEE. 2010, pp. 123–128.

[C12] Mark D. Fairchild. "Color Appearance Models". In: *Color Appearance Models*. John Wiley & Sons, Ltd, 2013. Chap. 10, pp. 199–212. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781118653128.ch10.

[C13] Ian Fette, Norman Sadeh, and Anthony Tomasic. "Learning to detect phishing emails". In: *Proceedings of the 16th international conference on World Wide Web*. 2007, pp. 649–656.

[C14] Anthony Fu, Liu Wenyin, and Xiaotie Deng. "Detecting Phishing Web Pages with Visual Similarity Assessment Based on Earth Mover's Distance (EMD)". In: *Dependable and Secure Computing, IEEE Transactions on* 3 (Nov. 2006), pp. 301–311. DOI: 10.1109/TDSC.2006.50.

[C15] Sujata Garera et al. "A framework for detection and measurement of phishing attacks". In: *Proceedings of the 2007 ACM workshop on Recurring malcode*. 2007, pp. 1–8.

[C16] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. "Explaining and Harnessing Adversarial Examples". In: *International Conference on Learning Representations*. 2015.

[C17] Anti-Phishing Working Group. *Phishing Activity Trends Report 2016*. Feb. 2017. URL: `http://docs.apwg.org/reports/apwg_trends_report_q4_2016.pdf`.

[C18] Grant Ho et al. "Detecting and Characterizing Lateral Phishing at Scale". In: *28th USENIX Security Symposium (USENIX Security 19)*. Santa Clara, CA: USENIX Association, Aug. 2019, pp. 1273–1290. ISBN: 978-1-939133-06-9. URL: `https://www.usenix.org/conference/usenixsecurity19/presentation/ho`.

[C19] Markus Huber et al. "Towards Automating Social Engineering Using Social Networking Sites". In: *2009 International Conference on Computational Science and Engineering*. 2009. DOI: `10.1109/cse.2009.205`.

[C20] Andrew Ilyas et al. "Adversarial Examples Are Not Bugs, They Are Features". In: *Advances in Neural Information Processing Systems 32*. 2019, pp. 125–136.

[C21] Rafiqul Islam and Jemal Abawajy. "A multi-tier phishing detection and filtering approach". In: *Journal of Network and Computer Applications* 36.1 (2013), pp. 324–335.

[C22] Mahmoud Khonji, Youssef Iraqi, and Andrew Jones. "Phishing Detection: A Literature Survey". In: *IEEE Communications Surveys Tutorials* 15.4 (2013). DOI: `10.1109/SURV.2013.032213.00009`.

[C23] Elmer Lastdrager. "Achieving a Consensual Definition of Phishing Based on a Systematic Review of the Literature". In: *Crime Science* 3.1 (2014), p. 9. DOI: `10.1186/s40163-014-0009-y`.

[C24] Anh Le, Athina Markopoulou, and Michalis Faloutsos. "Phishdef: URL names say it all". In: *IEEE INFOCOM* (Sept. 2010). DOI: `10.1109/INFCOM.2011.5934995`.

[C25] Victor Le Pochat et al. "Tranco: A Research-Oriented Top Sites Ranking Hardened Against Manipulation". In: *Proceedings of the 26th Annual Network and Distributed System Security Symposium*. NDSS 2019. Feb. 2019. DOI: `10.14722/ndss.2019.23386`.

[C26] Kang Leng Chiew et al. "Utilisation of website logo for phishing detection". In: *Computers & Security* 54 (Aug. 2015). DOI: `10.1016/j.cose.2015.07.006`.

[C27] Scott M Lundberg and Su-In Lee. "A Unified Approach to Interpreting Model Predictions". In: *Advances in Neural Information Processing Systems 30*. 2017, pp. 4765–4774.

[C28] Jian Mao et al. "Detecting Phishing Websites Via Aggregation Analysis of Page Layouts". In: *Procedia Computer Science* 129 (2018), pp. 224–230. DOI: 10.1016/j.procs.2018.03.053.

[C29] Jian Mao et al. "Phishing Website Detection Based on Effective CSS Features of Web Pages". In: *Wireless Algorithms, Systems, and Applications*. Springer International Publishing, 2017, pp. 804–815. ISBN: 978-3-319-60033-8. DOI: 10.1007/978-3-319-60033-8_68.

[C30] Samuel Marchal and N. Asokan. "On Designing and Evaluating Phishing Webpage Detection Techniques for the Real World". In: *11th USENIX Workshop on Cyber Security Experimentation and Test (CSET 18)*. Baltimore, MD: USENIX Association, 2018.

[C31] Rami Mohammad, T. Mccluskey, and Fadi Thabtah. "Intelligent Rule based Phishing Websites Classification". In: *IET Information Security* 8 (Jan. 2013). DOI: 10.1049/iet-ifs.2013.0202.

[C32] Y. Pan and X. Ding. "Anomaly Based Web Phishing Page Detection". In: *2006 22nd Annual Computer Security Applications Conference (ACSAC'06)*. 2006, pp. 381–392.

[C33] T. K. Panum, R. R. Hansen, and J. M. Pedersen. "Kraaler: A User-Perspective Web Crawler". In: *2019 Network Traffic Measurement and Analysis Conference (TMA)*. 2019, pp. 153–160.

[C34] PhishLabs. *Phishing Trends and Intelligence Report 2019*. 2019. URL: https://info.phishlabs.com/hubfs/2019%20PTI%20Report/2019%20Phishing%20Trends%20and%20Intelligence%20Report.pdf.

[C35] John-Paul Power. *Latest Intelligence for March 2018*. Tech. rep. https://www.symantec.com/blogs/threat-intelligence/latest-intelligence-march-2018, Last accessed on 05-16-2020. Symantec, 2018.

[C36] Swapan Purkait. "Phishing Counter Measures and Their Effectiveness - Literature Review". In: *Information Management & Computer Security* 20.5 (2012), pp. 382–420. DOI: 10.1108/09685221211286548.

[C37] Koceilah Rekouche. *Early Phishing*. 2011. arXiv: 1106.4692 [cs.CR].

[C38] Congzheng Song and Vitaly Shmatikov. *Fooling OCR Systems with Adversarial Text Images*. 2018. arXiv: 1802.05385 [cs.LG].

[C39] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. "Axiomatic Attribution for Deep Networks". In: *Proceedings of the 34th International Conference on Machine Learning - Volume 70*. ICML'17. 2017, pp. 3319–3328.

[C40] Symantec. *Internet Security Threat Report*. report. Symantec, 2018. URL: https://www.symantec.com/content/dam/symantec/docs/reports/istr-23-2018-en.pdf.

[C41] Christian Szegedy et al. "Intriguing properties of neural networks". In: *International Conference on Learning Representations*. 2014.

[C42] Colin Whittaker, Brian Ryner, and Marria Nazif. “Large-Scale Automatic Classification of Phishing Pages”. In: *NDSS ’10*. 2010. URL: http://www.isoc.org/isoc/conferences/ndss/10/pdf/08.pdf.

[C43] Adam Wright, Skye Aaron, and David W. Bates. “The Big Phish: Cyberattacks Against U.S. Healthcare Systems"”. In: *Journal of General Internal Medicine* 31.10 (Oct. 2016), pp. 1115–1118. ISSN: 1525-1497. DOI: 10.1007/s11606-016-3741-z.

[C44] Guang Xiang et al. “Cantina+ a feature-rich machine learning framework for detecting phishing web sites”. In: *ACM Transactions on Information and System Security (TISSEC)* 14.2 (2011), pp. 1–28.

[C45] Yue Zhang, Jason I. Hong, and Lorrie Faith Cranor. “CANTINA: a content-based approach to detecting phishing web sites”. In: *Proceedings of the 16th International Conference on World Wide Web (WWW 2007)*. Jan. 2007, pp. 639–648. DOI: 10.1145/1242572.1242659.

Paper D

Exploring Adversarial Robustness of Deep Metric Learning

Thomas Kobber Panum, Zi Wang, Pengyu Kan, Earlence Fernandes, Somesh Jha

Pre-print and in review

Abstract. *Deep Metric Learning (DML), a widely-used technique, involves learning a distance metric between pairs of samples. DML uses deep neural architectures to learn semantic embeddings of the input, where the distance between similar examples is small while dissimilar ones are far apart. Although the underlying neural networks produce good accuracy on naturally occurring samples, they are vulnerable to adversarially-perturbed samples that reduce performance. We take a first step towards training robust DML models and tackle the primary challenge of the metric losses being dependent on the samples in a mini-batch, unlike standard losses that only depend on the specific input-output pair. We analyze this dependence effect and contribute a robust optimization formulation. Using experiments on three commonly-used DML datasets, we demonstrate* 5–76 *fold increases in adversarial accuracy, and outperform an existing DML model that sought out to be robust.*

The layout has been revised.

D.1 Introduction

Many machine learning (ML) tasks rely on ranking entities based on the similarities of data points in the same class. Deep Metric Learning (DML) is a popular technique for such tasks, particularly for applications involving test-time inference of classes that are not present during training (e.g. zero-shot learning). Example applications of DML include person re-identification [D15], face verification [D25, D8], phishing detection [D1], and image retrieval [D38, D23]. At its core, DML relies on state-of-the-art deep learning techniques for training models that output lower-dimensional semantic feature embeddings from high-dimensional inputs. Points in this embedding space cluster similar inputs together while dissimilar inputs are far apart.

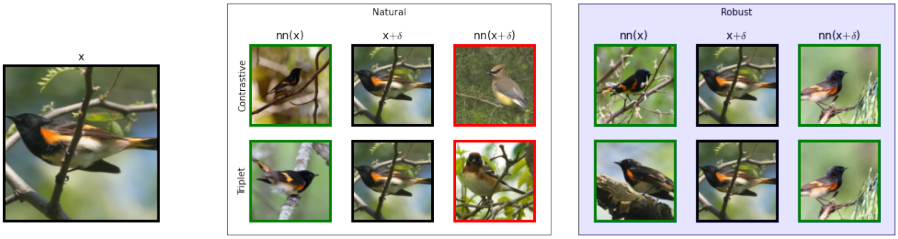


Fig. D.1.1: Example of inference of a naturally-trained DML model and robustly trained variant on the CUB200-2011 dataset. Each model infers the class of the natural data point $\mathbf{x}$, and its perturbed $\mathbf{x} + \delta$ counterpart, using the class of the nearest anchor $nn(\cdot)$. Green and red borders indicate correct- (same class) and incorrect inference, respectively. Both models infer the natural input correctly, however, the naturally-trained DML model fails to infer the adversarial perturbed input correctly.

Traditional deep learning classifiers are vulnerable to adversarial examples [D30, D3] — inconspicuous input changes that can cause the model to output attacker-desired values. Few studies have addressed whether DML models are similarly susceptible towards these attacks, and the results are contradictory [D1, D22]. Given the wide usage of DML models in diverse ML tasks, including security-oriented ones, it is important to clarify their susceptibility towards attacks and ultimately address their lack of robustness. We investigate the vulnerability of DML towards these attacks and address the open problem of training DML models using robust optimization techniques [D2, D19].

A key challenge in robust training of DML models concerns the so-called *metric losses* [D38, D35, D6, 55]. Unlike loss functions used in typical deep learning settings, the metric loss for a single data point is interdependent on other data points. For example, the widely-used triplet loss requires three input points: an anchor, a positive sample similar to the anchor, and a negative sample dissimilar to the anchor. For training, this interdependence impacts the effectiveness of learning [D38]. Thus, several works have identified sampling strategies that turn mini-batches into tuples or triplets to ensure that training remains effective [D25, D41, D40]

This *interdependence* between data points of metric losses poses a challenge for creating effective adversarial perturbations, as these are typically computed by approximating the inconspicuous noise that maximizes the loss for the specific data point.

Consequently, as adversarial training depends on this ability during training, it has to remain efficient in order to reduce the additional computation as natural training procedures for certain DML models are already considered resource intensive [D24]. Additionally, metric losses are sensitive to samples with high levels of noise during training, that can cause training to reach an undesired local minima [D38]. Adversarial perturbations are effectively noise, and thus adversarial training procedure for DML models has to account for this sensitivity.

We systematically approach the above challenges and contribute a robust training objective formulation for DML models by considering the two widely-used metric losses — contrastive and triplet loss. An example of the influence the robust training objective on inference is shown in Figure D.1.1. Our key insight is that during an inference-time attack, adversaries seek to perturb data points such that the intra-class distance maximize, and thus this behavior needs to be accounted for during training to improve robustness. Recent work has attempted to train robust DML models but has not considered the dependence and sensitivity to sampling [D1]. When we subject these models to our attack techniques, we find that their robustness is actually less than what is reported.

Prior work on traditional classifiers have established a connection between Lipschitz constant and robustness [D14]. Our intuition is the adversarial training of lead to a lower Lipschitz constant of the deep metric embedding. We explore this further in supplementary materials.

Contributions.

- We contribute a principled robust training framework for DML models by considering the dependence of metric losses on the other data points in the mini-batch and the sensitivity to sampling.

- We experiment with naturally-trained DML models across three commonly-used datasets for DML (CUB200-2011, CARS196, SOP) and show that they have poor robustness — their accuracy (R@1) drops from 59.1% (or more) to 4.0% (or less) when subjected to PGD attacks of the proposed attack formulation.

- Using our formulation for adversarial training, DML models reliably increase their adversarial robustness, outperforming prior work. For $\ell_\infty(\epsilon = 0.01)$, we obtain an adversarial accuracy of 53.6% compared to the state-of-the-art natural accuracy baseline of 71.8% for the SOP dataset (in terms of R@1 score, a common metric in DML to assess the accuracy of models). Furthermore, the resulting robust model accuracies are largely unaffected for natural (unperturbed) samples.

D.2 Related Work

Deep Metric Learning. DML is a popular technique to obtain semantic feature embeddings with the property that similar inputs are geometrically close to each other in the embedding space while dissimilar inputs are far apart [D24]. DML losses involve pairwise distances between embeddings [D4]. Examples include contrastive loss [D12], triplet loss [D25], Neighborhood Component Analysis [D10], and various extensions of these losses [D27, D35, D42]. Throughout this work, we refer to these types of losses as *metric losses*. Recent surveys [D24, D21] highlight that performance of newer metric losses are lesser than previously reported. Thus, we focus on the two established metric losses — contrastive and triplet — as they are widely used and have good performance.

Adversarial Robustness. Since early work in the ML community discovered adversarial examples in deep learning models [D30, D3], a big focus has been to train adversarially-robust models. We focus on robust optimization-based training that utilizes a saddle-point formulation (min-max) [D2, 38]. To the best of our knowledge, training DML models using robust-optimization techniques has not been thoroughly studied, and only recently has work begun in this area [D1].

Using the Generative Adversarial Network architecture [D11], Duan et al. [D9] create a framework that uses generative models during training to derive hard negative samples from easy negatives. They focus on improving the effectiveness of *naturally* training DML models rather than obtaining adversarial robustness, which is our focus.

Recent studies have shown that metric losses can function as a supplementary regularization method that enhances adversarial robustness of deep neural network classifiers (e.g., CNNs) [D20, D18]. However, these studies are not applicable to training robust DML models, as they do not solve the problem of dependence between data points due to the use of metric losses. We propose a principled framework for robustly training DML models that accounts for this problem.

D.3 Towards Robust Deep Metric Models

First, we describe some basic machine learning (ML) notation and concepts required to describe our algorithm. We assume a data distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}$ is the sample space and $\mathcal{Y} = \{y_1, \cdots, y_L\}$ is the finite space of labels. Let $\mathcal{D}_\mathcal{X}$ be the marginal distribution over $\mathcal{X}$ induced by $\mathcal{D}$ [1]. Given $Y \subseteq \mathcal{Y}$ we define $\mathcal{D}_Y$ to be the measure of the subsets of $\mathcal{X} \times Y$ induced by $\mathcal{D}$. For $y \in \mathcal{D}$, $\mathcal{D}_y$ and $\mathcal{D}_{-y}$ denote the measures $D_{\{y\}}$ and $D_{\mathcal{Y}\setminus\{y\}}$, respectively.

In the *empirical risk minimization (ERM)* framework we wish to solve the following optimization problem:

$$\min_{w \in \mathcal{H}} \; E_{(\mathbf{x},y)\sim\mathcal{D}} \; l(w, \mathbf{x}, y) \tag{D.1}$$

In the equation given above $\mathcal{H}$ is the hypothesis space and l is the loss function. We will denote vectors in boldface (e.g. $\mathbf{x}$, $\mathbf{y}$). Since the distribution is usually unknown, a learner solves the following problem over a data set $S = \{(\mathbf{x}_1, y_1), \cdots .(\mathbf{x}_n, y_n)\}$ sampled from the distribution $\mathcal{D}$.

$$\min_{w \in \mathcal{H}} \; \frac{1}{n} \sum_{i=1}^{n} l(w, \mathbf{x}_i, y_i) \tag{D.2}$$

[1] The measure of set $Z \subseteq \mathcal{X}$ in distribution $\mathcal{D}_\mathcal{X}$ is the measure of the set $Z \times \mathcal{Y}$ in distribution $\mathcal{D}$.

Once we have solved the optimization problem given above, we obtain a $w^* \in \mathcal{H}$ which yields a classifier $F\colon \mathcal{X} \to \mathcal{Y}$ (the classifier is usually parameterized by w^*, but we will omit it for brevity).

Deep Metric Models

The goal of *deep metric learning (DML)* is to create a *deep metric model* f_θ is function from $\mathcal{X}$ to S^d, where $\theta \in \Theta$ is a parameter and S^d is an unit sphere in $\mathbb{R}^d$ (i.e. $\mathbf{x} \in S^d$ iff $\|\mathbf{x}\|_2 = 1$). Since deep metric models embed a space $\mathcal{X}$ (which can itself be a metric space) in another metric space, we also sometimes refer to them deep embedding. Frequently, deep metric models use very different loss functions than typical classification networks described previously. Next we discuss two kinds of loss functions – contrastive and triplet. Let $S = \{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_n, y_n)\}$ be a dataset drawn from $\mathcal{D}$. A *contrastive* loss function $l_c(\theta, (\mathbf{x}, y), (\mathbf{x}_1, y_1))$ of labeled samples from $\mathcal{X} \times \mathcal{Y}$ and is defined as:

$$1_{y=y_1}\ d_\theta(\mathbf{x}, \mathbf{x}_1) + 1_{y \neq y_1}\ [\alpha - d_\theta(\mathbf{x}, \mathbf{x}_1)] \tag{D.3}$$

In the equation given above, 1_E is an indicator function for event E (1 if event E is true and 0 otherwise), and $d_\theta(\mathbf{x}, \mathbf{x}_1)$ is $\sqrt{\sum_{j=1}^{d} (f_\theta(\mathbf{x})_j - f_\theta(\mathbf{x}_1)_j)^2}$, the ℓ_2 distance in the embedding space. The goal of the contrastive loss function is to reduce the distance in the embedding space between two samples with the same label, and analogously increase the distance in the embedding space between the two samples with different labels. A *triplet* loss function l_t is defined over three $l_t(\theta, (\mathbf{x}, y), (\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2))$ labeled samples and is defined as follows:

$$1_{y=y_1}\ 1_{y \neq y_2}\ [d_\theta(\mathbf{x}, \mathbf{x}_1) - d_\theta(\mathbf{x}, \mathbf{x}_2) + \alpha]_+ \tag{D.4}$$

In the equation given above $[x]_+$ is $\max(x, 0)$. In order for the expression to be non-zero $(\mathbf{x}_1, y_1)$ has to have the same label as $(\mathbf{x}, y)$, and $(\mathbf{x}_2, y_2)$ has to have a different label as $(\mathbf{x}, y)$.

Attacks on Deep Metric Models

Assume that we have learned a deep embedding network with parameter $\theta \in \Theta$ using one of the loss functions described above. Next we describe how the network is used. Let $A = \{(\mathbf{a}_1, c_1), \cdots, (\mathbf{a}_m, c_m)\}$ be a reference or test dataset (e.g. a set of faces along with their label). A is distinct from the dataset S used during training time. Suppose we have a sample $\mathbf{z}$ and let $k(A, \mathbf{z})$ be the index that corresponds to $\arg\min_{j \in \{1, \cdots, m\}} d_\theta(\mathbf{a}_j, \mathbf{z})^2$[2]. We predict the label of $\mathbf{z}$ as $lb(A, \mathbf{z}) = c_{k(A,\mathbf{z})}$(we will use the functions $k(.,.)$ and $lb(.,.)$ throughout this section).

Next we describe test-time attacks on a deep embedding with parameter θ. Let $\mathbf{z} \in \mathcal{X}$. *Untargeted attack* on $\mathbf{z}$ can be described as follows (we want the perturbed

[2] In case one or more anchors share the minimal distance to $\mathbf{z}$, the tie is broke by a random selection among these anchors.

point to have a different label than before):

$$\begin{array}{l}\min_{\delta\in\mathcal{X}}\ \ \mu(\delta)\\ \textit{such that } lb(A,\mathbf{z})\neq lb(A,\mathbf{z}+\delta)\end{array}\tag{D.5}$$

Targeted attack (with a target label $t \neq lb(A, \mathbf{z})$) can be described as follows (we desire to the predicted label of the perturbed point to be a specific label):

$$\begin{array}{l}\min_{\delta\in\mathcal{X}}\ \ \mu(\delta)\\ \textit{such that } lb(A,\mathbf{z}+\delta)=t\end{array}\tag{D.6}$$

In the formulations given above we assume that $\mathcal{X}$ is a metric space with μ a metric on $\mathcal{X}$ (e.g. $\mathcal{X}$ could $\mathbb{R}^n$ with usual norms, such as ℓ_∞, ℓ_1, or ℓ_p (for $p \geq 2$)). Any algorithm that solves the optimization problem described above leads to a specific attack on deep metric models.

Robust Deep Metric Models

Let $S = \{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_n, y_n)\}$ be a dataset drawn from distribution $\mathcal{D}$. For a sample $(\mathbf{x}_i, y_i)$ where $1 \leq i \leq n$ we define the following surrogate loss function $\hat{l}(\theta, (x_i, y_i), S)$ for the contrastive loss function l_c :

$$\hat{l}(\theta,(x_i,y_i),S)=\frac{1}{n}\sum_{j=1}^{n} l_c(\theta,(\mathbf{x}_i,y_i),(\mathbf{x}_j,y_j))\tag{D.7}$$

Similarly, for the triplet loss function l_t we can define the following surrogate loss function $\hat{l}(\theta, (x_i, y_i), S)$:

$$\frac{1}{n_{y_i}n_{y_i}^{-}}\sum_{j=1}^{n_{y_i}}\sum_{k=1}^{n_{y_i}^{-}} l_t(\theta,(\mathbf{x}_i,y_i),(\mathbf{x}_j,y_j),(\mathbf{x}_k,y_k))\tag{D.8}$$

Let S_y and S_{-y} be defined as the following sets: $\{(\mathbf{x}, y) \mid (\mathbf{x}, y) \in S\}$ and $\{(\mathbf{x}, y') \mid (\mathbf{x}, y') \in S \text{ and } y' \neq y\}$. In the equation given above the sizes of the sets S_y and S_{-y} are denoted by n_y and n_y^-, respectively.

Having defined the surrogate loss function $\hat{l}$ the learner's problem can be defined as:

$$\min_{\theta\in\Theta}\frac{1}{n}\sum_{i=1}^{n}\hat{l}(\theta,(\mathbf{x}_i,y_i),S)\tag{D.9}$$

Recall that the learner's problem for the usual classification case is:

$$\min_{w\in\mathcal{H}}\ \frac{1}{n}\sum_{i=1}^{n} l(w,\mathbf{x}_i,y_i)\tag{D.10}$$

Note that in the classification case the loss function $l(w, \mathbf{x}_i, y_i)$ of a sample $(\mathbf{x}_i, y_i)$ does not depend on the other samples in the dataset S. However, in the deep metric model case the surrogate loss function $\hat{l}(\theta, (\mathbf{x}_i, y_i), S)$ for a sample $(\mathbf{x}_i, y_i)$ depends on the rest of the data set S (see the equations for $\hat{l}$) This is the main difference between the embedding and classification scenarios.

Formulation 1. Let $B_p(\mathbf{x}, \epsilon)$ denote the ϵ-ball around the sample $\mathbf{x}$ using the ℓ_p-norm. The straightforward robust formulation is given in the equation below.

$$\min_{\theta \in \Theta} \max_{(\mathbf{z}_1, \cdots, \mathbf{z}_n) \in \prod_{j=1}^{n} B_p(\mathbf{x}_j, \epsilon)} \frac{1}{n} \sum_{i=1}^{n} \hat{l}(\theta, (\mathbf{z}_i, y_i), S) \tag{D.11}$$

In the formulation given above, all samples are adversarially perturbed at the same time (note that the $\max$ is outside the summation). Therefore, this formulation is not convenient for current training algorithms, such as SGD and ADAM. This is because the entire dataset S has to be perturbed at the same time. Moreover, this formulation is not conducive to various sampling strategies used in training of deep metric models.

Formulation 2. In this formulation we push the $\max$ inside the sum so that each term can be individually processed. This is especially useful for adversarial training because each tuple or triple can be perturbed separately. Our formulation will be indexed by r ($r \in \{1, 2\}$ for contrastive loss and $r \in \{1, 2, 3\}$ for triplet loss). Intuitively, r denotes what component of the tuple of triple is being perturbed. We define operator $\max(r, \epsilon, \theta)$ which perturbs the r-th component in an ϵ ball to maximize the loss. For example, $\max(r, \epsilon, \theta)$ for $((\mathbf{x}, y), (\mathbf{x}_1, y_1))$ is defined as:

$$\max_{\mathbf{z} \in B_p(\mathbf{x}_1, \epsilon)} l_c(\theta, (\mathbf{x}, y), (\mathbf{z}, y_1)) \tag{D.12}$$

Now we can define $\hat{l_r}$ for the contrastive case as:

$$\hat{l_r}(\theta, (x_i, y_i), S) = \frac{1}{n} \sum_{j=1}^{n} \max(r, \epsilon, \theta)((\mathbf{x}_i, y_i), (\mathbf{x}_j, y_j))$$

The equation for the triplet loss is similar. Now the entire minimization problem becomes.

$$\min_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^{n} \hat{l_r}(\theta, (\mathbf{z}, y_i), S) \tag{D.13}$$

Formulation 3. Our third formulation adds a regularizer which enforces the following informal constraint: if $\mathbf{x}$ changes a bit, the distance in the embedding space does not change too much.

$$\min_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^{n} [\hat{l}(\theta, (\mathbf{x}_i, y_i), S) \; + \; \lambda \max_{\mathbf{z} \in B_p(\mathbf{x}_i, \epsilon)} d_\theta(\mathbf{z}, \mathbf{x}_i)] \tag{D.14}$$

These robust optimization formulations follow the common notion of robustness from robust optimization [D2], thus given an algorithm for solving one of the robust optimization formulations, leads to a robust model.

Attack Algorithm

We will focus on untargeted attacks because our main goal is to use these algorithms to robustify embeddings using adversarial training. Recall that $d_\theta(\mathbf{x}, \mathbf{x}_1)$ is the l_2 distance between $f_\theta(\mathbf{x})$ and $f_\theta(\mathbf{x}_1)$. The gradient $\nabla_\mathbf{x} d_\theta(\mathbf{x}, \mathbf{x}_1)$ of $d_\theta(\mathbf{x}, \mathbf{x}_1)$ with-respect-to (wrt) to $\mathbf{x}$ is given by:

$$\frac{1}{d_\theta(\mathbf{x}, \mathbf{x}_1)} (f_\theta(\mathbf{x}) - f_\theta(\mathbf{x}_1))^T \cdot \nabla_\mathbf{x} f_\theta(\mathbf{x}) \tag{D.15}$$

A similar expression can be written for $\nabla_{\mathbf{x}_1} d_\theta(\mathbf{x}, \mathbf{x}_1)$.

Consider the contrastive loss l_c on a tuple $(\mathbf{x}, \mathbf{x}_1)$.

$$1_{y=y_1}\ d_\theta(\mathbf{x}, \mathbf{x}_1) + 1_{y \neq y_1}\ [\alpha - d_\theta(\mathbf{x}, \mathbf{x}_1)] \tag{D.16}$$

The gradient of the contrastive loss $\nabla_{\mathbf{x}_1} l_c(\theta, (\mathbf{x}, y), (\mathbf{x}_1, y_1))$ wrt $\mathbf{x}_1$ is shown below:

$$1_{y=y_1}\ \nabla_{\mathbf{x}_1} d_\theta(\mathbf{x}, \mathbf{x}_1) - 1_{y \neq y_1}\ \nabla_{\mathbf{x}_1} d_\theta(\mathbf{x}, \mathbf{x}_1) \tag{D.17}$$

Similar to contrastive loss, we can define gradients of $l_t(\theta, (\mathbf{x}, y), (\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2))$ wrt $\mathbf{x}$, $\mathbf{x}_1$, or $\mathbf{x}_2$.

Once we can compute the gradients of the loss, we can readily adapt attack algorithm, such as FGSM and PGD, to our context. Note that for formulation 3 we need to only compute the gradient of In fact any attack algorithm that only relies on gradients of the loss function can be $d_\theta(\mathbf{z}, \mathbf{x}_1)$ with respect to $\mathbf{x}$. adapted for our case. For example the PGD attack can be adapted for contrastive loss l_c as follows:

$$\mathbf{x}_1^{t+1} \quad = \quad \Pi_{\mathbf{x}_1 + B_p(\epsilon)}(\mathbf{x}_1^t + \alpha\ \nabla_{\mathbf{x}_1} l_c(\theta, (\mathbf{x}, y), (\mathbf{x}_1, y_1)))$$

In the equation we are showing one iteration of the PGD and $B_p(\epsilon)$ is the ϵ ball centered at the origin using the l_p norm. For computational reasons, in our attack algorithms we only perturb one of the components for the tuples of triples.

Adversarial Training

Once we have the attack algorithm, adversarial training for robustifying the model is relatively straightforward. We assume that the attack algorithm only perturbs one component of the tuple or triple. Let $\mathcal{A}_r^c(\cdot, \cdot)$ ($r \in \{1, 2\}$) and $\mathcal{A}_r^t(\cdot, \cdot, \cdot)$ ($r \in \{1, 2, 3\}$) be the attack algorithms for the contrastive and the triple losses, respectively. In the attack algorithms given above r refers to the index of the component being perturbed (e.g. $\mathcal{A}_2((\mathbf{x}, y), (\mathbf{x}_1, y_1))$ and returns $((\mathbf{x}, y), (\mathbf{x}_1 + \delta, y_1))$. Next we describe adversarial training for contrasitive loss (the case for triple loss is similar). $\mathcal{A}(\mathbf{x})$ corresponds to formulation 3 (attempts to solve $\max_{\mathbf{z} \in B_p(\mathbf{x}, \epsilon)} d_\theta(\mathbf{z}, \mathbf{x})$).

As pointed before, formulation 1 is computationally prohibitive. We will focus on formulations 2 and 3. Let $S = \{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_n, y_n)\}$ be the dataset. At each iteration, a tuple $T = ((\mathbf{x}_i, y_i), (\mathbf{x}_j, y_j))$ is sampled from S. We construct the tuple T' from

T using attack algorithm (i.e. $T' = A^c_r(T)$ or $T' = A^t_r(T)$), and run one step of the learning algorithm, such as SGD or ADAM, on T'. This corresponds to formulation 2. For formulation 3 we use attack algorithm $\mathcal{A}$.

D.4 Experiments

Our experiments explore the following research questions:

Q1. How robust are *naturally* trained DML models towards established adversarial example attacks?

Among commonly used datasets for visual similarity, we find that DML models, trained with state-of-the-art parameter choices, are vulnerable to adversarial examples, similar to non-DML models (Table D.4.1). This forms our baseline for adversarial robustness.

Q2. What is the accuracy of DML models when they are trained using our robust formulation?

We find that DML models can be trained to become more robust across a variety of norms. For example, for a PGD attack with 5 iterations under $\ell_\infty(\epsilon = 0.01)$, we increase the adversarial accuracy to 53.6% compared from the state-of-the-art natural baseline of 0.2% for contrastive loss on the SOP dataset (Table D.4.1).

Q3. How does the robust training objective affect the learned embedding space?

Using a synthetic dataset, we demonstrate that the proposed adversarial training reduces the amount of shifting that adversarial perturbations can cause in the embedding space (Figure D.4.1).

We run all experiments on Nvidia Tesla V100 GPUs (32 GB) RAM. Our code is available at

`https://github.com/anonymous-koala-supporter/adversarial-deep-metric-learning`.

Experimental Setup

Datasets. We use the following four real-world image datasets for our experiments:

- **CUB200-2011** [D36]: Images of birds across 200 species and have a total of 11 788 images.
- **CARS196** [D17]: Dataset with images of cars spanning across 196 models, with a total of 16 185 images.
- **SOP** [D28]: Product images from eBay listings 120 053 images of 22 634 different online products.

- **VisualPhish** [D1]: Screenshots of benign websites, from a set of established brands, and phishing websites that attempt to replicate the visual appearance of their benign counterpart. It covers 146 brands across a total of 10 558 screenshots.

CUB200-2011, CARS196, and SOP are commonly used within the DML literature [D21]. These three datasets are divided into a training and testing set of approximately the same size by selecting the first half of classes for the training set, while having the remaining classes be in the testing set [D24]. This setup reflects an out-of-distribution scenario — a common application of DML. VisualPhish is a newer dataset that underlies the robust phishing detection model, VisualPhishNet [D1]. For a fair comparison, we adopt the train-test split from the VisualPhish implementation. This yields a test set of 717 website screenshots. In addition to these real-world datasets, we also include the following synthetic dataset:
Synth Dataset: A dataset with two classes a and b where data points $\mathbf{x} \in [0,1]^k$ and $k = 224 \times 224 \times 3$ to maintain identical dimensionality of the real-world datasets. Data points from each class are drawn from $\mathcal{N}(\mu, \sigma^2 I)$ st. $\sigma = 0.075$ while $\mu = 0.25$ for class a and $\mu = 0.75$ for class b.

Models & Training Parameters. We use default parameter choices from prior work that yield state-of-the-art performance on natural samples [50]. Main parameters are summarized in this section and provide a complete listing in Appendix D.I. Deviations from the default parameter choices are discussed and emphasized.

All models are ResNet50 [D13] initialized with pre-trained weights from an ImageNet classifier. We replace the last fully connected layer with another that matches the embedding space dimensionality. Embeddings are normalized to be on the n-dimensional unit sphere, where $n = 128$ throughout our experiments. We use ADAM [D16] with learning rate[3] of 10^{-6}, weight decay of $4 \cdot 10^{-4}$. We use contrastive and triplet losses during training, setting $\alpha = 1.0$ and $\alpha = 0.2$, respectively.

To the best of our knowledge, VisualPhishNet is the only previous attempt at creating an adversarially robust DML model trained using metric losses. At the core, the model is a variant of the VGG16 [D26] architecture with an unnormalized embedding layer of size 512. It was trained using the VisualPhish dataset and is expected to learn a visual similarity metric between web sites of various origins.

Training on the real-world datasets is performed over 150 epochs, with the exception of SOP that is trained for 100 epochs due to its volume [D24]. Mini-batches are of size 112 and sampled using the sampling technique SPC-2, which ensures that each batch contains exactly two samples per class for the selected classes in the batch.

Adversarial Robustness. To establish a benchmark for adversarial robustness, we employ the attack algorithm covered in Section D.3. For each data point being perturbed, we sample the nearest positive neighbor to reflect the ideal attack setting for an adversary. The formulation uses Projected Gradient Decent (PGD) [D19] because

[3] This learning rate differs from the one stated by Roth et al. [50] in their publication, 10^{-5}, but reflects the actual learning rate used throughout their experiments. See the field "lr" within experiment configuration: https://bit.ly/3a4FyHP.

it is considered one of the strongest white-box attacks available [D37]. Each attack is run for five iterations ($i = 5$) and has a step size given by $2\epsilon\frac{1}{i}$, such that the step size remains small while not hindering the optimization from reaching any point within the ϵ-ball despite random initialization. Throughout the experiments we use the notation of $\ell_p(\epsilon = 0.01)$ to indicate that for any data point $\mathbf{x}$, its valid perturbations are contained in $B_p(\mathbf{x}, 0.01)$. We compute the adversarial robustness for $\ell_\infty(\epsilon = 0.01)$ to accommodate VisualPhishNet [D1], and $\ell_2(\epsilon = 4)$ to provide comparisons for an alternative norm. In addition to PGD, we also investigate adversarial robustness towards the Carlini-Wagner (C&W) attack algorithm [D5], which can be found in Appendix D.II.

Adversarial Training. Given that natural training of DML models is already considered an expensive procedure [D24], solving the inner-maximization of the proposed robust formulations in Section D.3 can make the procedure even more expensive and potentially infeasible for practical applications. As previously discussed, the inner-maximization is solvable using traditional first-order attack methods, e.g. FGSM, PGD, and C&W. This fact enables us to apply a training technique by Wong, Rice, and Kolter [D37], that involves adversarial training using the cheaper R+FGSM [D31] attack, in conjunction with early-stopping. This yields similar increases in robustness towards stronger and more expensive attacks, such as PGD, despite not being directed trained on these attacks. For this attack, we define $\alpha = \epsilon \cdot 0.25$ as we have empirically determined that it is effective and training DML models. Using the proposed attack algorithm for adversarial training (Section D.3), we perturb the positive data points. This choice was to avoid affecting the relative distances to negative data points, which can induce instabilities during the training of DML models if they become too small [D38].

Evaluation Metrics. To evaluate the performance of the trained models, we employ the following DML-specific evaluation metrics: Recall at One (R@1) and Mean Average Precision at R (mAP@R) [D21]. R@1 is effectively the accuracy of class inference using the class of the nearest neighboring anchor within the embedding space produced by the model. Given the test set $S = \{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_n, y_n)\}$, and the function $I_k(i)$ that outputs the indices of the k-nearest neighbors for a data point $\mathbf{x}_i$, such that

$$I_k(i) = \underset{\substack{|K|=k \\ i \notin K}}{\arg\min} \sum_{j \in K} d_\theta(\mathbf{x}_i, \mathbf{x}_j) \ , \tag{D.18}$$

then R@1 given by:

$$\text{R@1} = \frac{1}{n} \sum_{\substack{i \in \{1, \cdots, n\} \\ j \in I_1(i)}} 1_{y_i = y_j} \ . \tag{D.19}$$

mAP@R is metric for measuring a model's ability to rank classes in the embedding space; we adopted this metric for the reasons covered by Musgrave, Belongie, and Lim [D21]. It is defined as

$$\text{mAP@R} = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{R_i}\sum_{k=1}^{R_i}\frac{1}{k}\sum_{j\in I_k(i)} 1_{y_i=y_j}\ , \tag{D.20}$$

where $R_i = \sum_{j\in\{1,\cdots,n\}\setminus\{i\}} 1_{y_i=y_j}$.

Experimental Results

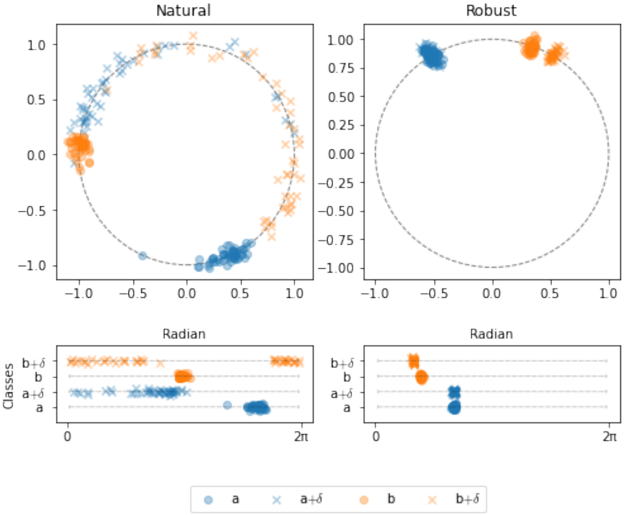


Fig. D.4.1: Embedding spaces of two trained models. (first row) Visualization of the embedding spaces of the two models (natural and robust). Effectively the embeddings are normalized to the unit circle, however, small distortions are added to improve visual clarity. Circles mark embeddings of benign data points, while crosses are data points with an adversarial perturbation (δ). (second row) Alternative visualization of the embedding space, showing the point relative to their radian on isolated spheres. It can be seen that embeddings of the robustly-trained model shifts much less, when faced with adversarial perturbed input, and are thus more robust.

Robustness of Natural Training (Q1) We establish a baseline of robustness against adversarial perturbations for naturally-trained DML models across the covered metric losses, ℓ_p-norms, and datasets. Results can be seen in Table D.4.1. Across any of the common real-world datasets (CUB200-2011, CARS196, SOP) it can be seen that both the model's ability to infer the correct class from its nearest neighbor (R@1) and its ability to rank classes (mAP@R) drops by several orders of magnitude. Exemplifying this, the naturally-trained model using triplet loss on CUB200-2011 drops from 59.3% accuracy (on benign data) down to 4.0% (on adversarially-perturbed data) for $\ell_\infty(\epsilon = 0.01)$. Naturally-trained models on CARS196 and SOP yield comparable or worse adversarial robustness. Notably, naturally-trained models on the VisualPhish dataset achieves a higher baseline for adversarial robustness. We suspect this deviation, from the other common real-world datasets, is related to the underlying data distribution of the dataset.

We conclude that naturally-trained DML models are not inherently robust, contrary to what results of prior work might indicate [D1]. We suspect this difference might stem from the method of attack or the fact we use a stronger first-order attack.

Adversarial Training for Robustness (Q2) From Table D.4.1, it can be seen that the proposed method for adversarial training increases adversarial robustness (accuracy and ability to rank) across the chosen metric losses, norms, attacks and datasets. As an example, the robust R@1 on SOP increases from 0.7% (naturally-trained) to 56.3% for $\ell(\epsilon = 0.01)$. We also observe that the proposed method increases the adversarial robustness (in terms of R@1) on the VisualPhish dataset to 56.7%, and thus outperforms the prior work of VisualPhishNet [D1], which achieves 42.8%. As shown in Appendix D.II, it can be seen that the gained robustness also applies to alternative attacks (C&W) for $\ell_2(\epsilon = 4)$. Performance of the trained robust models on benign input remains largely unaffected. Figure D.1.1 shows an example of inference under different training objectives, and we provide a publicly available gallery of other examples[4].

Effects on Embedding Space (Q3) Using the described synthetic dataset, we investigate the effect of adversarial training on the learned embedding space. The experiment involves training a DML model to map data points of the high-dimensional synthetic dataset, with the classes a and b, onto to a two-dimensional embedding space. We choose to have the embedding space be two-dimensional to allow visualizations of the learned embedding space. Each of the models, naturally-trained and robust, uses contrastive loss and is trained on approximately 15K data points. Adversarial perturbations are derived using the proposed attack formulation with PGD under $\ell_\infty(\epsilon = 0.01)$. Differences of the learned embedding spaces, and the influence of the adversarial perturbations, is shown in Figure D.4.1. We observe that the robust model is capable of maintaining smaller inter-class distances between adversarially perturbed data points and benign data points.

D.5 Conclusion

Deep Metric Learning (DML) creates feature embedding spaces where similar input points are geometrically close to each other, while dissimilar points are far apart. However, the underlying DNNs are vulnerable to adversarial inputs, thus making the DML models themselves vulnerable. We demonstrate that naturally-trained DML models are vulnerable to strong attackers, similar to other types of deep learning models. To create robust DML models, we contribute a robust training objective that can account for the *dependence* of metric losses — the phenomenon that the loss at any point depends on the other items in the mini-batch and the sampling process that was used to derive the mini-batch. Our robust training formulation yields robust DML models that

[4] `https://anonymous-koala-supporter.github.io/sample-gallery/`

can withstand PGD attacks without severely degrading their performance on benign inputs.

Table D.4.1: Performance of DML models across datasets, types of perturbations, and attack algorithms. Results are an average of five random seeds and best performances for a given combination of input data (*Benign*, $\ell_2(\epsilon = 4)$, $\ell_\infty(\epsilon = 0.01)$) and dataset is marked in **bold**. We highlight our robust training technique using a shaded background and suffix them by their trained norm (e.g. (ℓ_∞)). Recall that, R@1 reflects a model's inference accuracy, while mAP@R reflects its ability to rank similar entities. Naturally-trained DML models (Contrastive, Triplet) demonstrate low robustness towards the proposed attack formulation across the adversarial settings ($\ell_2(\epsilon = 4)$, $\ell_\infty(\epsilon = 0.01)$). Our robust training objective improves robustness towards adversarial attacks, and outperforms previous attempts at establishing robust DML models (VisualPhishNet).

		CUB200-2011		**CARS196**		**SOP**		**VisualPhish**	
Model	Attack	R@1	mAP@R	R@1	mAP@R	R@1	mAP@R	R@1	mAP@R
Benign (Natural samples)									
Contrastive	—	59.1 ± 0.0	21.0 ± 0.0	$\mathbf{74.0 \pm 0.0}$	20.9 ± 0.0	$\mathbf{71.8 \pm 0.0}$	$\mathbf{44.7 \pm 0.0}$	78.2 ± 0.0	73.6 ± 0.0
Triplet	—	$\mathbf{59.3 \pm 0.0}$	$\mathbf{21.7 \pm 0.0}$	74.0 ± 0.0	$\mathbf{21.4 \pm 0.0}$	69.6 ± 0.0	42.1 ± 0.0	$\mathbf{81.9 \pm 0.0}$	$\mathbf{77.0 \pm 0.0}$
VisualPhishNet	—	N/A						64.0 ± 0.0	17.6 ± 0.0
Contrastive (ℓ_2)	—	55.6 ± 0.0	19.5 ± 0.0	71.6 ± 0.0	18.8 ± 0.0	66.3 ± 0.0	38.4 ± 0.0	76.0 ± 0.0	70.9 ± 0.0
Triplet (ℓ_2)	—	55.9 ± 0.0	19.8 ± 0.0	71.6 ± 0.0	18.8 ± 0.0	62.2 ± 0.0	34.6 ± 0.0	76.8 ± 0.0	73.8 ± 0.0
Contrastive (ℓ_∞)	—	58.2 ± 0.0	20.2 ± 0.0	72.1 ± 0.0	19.5 ± 0.0	66.7 ± 0.0	39.0 ± 0.0	76.8 ± 0.0	72.1 ± 0.0
Triplet (ℓ_∞)	—	53.4 ± 0.0	17.9 ± 0.0	71.9 ± 0.0	19.8 ± 0.0	64.0 ± 0.0	36.4 ± 0.0	79.1 ± 0.0	76.0 ± 0.0
$\ell_2(\epsilon = 4)$									
Contrastive	PGD	8.6 ± 0.2	4.8 ± 0.1	3.7 ± 0.2	2.4 ± 0.0	2.3 ± 0.0	2.3 ± 0.0	45.5 ± 0.2	43.1 ± 0.1
Triplet	PGD	9.6 ± 0.3	5.5 ± 0.1	2.9 ± 0.1	2.2 ± 0.0	1.2 ± 0.0	1.8 ± 0.0	34.6 ± 0.6	28.5 ± 0.1
Contrastive (ℓ_2)	PGD	26.4 ± 0.3	11.1 ± 0.0	$\mathbf{37.2 \pm 0.2}$	$\mathbf{9.7 \pm 0.0}$	$\mathbf{51.6 \pm 0.0}$	$\mathbf{29.2 \pm 0.0}$	$\mathbf{56.6 \pm 0.4}$	$\mathbf{54.0 \pm 0.2}$
Triplet (ℓ_2)	PGD	$\mathbf{27.2 \pm 0.2}$	$\mathbf{11.3 \pm 0.1}$	37.0 ± 0.4	9.3 ± 0.0	37.7 ± 0.1	20.3 ± 0.0	55.1 ± 0.1	50.9 ± 0.2
$\ell_\infty(\epsilon = 0.01)$									
Contrastive	PGD	3.9 ± 0.3	3.4 ± 0.1	1.4 ± 0.2	1.9 ± 0.0	0.7 ± 0.0	1.5 ± 0.0	35.7 ± 0.2	34.3 ± 0.2
Triplet	PGD	4.0 ± 0.3	3.8 ± 0.1	0.7 ± 0.0	1.7 ± 0.0	0.2 ± 0.0	1.3 ± 0.0	20.8 ± 0.1	16.8 ± 0.3
VisualPhishNet	PGD	N/A						42.8 ± 0.2	13.2 ± 0.0
Contrastive (ℓ_∞)	PGD	$\mathbf{20.3 \pm 0.5}$	$\mathbf{8.8 \pm 0.0}$	35.7 ± 0.2	$\mathbf{9.7 \pm 0.1}$	$\mathbf{53.6 \pm 0.0}$	$\mathbf{30.4 \pm 0.0}$	$\mathbf{56.7 \pm 0.2}$	$\mathbf{53.6 \pm 0.1}$
Triplet (ℓ_∞)	PGD	16.9 ± 0.1	7.4 ± 0.1	$\mathbf{36.2 \pm 0.5}$	9.6 ± 0.1	39.3 ± 0.1	21.3 ± 0.0	54.7 ± 0.1	49.1 ± 0.1

Appendix

D.I Training Parameters (Expanded)

This section expands upon details and hyper-parameters used throughout the training of the respective DML models.

Batches & Sampling. Recall, that the training process uses a mini-batch size of 112 data points. Each mini-batch is sampled such that it contains exactly two samples per class [D24]. Following this, sets of tuples or triplets are derived (depended on loss) from the mini-batch using distance weighted sampling [D38] for negatives, while positives are given by pair-based sampling. Distance weighted sampling enhances the stability of training using metric losses, that can suffer from becoming stuck at a local minima early on in the training procedure [D38]. The cardinality of the triplet-set is identical to the mini-batch size. The size of the tuple-set is double the size of the mini-batch, thus balancing out the number of data points being compared relative to the triplet-set. Furthermore, each data point within the tuple-set is used in a positive and negative pair.

Data Augmentation. We augment the dataset using the following operations for each input image: (1) random cropping to an image patch of size 60-100% of the original image area; (2) scaling; (3) normalization of pixel intensities. One difference is that our patch sizes differ from Roth et al. [D24] that employs patches of size 8-100% of original area. We change this parameter because recent work suggests that computer vision models can be biased by backgrounds and textures during during [D39]. To combat this, we use cropping and scaling values based on Szegedy et al. [D29].

D.II Alternative Attack (Carlini-Wagner)

The Carlini-Wagner (C&W) attack is an unbounded attack, and thus constrains perturbations to lie within the given ℓ_p [D5]. We employ a clipping technique similar to Tramèr and Boneh [D32], which projects the perturbation to the ℓ_p-ball at every step. Additionally, as inference is costly for DML models (nearest neighbor search across embedding space), the ability of providing early stopping mechanism has been disabled. Results are presented in Table D.II.1. This is our best effort on providing strong hyper-parameters for the attack. It can be seen that the robustly trained model manages to remain higher robustness towards C&W attacks than the stronger PGD attack. The impact of the mentioned alterations, and the used hyper-parameters could yield the C&W attack to be non-optimal. Thereby, these results should be seen as an lower-bound representation of robustness towards the C&W, despite PGD generally being consider the state-of-the-art [D37].

D.III Theoretical Analysis

Robustness and Lipschitzness of DML

In Section D.1, we pointed out that the Lipschitzness of the DML model also plays an important role as in the traditional classifier situation. Here we have a formal analysis.

Let the sample space $\mathcal{X}$ be $\bigcup_{y \in \mathcal{Y}} \mathcal{X}_y$, where $\mathcal{Y} = \{y_1, \cdots, y_L\}$ is the space of labels and $\mathcal{X}_y \subseteq \mathcal{X}$ is the set of samples with label y. Suppose we have a deep embedding model f_θ with parameter θ trained using one of the loss functions described earlier. Let $A = \{(\mathbf{a}_1, c_1), \cdots, (\mathbf{a}_m, c_m)\}$ be a reference dataset (e.g. a set of faces along with their label), which we call *anchors*. Suppose we have a sample $\mathbf{z}$ and let $k(A, \mathbf{z})$ be the index that corresponds to $\arg\min_{j \in \{1, \cdots, m\}} d_\theta(\mathbf{a}_j, \mathbf{z})$. We predict the label of $\mathbf{z}$ as $lb(A, \mathbf{z}) = c_{k(A,\mathbf{z})}$. Recall that $d_\theta(\mathbf{x}, \mathbf{z})$ is the distance metric in the embedding space $\| f_\theta(\mathbf{x}) - f_\theta(\mathbf{z}) \|_2$.

We will assume that our sample space $\mathcal{X}$ is a metric space with metric μ. A point $\mathbf{z} \in \mathcal{X}$ is *ϵ-robust* w.r.t. A, μ and d_θ iff for $k(A, \mathbf{z}) = \arg\min_{1 \leq i \leq m} d_\theta(a_i, \mathbf{z})$, we have that for all $j \neq k(A, \mathbf{z})$ and $\mu(\mathbf{x}, \mathbf{z}) \leq \epsilon$, $d_\theta(a_j, \mathbf{x}) > d_\theta(a_{k_\mu(A,\mathbf{z})}, \mathbf{x})$. In other words, perturbing $\mathbf{z}$ by ϵ in the sample space does not change the anchor it is close to in the embedding space.

A point $\mathbf{z} \in \mathcal{X}$ is *δ-separated* w.r.t. A and d_θ iff for $k(A, \mathbf{z}) = \arg\min_{1 \leq i \leq m} d_\theta(a_i, \mathbf{z})$ we have that for all $j \neq k(A, \mathbf{z})$, $d_\theta(a_j, \mathbf{z}) > d_\theta(a_{k_\mu(A,\mathbf{z})}, \mathbf{z}) + \delta$. In other words, $f_\theta(\mathbf{z})$ is at least δ closer to its anchor than other anchors in the embedding space. As a result, f_θ correctly classifies $\mathbf{z}$.

We assume that d_θ is L-Lipschitz, i.e., for all $\mathbf{x}$ and $\mathbf{z}$ in $\mathcal{X}$:

$$d_\theta(\mathbf{x}, \mathbf{z}) \leq \quad L\mu(\mathbf{x}, \mathbf{z})$$

Lemma 1. *If $L \leq \delta/(2\epsilon)$, and $\mathbf{z}$ is δ-separated w.r.t. A and d_θ, then $\mathbf{z}$ is ϵ-robust.*

Proof. Let $i = k(A, \mathbf{x})$, $j \neq i$, and $\mu(\mathbf{x}, \mathbf{z}) \leq \epsilon$.

$$\begin{aligned} d_\theta(a_i, \mathbf{x}) &\leq d_\theta(\mathbf{x}, \mathbf{z}) + d_\theta(\mathbf{z}, a_i) \\ &\leq L\mu(\mathbf{x}, \mathbf{z}) + d_\theta(\mathbf{z}, a_i) \\ &< L\mu(\mathbf{x}, \mathbf{z}) + d_\theta(\mathbf{z}, a_j) - \delta \\ &\leq L\mu(\mathbf{x}, \mathbf{z}) + [d_\theta(\mathbf{z}, \mathbf{x}) + d_\theta(\mathbf{x}, a_j)] - \delta \\ &\leq L\mu(\mathbf{x}, \mathbf{z}) + L\mu(\mathbf{z}, \mathbf{x}) + d_\theta(\mathbf{x}, a_j) - \delta \end{aligned}$$

Because $L \leq \delta/(2\epsilon)$, $\mu(\mathbf{z}, \mathbf{x}) \leq \epsilon$, we have

$$d_\theta(a_j, \mathbf{x}) > d_\theta(a_i, \mathbf{x}) + (\delta - 2L\epsilon) \geq d_\theta(a_i, \mathbf{x}),$$

so

$$d_\theta(a_i, \mathbf{x}) < d_\theta(a_j, \mathbf{x}).$$

□

DML with Gaussian Mixture Model

To further motivate the connection between robustness of an embedding and its Lipschitz constant, we consider a Gaussian mixture model. These models have been considered in the theoretical analysis of robustness in the classification setting [D7, D33]. Our synthetic dataset experiment (Figure D.4.1) illustrates this Gaussian mixture model setting. Let $\mathcal{N}(\mu, \Sigma)$ be the Gaussian distribution in $\mathbb{R}^n$ with mean $\mu \in \mathbb{R}^n$ and Σ a $n \times n$ positive-definite matrix. We will consider Gaussian distributions of the form $\mathcal{N}(\mu, I_n)$ where I_n is the $n \times n$ identity matrix.

Let $\mathcal{X} \times \mathcal{Y}$ (where $\mathcal{Y} = \{-1, 1\}$) be generated from a distribution $\mathcal{D}$ as follows: $y \in \mathcal{Y}$ is equally probable with probability $\frac{1}{2}$ and given y, generate $\mathbf{x}$ according to $\mathcal{N}(y\mu, I_n)$.

We have the following concentration of measure result from Theorem 5.2.2 [D34].

Theorem 1. (*Gaussian concentration*) *Consider a random vector* $\mathbf{x} \sim \mathcal{N}(0, I_n)$ *and a Lipschitz function* $f : \mathbb{R}^n \to \mathbb{R}$. *Then*

$$\| f(\mathbf{z}) - Ef(\mathbf{x}) \|_{\psi_2} \quad \leq \quad C \| f \|_{Lip},$$

where $\| f \|_{Lip}$ *is the Lipschitz constant of f, and* $\| \cdot \|_{\psi_2}$ *is the sub-Gaussian metric.*

Consider a DML model $f_\theta : \mathbb{R}^n \to S^d$, and let d_θ be the associated distance metric. Let a_1 and a_{-1} be the anchors for labels 1 and -1 respectively. Consider the two functions defined as follows: $f_1(\mathbf{x}) = d_\theta(a_1, \mathbf{x})$ and $f_{-1}(\mathbf{x}) = d_\theta(a_{-1}, \mathbf{x})$ (the functions correspond to the distances from the two anchors).

$$\begin{aligned} \beta_1 &= E_{\mathbf{x} \sim \mathcal{N}(\mu, I_n)} f_1(\mathbf{x}) \\ \beta_{-1} &= E_{\mathbf{x} \sim \mathcal{N}(-\mu, I_n)} f_2(\mathbf{x}) \end{aligned}$$

We first show that f_1 is L-Lipschitz if f_θ is L-Lipschitz. Take $\mathbf{x}, \mathbf{z} \in \mathcal{X}$,

$$\begin{aligned} |f_1(\mathbf{x}) - f_1(\mathbf{z})| &= |d_\theta(a_1, \mathbf{x}) - d_\theta(a_1, \mathbf{z})| \\ &\leq d_\theta(\mathbf{x}, \mathbf{z}) \\ &\leq L\mu(\mathbf{x}, \mathbf{z}) \end{aligned}$$

As a result, $\| f_1 \|_{Lip} = \| f_\theta \|_{Lip}$. Intuitively, if the Lipschitz constant of f_1 is lower, the points drawn from $\mathcal{N}(\mu, I_n)$ get closer to β_1. In other words, as the Lipschitz constant of embedding gets smaller, the "point clouds" corresponding to the two Gaussian distributions in the mixture get farther apart, because they are concentrated more around their means.

Next we formalize this intuition. Let $E(\mathbf{x}, a_1, a_{-1})$ represent the event that $\mathbf{x}$ is closer to a_{-1} than a_1. We prove the following:

$$P_{\mathbf{x} \sim \mathcal{N}(\mu, I_n)}(1_{E(\mathbf{x}, a_1, a_{-1})}) \leq 2 \exp\left(- \frac{C' z^2}{\| f_1 \|_{Lip}} \right) \tag{D.21}$$

In the equation given above, $C' > 0$ is a positive constant, and z is given by the following expression:

$$\frac{d_\theta(a_{-1}, a_1)}{2} - \beta_1$$

Notice that $P_{\mathbf{x} \sim \mathcal{N}(\mu, I_n)}(1_{E(\mathbf{x}, a_1, a_{-1})})$ represents the probability that a point drawn from $\mathcal{N}(\mu, I_n)$ is closer to a_{-1} than a_1, and hence represents an "undesirable event". Also note that the upper bound goes down as the Lipschitz constant $\| \ f_1 \ \|_{Lip}$ goes down, and thus confirming our intuition. Next we prove Equation D.21.

Let X be a sub-Gaussian random variable, then the following equation is well-known:

$$P(| \ X \ | \geq t) \quad \leq \quad 2 \exp \left(\frac{-ct^2}{\| \ X \ \|_{\psi_2}^2} \right) \tag{D.22}$$

To prove the Equation D.21, we use the following sequence of inequalities (let $q = P_{\mathbf{x} \sim \mathcal{N}(\mu, I_n)}(1_{E(\mathbf{x}, a_1, a_{-1})})$)

$$\begin{aligned} q &\leq P_{\mathbf{x} \sim \mathcal{N}(\mu, I_n)} \left(f_1(\mathbf{x}) \geq \frac{d_\theta(a_{-1}, a_1)}{2} \right) \\ &\leq P_{\mathbf{x} \sim \mathcal{N}(\mu, I_n)} \left(| \ f_1(\mathbf{x}) - \beta_1 \ | \geq \frac{d_\theta(a_{-1}, a_1)}{2} - \beta_1 \right) \\ &\leq 2 \exp \left(- \frac{C' z^2}{\| \ f_1 \ \|_{Lip}} \right) \end{aligned}$$

The first step follows from the following observation: if $f_1(\mathbf{x})$ is less than $\frac{d_\theta(a_{-1}, a_1)}{2}$ then $\mathbf{x}$ is closer to a_1 than a_{-1}. The next two steps use Theorem 1 and Equation D.22.

Table D.II.1: Performance of robust DML models over four datasets for $\ell_2(\epsilon = 4)$ against the C&W attack algorithms. Results are an average of five random seeds. Recall that, R@1 reflects a model's inference accuracy, while mAP@R reflects its ability to rank similar entities. Our robust training objective also provides robustness towards C&W attacks.

		CUB200-2011		**CARS196**		**SOP**		**VisualPhish**	
Model	Attack	R@1	mAP@R	R@1	mAP@R	R@1	mAP@R	R@1	mAP@R
				$\ell_2(\epsilon = 4)$					
Contrastive (ℓ_2)	PGD	26.4 ± 0.3	11.1 ± 0.0	37.2 ± 0.2	9.7 ± 0.0	51.6 ± 0.0	29.2 ± 0.0	56.6 ± 0.4	54.0 ± 0.2
$\hookrightarrow$	**C&W**	**40.6 ± 0.4**	**17.2 ± 0.1**	**58.7 ± 0.2**	**16.4 ± 0.0**	**63.2 ± 0.0**	**37.2 ± 0.0**	**67.8 ± 0.2**	**64.6 ± 0.1**
Triplet (ℓ_2)	PGD	27.2 ± 0.2	11.3 ± 0.1	37.0 ± 0.4	9.3 ± 0.0	37.7 ± 0.1	20.3 ± 0.0	55.1 ± 0.1	50.9 ± 0.2
$\hookrightarrow$	**C&W**	**42.0 ± 0.1**	**17.7 ± 0.1**	**59.6 ± 0.1**	**16.4 ± 0.0**	**54.5 ± 0.0**	**30.5 ± 0.0**	**68.7 ± 0.2**	**65.8 ± 0.2**

References

[D1] Sahar Abdelnabi, Katharina Krombholz, and Mario Fritz. "VisualPhishNet: Zero-Day Phishing Website Detection by Visual Similarity". In: *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security, CCS 2020, November 9-13, 2020*. ACM, 2020.

[D2] A. Ben-Tal, L. El Ghaoui, and A.S. Nemirovski. *Robust Optimization*. Princeton Series in Applied Mathematics. Princeton University Press, Oct. 2009.

[D3] Battista Biggio et al. "Evasion Attacks against Machine Learning at Test Time". In: *Lecture Notes in Computer Science* (2013), pp. 387–402. ISSN: 1611-3349. DOI: 10.1007/978-3-642-40994-3_25. URL: http://dx.doi.org/10.1007/978-3-642-40994-3_25.

[D4] Malik Boudiaf et al. "A unifying mutual information view of metric learning: cross-entropy vs. pairwise losses". In: *European Conference on Computer Vision*. Springer. 2020, pp. 548–564.

[D5] N. Carlini and D. Wagner. "Towards Evaluating the Robustness of Neural Networks". In: *2017 IEEE Symposium on Security and Privacy (SP)*. 2017, pp. 39–57.

[D6] Gal Chechik et al. "Large Scale Online Learning of Image Similarity Through Ranking". In: *J. Mach. Learn. Res.* 11 (Mar. 2010), pp. 1109–1135. ISSN: 1532-4435.

[D7] C. Dan, Y. Wei, and P. Ravikumar. "Sharp statistical guarantees for adversarially robust gaussian classification". In: *CoRR, abs/2006.16384* (2020).

[D8] Jiankang Deng et al. "ArcFace: Additive Angular Margin Loss for Deep Face Recognition". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2019). DOI: 10.1109/cvpr.2019.00482. URL: http://dx.doi.org/10.1109/CVPR.2019.00482.

[D9] Y. Duan et al. "Deep Adversarial Metric Learning". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 2780–2789.

[D10] Jacob Goldberger et al. "Neighbourhood components analysis". In: *Advances in neural information processing systems* 17 (2004), pp. 513–520.

[D11] Ian Goodfellow et al. "Generative Adversarial Nets". In: *Advances in Neural Information Processing Systems 27*. Ed. by Z. Ghahramani et al. Curran Associates, Inc., 2014, pp. 2672–2680. URL: http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf.

[D12] Raia Hadsell, Sumit Chopra, and Yann LeCun. "Dimensionality Reduction by Learning an Invariant Mapping". In: *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2*. CVPR '06. IEEE Computer Society, 2006. DOI: 10.1109/CVPR.2006.100. URL: https://doi.org/10.1109/CVPR.2006.100.

[D13] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2016).

[D14] Matthias Hein and Maksym Andriushchenko. "Formal guarantees on the robustness of a classifier against adversarial manipulation". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. 2017, pp. 2263–2273.

[D15] Alexander Hermans, Lucas Beyer, and Bastian Leibe. "In Defense of the Triplet Loss for Person Re-Identification". In: *ArXiv* abs/1703.07737 (2017).

[D16] Diederick P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *International Conference on Learning Representations (ICLR)*. 2015.

[D17] Jonathan Krause et al. "3D Object Representations for Fine-Grained Categorization". In: *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*. Sydney, Australia, 2013.

[D18] Pengcheng Li et al. "Improving the Robustness of Deep Neural Networks via Adversarial Training with Triplet Loss". In: *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, July 2019, pp. 2909–2915. DOI: 10.24963/ijcai.2019/403. URL: https://doi.org/10.24963/ijcai.2019/403.

[D19] Aleksander Madry et al. "Towards Deep Learning Models Resistant to Adversarial Attacks". In: *International Conference on Learning Representations*. 2018. URL: https://openreview.net/forum?id=rJzIBfZAb.

[D20] Chengzhi Mao et al. *Metric Learning for Adversarial Robustness*. 2019. arXiv: 1909.00900 [cs.LG].

[D21] Kevin Musgrave, Serge Belongie, and Ser-Nam Lim. *A Metric Learning Reality Check*. 2020. arXiv: 2003.08505 [cs.CV].

[D22] T. K. Panum et al. *Towards Adversarial Phishing Detection*. 2020.

[D23] Karsten Roth, Biagio Brattoli, and Bjorn Ommer. "MIC: Mining Interclass Characteristics for Improved Metric Learning". In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* (Oct. 2019). DOI: 10.1109/iccv.2019.00809. URL: http://dx.doi.org/10.1109/ICCV.2019.00809.

[D24] Karsten Roth et al. *Revisiting Training Strategies and Generalization Performance in Deep Metric Learning*. 2020. arXiv: 2002.08473 [cs.CV].

[D25] Florian Schroff, Dmitry Kalenichenko, and James Philbin. "FaceNet: A unified embedding for face recognition and clustering". In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2015).

[D26] Karen Simonyan and Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *3rd International Conference on Learning Representations, ICLR 2015*. Ed. by Yoshua Bengio and Yann LeCun. 2015.

[D27] Kihyuk Sohn. "Improved deep metric learning with multi-class n-pair loss objective". In: *Proceedings of the 30th International Conference on Neural Information Processing Systems*. 2016, pp. 1857–1865.

[D28] Hyun Oh Song et al. "Deep Metric Learning via Lifted Structured Feature Embedding". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[D29] Christian Szegedy et al. "Going Deeper with Convolutions". In: *Computer Vision and Pattern Recognition (CVPR)*. 2015.

[D30] Christian Szegedy et al. "Intriguing properties of neural networks". In: *International Conference on Learning Representations*. 2014. URL: `http://arxiv.org/abs/1312.6199`.

[D31] F Tramèr et al. "Ensemble adversarial training: Attacks and defenses". In: *6th International Conference on Learning Representations, ICLR 2018-Conference Track Proceedings*. 2018.

[D32] Florian Tramèr and Dan Boneh. "Adversarial Training and Robustness for Multiple Perturbations". In: *2019 Conference on Neural Information Processing Systems (NeurIPS)*. Vol. 32. 2019.

[D33] D. Tsipras et al. "Robustness may be at odds with accuracy". In: *In 7th International Conference on Learning Representations (ICLR)*. 2019.

[D34] Roman Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge University Press, 2018.

[D35] Xun Wang et al. "Multi-Similarity Loss With General Pair Weighting for Deep Metric Learning". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Oct. 2019). DOI: `10.1109/cvpr.2019.00516`. URL: `http://dx.doi.org/10.1109/CVPR.2019.00516`.

[D36] P. Welinder et al. *Caltech-UCSD Birds 200*. Tech. rep. CNS-TR-2010-001. California Institute of Technology, 2010.

[D37] Eric Wong, Leslie Rice, and J. Zico Kolter. "Fast is better than free: Revisiting adversarial training". In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. 2020. URL: `https://openreview.net/forum?id=BJx040EFvH`.

[D38] Chao-Yuan Wu et al. "Sampling Matters in Deep Embedding Learning". In: *2017 IEEE International Conference on Computer Vision (ICCV)* (Oct. 2017). DOI: `10.1109/iccv.2017.309`. URL: `http://dx.doi.org/10.1109/ICCV.2017.309`.

[D39] Kai Xiao et al. *Noise or Signal: The Role of Image Backgrounds in Object Recognition*. 2020. arXiv: `2006.09994 [cs.CV]`.

[D40] Hong Xuan, Abby Stylianou, and Robert Pless. "Improved embeddings with easy positive triplet mining". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2020, pp. 2474–2482.

[D41] Yuhui Yuan, Kuiyuan Yang, and Chao Zhang. "Hard-aware deeply cascaded embedding". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 814–823.

[D42] Wenzhao Zheng et al. “Hardness-aware deep metric learning”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 72–81.

Paper E

Haaukins: A Highly Accessible and Automated Virtualization Platform for Security Education

Thomas Kobber Panum, Kaspar Hageman, Jens Myrup Pedersen, René Rydhof Hansen

In proceedings of the
IEEE 19th International Conference on Advanced Learning Technologies (ICALT 2019).

Abstract. *Education of IT security can include a tedious and frustrating experience for novice students and organizers. We have sought out to create an education platform that improves upon this experience, through automation, and individualized learning labs. These learning labs hosts are isolated clusters of virtual computer instances, representing real and insecure computer networks. The platform, named Haaukins, improves upon typical accessibility issues of students and cumbersome configuration management for organizers. In order make the platform accessible for other organizations, it has been open sourced.*

©2019 IEEE. Reprinted, with permission, from Thomas Kobber Panum, Kaspar Hageman, Jens Myrup Pedersen, René Rydhof Hansen.
The layout has been revised.

E.1 Introduction

On a global scale, there is an increasing demand for qualified personnel for information security related positions. The $(ISC)^2$ estimates the shortage of security professionals to be 2.93 million persons [E1]. The Danish government has recognized this gap and is increasing its funding of cyber security education and research in response [E6]. In parallel, Aalborg University (Denmark) has started to address this issue by adding cyber security to the curriculum of relevant engineering educations.

This effort is accompanied by the university's ambassador programme, which engages university students to reach out and teach high school students about their respective field of education. For cyber security, this involves running introductory workshops with subjects such as vulnerability scanning and exploitation. The sessions are typically short, between one and two hours of duration, and are initiated by a short lecture followed by exercises where participants interact with a computer system, referred to as a *lab*. As an organizer, managing these labs is time-consuming and error-prone, especially as participants often need assistance, e.g., to gain access to the lab. Our collective experience from hosting sessions with existing solutions led to a series of requirements for an education platform.

In order to have a solution that conforms to these requirements, we have developed a new virtualization platform, Haaukins, which grew from the collaboration project 'Danish Cyber Security Clusters'. The platform differs from existing work by automating the tedious management of the labs, while also providing an individualized experience to improve learning and making labs accessible with no configuration. It is designed to support the common exercise format, Jeopardy capture-the-flag (CTF) [E3], in which participants must gain access to certain secret information contained within a given computer system.

E.2 Related Work

Although there exist numerous platforms for deploying labs of connected, virtual instances, none of these fulfill the requirements of our specific use case in its entirety. The popularity of CTFs as a competition format has resulted in a range of commercial, closed platforms to support running such events [E9, E8]. The fact that these platforms are not open makes them unsuitable for an educational use, since the educator is completely reliant on the owner in both the access to the platform and the material hosted on them.

A range of existing platforms places various teams in a single, shared virtual network. The motivations for doing so range from a performance consideration [E10] to the desire to host attack and defense CTFs (ADCTFs) [E13, E11]. In ADCTFs, participants not only attempt to hack other machines, but actively have to protect their own against others. Given the limited prior knowledge of our target audience, it is infeasible for us to host ADCTFs.

PicoCTF is organized yearly and — similarly to Haaukins — aims to create an interest for information security among high school students [E2]. The platform is developed as an interactive game, and thereby does not reflect a realistic scenario.

In summary, the related work is either closed, not suitable for our CTF format or does not represent the real world in a realistic fashion.

E.3 Design Goals

The design of Haaukins has primarily been driven by four design goals: automation, transparency, accessibility and realism. In the following subsections, we describe each design goal in some detail.

Fully Automated (DG$_{FA}$) In the process of preparing a CTF, numerous components need to be instantiated, configured and connected correctly. Manual configuration of labs is time-consuming process and errors can have severe consequences, e.g. a virtual instance being completely unreachable from one wrongly assigned IP address. Haaukins reduces the preparation time for events through full automation of the configuration of its labs.

Transparent (DG$_{T}$) Discovering and identifying a security vulnerability requires investigation and exploration. This element of exploration must be contained in Haaukins, without causing participants to end up in dead ends that can hinder their learning [E7, E4]. If the design of the exercises is the cause of these dead ends, it has be identifiable. Ideally a platform allows for gaining insight into this, by providing a method for observing participants' behavior and make them available for further analysis.

Highly Accessible (DG$_{HA}$) In the setting of a one- or two hour lecture, time is a valuable asset that should not be wasted on irrelevant aspects. Given the short timespan, the overhead of accessing a virtual lab can take up a significant fraction of the allocated time. Beginners might find this non-trivial and it may be a hurdle for progressing. We strive to minimize the overhead, and ideally want participants to access their virtual lab in a matter of minutes, independently from their physical location and the operating system they use.

Realistic (DG$_{R}$) For skills learned through a simulated environment to be valuable, they must be transferable to the real world setting. Haaukins strives to do so by ensuring that the designed labs are realistic replicas of real vulnerable computers and their networks. The labs are interacted with using a professional toolkit, that continuously evolves to remain relevant for trends in security vulnerabilities. Experiences gained from the labs should indistinguishable from real world settings, and exercise developers should not be restricted by the limitations of the platform.

E.4 Overview

A key feature of Haaukins is the option for multiple organizers to host simultaneous sessions (referred to as EVENTS) with one or more exercises for participants. For an event, a group of participants registers as a TEAM which is assigned an environment

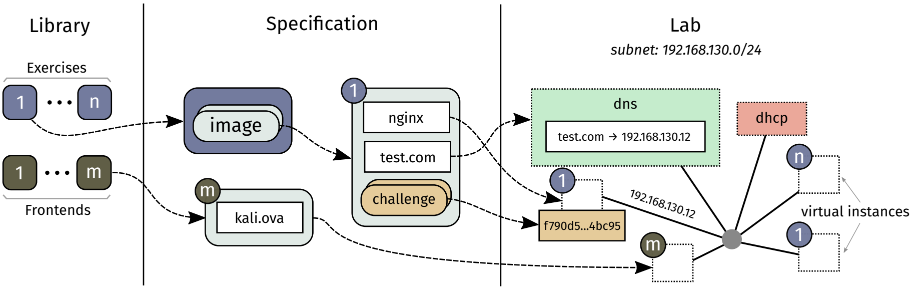


Fig. E.4.1: Levels of abstraction of a lab. Based on a selection of exercises and frontends (*left*), Haaukins uses their specification (*middle*) to create an individualized lab of instances per team (*right*).

upon registration. This environment is referred to as a LAB and is represented by a network of virtual INSTANCES. Teams are tasked to discover unique identifiers (or FLAGS), that function as proof for solving a CHALLENGE, which can be checked through a web application, and upon being valid, are counted as positive scores for the team.

A new event starts with any composition of EXERCISES and FRONTENDS (see Figure E.4.1). A frontend is an instance that a team gets to control via a graphical user interface. An exercise is composed of any number of IMAGES, which are templates from which an instance is created. The specification of an image consists of: a virtual disk image, any number of records, and any number of challenges. The virtual disk image can either be a Docker image or an Open Virtual Appliance (OVA) package, e.g. *nginx* or *kali.ova* in Figure E.4.1. The records are DNS records, which map domain names to IP addresses per lab, and are necessary for the communication among instances.

The instantiation of a lab consists of automatic creation and configuration of instances based on the collective specification of exercises and frontends. In addition to the associated instances, every lab also contains a set of core services to support connectivity and service discovery for instances, these services are DHCP and DNS respectively. The instantiation process also involves inserting unique flags in instances and randomizing IP address ranges, thereby individualizing each lab and its challenges.

E.5 Design

Haaukins consists of a client and a server component, `hkn` and `hknd` respectively, enabling multiple organizers to interact with the same instance of `hknd` independently of each other.

The daemon process, `hknd`, controls the life cycle of internal data structures and the orchestration of all components; it further serves as an application wide reverse HTTP proxy acting as a single point of entry for all the web traffic that comes in from the participants of the platform, and redirects the traffic to the correct virtual instances. On an event-level, there are two third-party components being managed: CTFd [E5] and Guacamole [E12]. CTFd is a web application responsible for the graphical user interface for the participants, which allows them to access their respective event through a web browser. Through this interface, the teams can view their respective exercises, fill in the results for their respective challenges, and are directed to their respective frontends which are accessible through Guacamole. Guacamole is a web application which allows for streaming remote desktops to a web browser, and is more thoroughly covered in Section E.5.

The client, `hkn`, provides a command-line interface (CLI) that allows organizers to interact with `hknd`, e.g., to create events, listing exercises or resetting instances for certain teams.

Automated Orchestration Haaukins uses Docker containers and Oracle VirtualBox (VirtualBox) virtual machines for deploying multiple isolated services within labs. These technologies are both used to allocate and isolate the resources (e.g. CPU and RAM) of a physical machine into virtual instances, but their ability to do so differs in terms of computation overhead and level of isolation.

All instances in a lab are connected to the same virtual network, to ensure that all instances can communicate among each other. Concretely, Docker Macvlan is used for the networks, resulting in a LAN network topology that allows for promiscuous network monitoring and gives participants the ability to observe the entire network traffic. External network access, such as the Internet, has been disabled to prevent participants from abusing the instances.

Lab Individualization In highly competitive CTFs, the sought-after flags are identical across participating teams, since the competitive nature is an incentive for keeping found flags private. From our experience this incentive does not transfer to an educational setting, as students are more inclined to share solutions, causing cheating that negatively influences DG_T through false progress. To combat this issue, Haaukins individualizes each lab through two techniques: dynamic flags and dynamic subnets. Creating dynamic flags is the process of creating unique flags on a per team basis in dynamic exercises, and thereby prevent the sharing of flags. Implementing dynamic subnets involves hosting labs on networks with randomized private IP ranges, which is a significant mutation in the setting of network analysis exercises. This prevents students from sharing information about IP addresses of instances, with only a few exceptions of core services, i.e., DNS and DHCPD.

Accessible Through a Web Browser Based on our experience, novice participants often encounter problems with the use of a (desktop-)client for remote access, such as VPN or RDP. These problems led to the choice of using Apache Guacamole within Haaukins, which is a web application that allows for accessing remote desktop protocols, e.g. RDP, through a modern web browser. Guacamole uses a backend daemon for translating standardized protocols to WebSocket traffic that is interpretable by a JavaScript client in the user's browser. Within Haaukins, the built-in capabilities of VirtualBox is utilized for creating RDP access to frontends. This access is then translated by Guacamole in order to be accessed from the participant's browser, requiring no installation process and being accessible from any physical location.

Monitoring Participants The design of effective exercises is complicated, as the exercises need to be open-ended in order to adhere to DG_R, while ensuring the openness does not cause participants to become stuck. In Haaukins, participants are initially tasked with identifying vulnerabilities, before actively exploiting them. If the participants become stuck at this stage, it will hinder their ability to learn from the exercises. To determine when and how this behavior occurs, Haaukins has the ability to monitor and log participant interaction with the platform. Only if consent is granted, the WebSocket traffic of Guacamole is captured using the reverse proxy of `hknd` and transformed to a stream of key presses. These streams can then be further analyzed in order to determine participant behavior, e.g. programming activities and terminal usage, that can potentially influence changes to the teaching material.

E.6 Conclusions

We present a novel education platform, Haaukins, that differentiates itself from existing CTF platforms by having improved accessibility, full automation, observability of participant behavior and high degree of realism. The platform presents itself as a web application that provides highly accessible lab environments for participants, accessible within minutes without prior experience. It completely automates the creation, configuration, teardown of all its components. Each lab is personalized through unique mutations of flags and IP addresses, which discourages cheating among participants. Since the labs in Haaukins are designed to be a realistic representation of a realistic network, learnings from the platform translate directly to real-world scenarios. In order to support other education institutions in conducting short CTF workshops, the platform is available as an open source project on GitHub[1] with a GNU GPLv3 license.

[1] `https://github.com/aau-network-security/haaukins`

References

[E1] (ISC)^2. *Cybersecurity Workforce Study*. URL: https://www.isc2.org/research/workforce-study.

[E2] Peter Chapman, Jonathan Burket, and David Brumley. "PicoCTF: A Game-Based Computer Security Competition for High School Students". In: *2014 USENIX Summit on Gaming, Games, and Gamification in Security Education (3GSE 14)* May (2014).

[E3] Tom Chothia and Chris Novakovic. "An Offline Capture The Flag-Style Virtual Machine and an Assessment of its Value for Cybersecurity Education". In: *2015 USENIX Summit on Gaming, Games, and Gamification in Security Education (3GSE 15)* (2015).

[E4] Kevin Chung and Julian Cohen. "Learning Obstacles in the Capture The Flag Model". In: *2014 USENIX Summit on Gaming, Games, and Gamification in Security Education (3GSE 14)*. 2014.

[E5] CTFd Maintainers. *CTFd*. URL: https://github.com/CTFd/CTFd.

[E6] Danish Ministry of Finance. *Danish Cyber and Information Security Strategy 2018-2021*. URL: https://uk.fm.dk/publications/2018/danish-cyber-and-information-security-strategy.

[E7] Wu-Chang Feng. "A Scaffolded, Metamorphic CTF for Reverse Engineering". In: *2015 USENIX Summit on Gaming, Games, and Gamification in Security Education (3GSE 15)* (2015).

[E8] Hack The Box Ltd. *HackTheBox.eu*. URL: https://www.hackthebox.eu/.

[E9] Hacking-Lab. *Hacking-lab*. URL: https://www.hacking-lab.com/.

[E10] Lucas McDaniel, Erik Talvi, and Brian Hay. "Capture the Flag as Cyber Security Introduction". In: *2016 49th Hawaii International Conference on System Sciences (HICSS)*. IEEE. 2016, pp. 5479–5486.

[E11] Jelena Mirkovic and Peter A. H. Peterson. "Class Capture-the-Flag Exercises". In: *2014 USENIX Summit on Gaming, Games, and Gamification in Security Education (3GSE 14)*. 2014.

[E12] The Apache Software Foundation. *Apache Guacamole*. URL: https://guacamole.apache.org.

[E13] Giovanni Vigna et al. "Ten Years of iCTF: The Good, The Bad, and The Ugly". In: *2014 USENIX Summit on Gaming, Games, and Gamification in Security Education (3GSE 14)* (2014), pp. 1–7.

Paper F

Bridging the Gap: Adapting a Security Education Platform to a New Audience

Gian Marco Mennecozzi, Kaspar Hageman, Thomas Kobber Panum, Ahmet Türkmen, Rasmi-Vlad Mahmoud, Jens Myrup Pedersen

In proceedings of the
IEEE 12th Global Engineering Education Conference (EDUCON 2021).

Abstract. *The current supply of a highly specialized cyber security professionals cannot meet the demands for societies seeking digitization. To close the skill gap, there is a need for introducing students in higher education to cyber security, and to combine theoretical knowledge with practical skills. This paper presents how the cyber security training platform Haaukins, initially developed to increase interest and knowledge of cyber security among high school students, was further developed to support the need for training in higher education. Based on the differences between the existing and new target audiences, a set of design principles were derived which shaped the technical adjustments required to provide a suitable platform - mainly related to dynamic tooling, centralized access to exercises, and scalability of the platform to support courses running over longer periods of time. The implementation of these adjustments has led to a series of teaching sessions in various institutions of higher education, demonstrating the viability for Haaukins for the new target audience.*

©2019 IEEE. Reprinted, with permission, from Gian Marco Mennecozzi, Kaspar Hageman, Thomas Kobber Panum, Ahmet Türkmen, Rasmi-Vlad Mahmoud, Jens Myrup Pedersen.

The layout has been revised.

F.1 Introduction

The demand for cyber security professionals has been increasing during the past years and is expected to increase even more in the future [F17]. To meet this increased demand, various higher education institutions globally have started to develop curricula specifically to educate engineers in cyber security, while others have integrated cyber security into their existing teaching curricula [F1, F16]. Practical exercises are an essential part in such curricula, and therefore several educational teaching platforms have been designed over the past years [F4, F14, F10, F13] to help out the learning process of students and encourage them to pursue a career in information security.

These platforms provide an environment that can be used for teaching and training purposes in cyber security courses by simulating real vulnerable systems and supporting complex cyber scenarios, as well as training users in monitoring and defending cyber infrastructure against malicious activities. This allows students to execute attacks on these systems without harming actual systems, and to obtain both an offensive and defensive perspective of security practices.

Most of these platforms are designed to support a common exercise format, Jeopardy Capture-The-Flag (CTF) [E3] which nowadays is a well known format to demonstrate the user skills in solving cyber security tasks and problems. In this format, the participants are tasked to find *flags* (i.e. a hidden string of text) by successfully exploiting vulnerable computer systems, being awarded points for each flag they discover. CTF competition brings several advantages when applied in teaching environments [F10]. For example, these types of competitions allow students to legally hack systems in a safe environment, by identifying vulnerabilities and trying to compromise them, while also allowing them to learn to defend against attacks [F19]. CTFs have also been shown to be effective in keeping the students engaged by their hands-on nature and through the entertaining experience [F9]. When involved in a CTF, being able to work as a group in a team is important for students to achieve their goals, leading the students to improve their communication skills, and to share, compare and broaden their knowledge [F18]. Moreover, challenge based learning stimulates the development of problem solving skills leading the students to be involved in finding better solutions [F5].

Although these existing educational teaching platforms contribute to the learning process, none of them automate the process of creating custom vulnerable environments for teaching classes for high schools students. This led Aalborg University, Denmark, to develop the first version of Haaukins [F11], an educational tool used for conducting short training events at high schools. These events, usually running between two hours up to a few days, were intended to engage students without prior information security knowledge, in a new topic, and to generate interest in a future career into the security field.

Since its launch, Haaukins has proven to be successful in high schools and attracted attention from other Danish institutions within higher education. In order to increase the usage of the platform, Aalborg University together with other educational institutions has started to adapt Haaukins from both a technical and an education perspective to accommodate IT and engineering students within higher education. The main contributions of this paper are as follows:

- Define a formalization of the differences in teaching environment, across high school and higher levels of education, for cyber security education.

- Present a set of design goals, based on the formalization, that the Haaukins platform is required to adopt to address these differences.

- Develop a solution for these design goals, that have been integrated into the existing open source platform.

The rest of this paper is organized as follows. A background of the initial design of Haaukins is presented in Section F.2, followed by a comparison with the new target audience in Section F.3. In order to adapt the platform to the new target audience, a list of design principles is presented in Section F.4, which is followed by the fulfilment of those principles in Section F.5. The platform deployment is presented in Section F.6, followed by the conclusion in Section F.7.

F.2 Background

Besides existing educational platforms there are a variety of commercial platforms where it is possible to practice cyber security in a CTF format, including 'Hack the Box' [F8] and the more recently developed 'Try Hack Me' [F15]. Whereas 'Hack the Box' provides challenging scenarios fitting for a more experienced target audience, 'Try Hack Me' provides several learning paths suitable for introducing beginners to the basics of security. However, such commercial platforms generally have no option to expand upon the training material and scenarios provided (such as adding more vulnerable machines), which is a severe restriction when teaching courses according to specific learning goals and objectives. An educational institution must have the opportunity to tailor the teaching material to their curriculum, and as such cannot rely on closed platforms.

'PicoCTF' [F3] is a platform developed by Carnegie Mellon University that achieved success during the past years. In order to encourage cyber security interest among high school students, it provided an interactive game and a terminal user interface used to interact with the exercises. Similar to Haaukins, it is open source and has it has been created for high schools students. However, in contrast to Haaukins, it constraints itself to exercise types which students can do on any computer systems, and without relying on professional tooling.

Setting up an environment composed of computer systems, vulnerable hosts and connections between them is time-consuming and errors can be hard to handle. Haaukins aims to facilitate the learning process by helping teachers to automate the tedious setup and management of those environments, by making them accessible with no prior, complex configuration. This allows students to have their own virtual and isolated environment to practice cyber security skills, while having the convenience of accessing it from their own devices simply through any web browser. High school students who, due to their limited amount of experience with cyber security and a limited understanding of underlying computer science topics in general, need an automated and highly accessible way to access those environments. Moreover, in order to support other educational institutions in conducting short CTF workshops, Haaukins has been made available as an open-source project on GitHub[1] with a GNU GPLv3 license.

Haaukins is implemented as a Jeopardy-style CTF platform, in which the teacher can create an event and students access their own individual environments, referred to as *lab*s, created within the scope of that event. Each virtual *lab*, accessible through a student's laptop, consists of a computer network that has multiple computer systems, which are under control of the student or student group. Within this *lab*, students have to work towards solving a set of *exercises*, where each of them are related to a specific concept within information security. Most of the exercises are designed with specific learning objectives in mind. Exercises are solved by exploiting vulnerabilities of computer systems in the virtual labs; vulnerabilities that are intentionally embedded in the computer systems by the exercise developer. In order to exploit these vulnerabilities, the students must understand the underlying problem with the software, which makes these exercises a valuable teaching tool. Each event can be composed of any set of exercises that the teacher wishes to use in that session. Figure F.3.1 illustrates the interaction between teacher, students and Haaukins itself. A website, which is automatically generated per event, provides both teacher and students with information about the event and its exercises.

The high school target audience required four main design goals to be fulfilled, that shaped the development of Haaukins in the very early stage of its existence. Those design goals are described in [F11] and are summarized as follow:

Fully Automated Haaukins was designed to automate the lab configuration process completely, and to do so in a relatively short time, making the preparation of an event painless.

The automation process will start and connect the required instances and components in order to have an environment available for the learning aspects. This process eliminates the need for manual configuration and provide error handling of labs for the teachers.

Transparent As students are more inclined to share solutions than in a competition setup, potentially leading to cheating that negatively influences their learning experience, a unique way of creating labs has been provided in Haaukins. For each lab, the flags to be found are unique, reducing the possibility of sharing flags. In addition, the platform monitors all actions that students make within it, allowing for the analysis of this afterwards.

Highly Accessible Given the short nature of the training events at high schools, it was considered imperative that students spend as little effort as possible on establishing access to their labs. In this case, Haaukins allows participants to access their virtual

[1] https://github.com/aau-network-security/haaukins

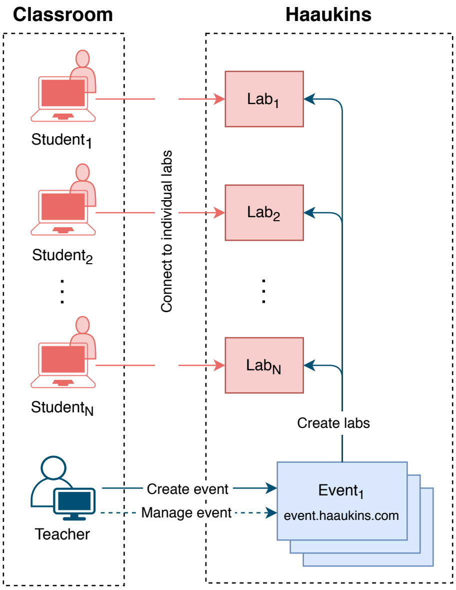


Fig. F.3.1: The interaction between a teacher, a classroom full of students and the Haaukins components in the first version of the platform

lab (1) quickly (i.e. in a matter of minutes), (2) without prior knowledge of accessing a virtual platform and (3) independently of their physical location and the operating system they use (i.e. access from anywhere via their Internet browser).

Realistic In order to teach students skills that are usable outside of the classroom, the platform was required to reflect a real world setting. The existing pool of exercises was thus designed to replicate real-life situations, and the platform had to support these types of exercises. In Haaukins all the exercises created are from real-life situations and interaction's realism is kept in order to ensure that students are gaining useful skills. Additionally, the taught techniques and available tool kit are identical to those used by security professionals.

F.3 Target audience

During the last year, the usage of Haaukins increased because of interest from new educational institutions. From the feedback received after every event, we have realised that the platform was able to satisfy the design goals allowing teachers to successfully conduct cyber security courses. The increase in usage has brought more visibility to the platform, thus leading to new expectations and new plans for its continuous development. Haaukins, could potentially be used in higher education as well, however due to the different target audience, it would need several improvements in order to make the new users comfortable with the platform.

This section seeks to clarify details of the new target audience, and their distinct traits, when compared to the traditional target audience for Haaukins. As previously stated, the audience intending to use the adjusted platform is *students in higher education*. To further clarify the expected skill set of this group, we define it as students who have attended at least one semester in computer science, computer engineering or comparable educations at least at undergraduate level. Concretely, students within this group should have a novice level of understanding of fundamental computer science topics, such as: (1) networking, (2) operating systems, and (3) programming. This thereby puts this group of students, in terms of competences, ahead of the original target audience of Haaukins (i.e. high school students).

Consequently, the established experience of the new target audience is reflected in their ability to interact with and understand computer systems.

Trait 1. Higher education students are expected to have preliminary knowledge of various tools, selected to their preference, to diagnose and interact with computer systems and computer networks.

This is an important trait that differs from the original high school setting, where students are expected to have no or very limited technical understanding. This additional knowledge is also reflected in the type of teaching material to be used for the new target audience.

Trait 2. With a deeper technical knowledge, students of higher education are able to work on more complex exercises that involve understanding and attacking highly-interconnected and complex multi-computer systems.

From a teacher perspective, designing such exercises can be challenging and time consuming, as ensuring the interconnection of a multitude of computers correctly is a complex task. These exercises typically encapsulate some intended-by-design vulnerabilities, to be found by the students, and the discovery of the given vulnerabilities serve as an objective of the exercise. Students are typically expected to interact with the computers of the systems, of the exercise, in an offensive and destructive manner. Therefore exercise designers must avoid unexpected vulnerabilities in the systems to be attacked, since such unexpected vulnerabilities can potentially halt the completion of the intended exercise. From a course point of view, the duration itself is often longer than the high schools events, and a class might run just once per week through a semester. As a consequence of this, reliability and availability must be provided even for such long term events, so the users of the platform do not experience interruptions and resets of events, users or exercise progress.

Trait 3. Given the multitude of attack vectors that a computer system can have, with the typical length and size of a university course, keeping each students lab in a healthy state (ability to complete exercises) is challenging.

The identified and distinct traits of the new target audience captures the challenges sought out to be addressed by technical solutions that can be integrated into the existing Haaukins platform, thus expanding its application domain. It is important that the application domain is expanded rather than changed, and that Haaukins still fulfill the requirements for the original target audience. The improved platform thus also appeals to the groups in-between, i.e. students in higher education without the competences listed above.

F.4 Design Principles

The three traits identified in Section F.3, serve as the problems which have to be addressed in order for Haaukins to viable for the new teaching context. Prior to the technical design, we seek to put forward three design principles that will drive the changes to the existing platform. These design principles have to coexist with existing ones (covered in Section F.2), thus their compatibility to the existing ones are discussed.

Haaukins provides rapid access to students' labs remotely (through a web browser) to a computer with pre-installed tools, serving as a fixed toolbox, used for accessing the lab environment and interacting with the exercises. This design was implemented in relation to the existing design principle of being highly accessible. However, the design of having a fixed toolbox (static tooling) could potentially violate Trait 1, as the desired set of tools might not match the provided set of tools. In order to address this problem, and in contrast to the existing static tooling currently provided, we present the following design principle:

Design Principle 1. [Dynamic Tooling] Students should be able to use their desired set of tools within their lab of exercises.

Having dynamic tooling could potentially be an undesired feature for the original teaching context (high schools), which includes more novice students that have less preliminary competences in the field. Thereby, it is important to ensure that this principle serves as an alternative, such that it can coexist with the current method of "static tooling". Knowing that the teaching context includes students with more preliminary knowledge (Trait 2), forces the exercises containing computer systems to become larger (groups of computers) and more inter-connected (network communication). These types of exercises are naturally costly to design due to their complexity, and the current architecture for Haaukins relies on teachers, on an individual level, to implementing these. Thereby, as a measure to reduce this cost across, we propose the following design principle that is based on sharing resources:

Design Principle 2. [Centralization of Exercises] Complex exercises should be provided by a centralized source, such that teachers across various institutions can share implementations of exercises, thus enabling future teachers to benefit from existing work.

Hosting and serving these exercises in individual labs for each student in a university class is a vastly different scale when compared to the high school setting (Trait 3). Moreover the platform has to support a higher volume of concurrent students coming from both the previous and new target audience. Scaling the platform to support a teaching context of a university class (more than 100 students) and an increased amount of events running at the same time, is a two-fold challenge. Indeed, both the technical capabilities (efficiency, resources) and orchestration of labs (maintenance, restarts) are required to scale to this new volume of students. Moreover teachers should have the possibility to monitor and manage student labs from a highly user friendly interface while the events are running.

Design Principle 3. [Scalability] The platform should provide both computational efficiency (technical scaling) and orchestration features (teaching scaling) that ensure exercises within labs remain healthy and available while several events are running concurrently.

F.5 Evolved Haaukins platform

The design principles defined in the previous section aim at improving Haaukins in order to make the new target audience comfortable in using the platform. The evolved platform should address the requirements of students in higher education without impacting the usability for high schools, which are still a core user group of Haaukins. This chapter presents a list of improvements implemented based on each of the principles defined in F.4.

Dynamic Tooling. In order to adhere to Design Principle 1, two alternatives were considered; either to integrate more tools in the lab or to provide an alternative access method to the lab. In the first approach, a more customizable lab environment can be provided to the users, e.g. by giving the teachers or students themselves the option to compose a toolkit from a curated list of tools and operating systems. This curated list would have to encompass all possible tools that students would like to use, which in practise would be difficult to accomplish. The second alternative instead relies on giving the students access to the lab through a different method than the browser-based method. Previously, students would remotely control a computer system prepared specifically for this, and these systems were identical across all labs in an event, leaving little room for customization. Instead, students could be given access to a *network connection* to the lab, thereby opening the option for students to connect their own computer systems to the labs, with their own custom tooling. In order to respect all previous and newly defined design principles, the platform must allow the user to still have a fully automated configuration process and a highly accessible way to the exercises.

From a technical point of view, the choice fell on the integration of a Virtual Private Network (VPN) [F7], and specifically Wireguard [F6], in which a secure connection to another network over the Internet is created, in our case to the *lab*. A VPN connection can be established from any operating system from any geographic location, merely requiring the students to configure their local Wireguard with a configuration file provided by Haaukins. Similar to the previous web-based access method, a VPN connection can generally be established from computer networks without requiring changes to the IT infrastructure. As a result, this solution does not violate the original highly-accessible design principle, but contributes to Design Principle 1. Although configuring a VPN is considered a fairly simple setup step for an experienced student, this it not the case for the traditional target audience, and therefore Haaukins supports both (1) the web-based access method and (2) the VPN connection, and a teacher can make a per-event decision as to what access method suits the target audience best.

Centralization of Exercises. The first version of the platform has been developed to allow teachers to run events using either custom exercises or readily-available exercises provided by the platform itself through an exercise library. In the first case, teachers could create specific exercises for their courses and use them in their events in order to teach different topics not already provided by the platform. Although this feature brought more flexibility to the platform when referring to high schools, university teachers could not benefit much of it. This is largely due to the fact that different target audiences rely on different exercises that differ from each other in terms of both content and complexity (Trait 2). Exercises for higher education students are not only harder to solve compared to those for high school students, they are also more complex and take longer more time and efforts to create. These exercises, in fact, have to be created either focusing on a specific topic and go into details or have a broad approach where it covers several topics. In both cases, the teacher has to design and create several steps of difficulty in order to let students improve their skills and time spent on the exercise.

Creating an exercise for a course is time-consuming, especially for the more advanced exercises which are to be used within higher education, and therefore is not always an option. To support teachers, many exercises have been created and made available to the use in their events, and a clear workflow has been established for creating, testing and including exercises for those who want to create their own. The main goal was to provide a centralized pool of exercises with different content and of different complexities where teachers can choose according to their courses (Design Principle 2). Each exercise is accompanied by a description, and a list of prerequisites and outcomes has been made in order to facilitate teachers in choosing a relevant composition of exercises for their courses. Finally, to facilitate an even better exercise selection phase, the exercises have been grouped based on different difficulty levels (e.g. the number of steps needed to solve it and the topic covered) and divided into different categories that cover different fields of cyber security (e.g. web exploitation, forensics, binary, reverse engineering and cryptography).

Scalability. Haaukins must also support managing a higher amount of events running at the same time. In order to provide a reliable and fault tolerance platform, it has to scale in two main directions (Design Principle 3) described as follows.

Teaching Scaling In order to maintain the labs healthy and available, a "reset functionality" has been created, available to both teachers and students in order to restart (i.e stop and start) *lab*s as well as individual exercises in case of crashes, which can happen if a student make mistakes when attempting to solve an exercise or when exercises are not properly developed with the destructive behaviour of the teaching context in mind (Trait 3). In such cases, one or more exercises in the *lab* are reverted back to the initial state right after the lab was created. The students lose their progressions towards this specific exercise, but it allows them to experiment with potential destructive offensive techniques that will break the exercise. In fact, as exercises largely focus on breaking existing computer systems, those systems are brought to a high degree of stress. As such, platform allows a graceful recovery from errors, instead of burdening teachers with developing bulletproof exercises. This functionality has been made available on the event website for the students while for teachers it has been implemented in the *web client*.

In the previous version of Haaukins, the event creation and management controls were provided via a command-line interface (or *cli*), that had to be downloaded and installed on the computer of the teacher. This command-line program could be used to send some basic commands remotely to the physical server on which Haaukins was running, thereby managing events. Prior to be able to use this program, a teacher had to be granted access by another teacher as a security mechanism, which introduced another barrier for quickly setting up an event. Feedback from high school organizers showed that this approach was not user-friendly enough, and that it was too time-consuming to use.

In order to overcome this issue a different way to interact with the platform had to be provided, and a user interface *web client* connected to the platform was created. In detail, the *web client* is the web-application version of the *cli* which provides the same functionalities of the *cli* along with a number of new functionalities designed to improve scalability as well as the overall organisation experience. With this solution it is no longer needed to download and install the *cli* on the teachers computer machine. Teachers can access the *web client* upon request in order to create and manage their own events no matters where they are - simply by their web browser of choice. From an intuitive user interface, the teachers are able to choose the event configuration (e.g. event name, event capacity, exercises and VPN option) and check the status of the teams signed up in their events.

The *web client* is linked to the centralized exercises pool thus allowing teachers to insert new custom exercises and get all information about already existing and available exercises. This connection aims to facilitate the teacher in choosing the relevant exercises for his or her event.

Besides management of events and their respective *lab*s, Haaukins has the ability to monitor and log student's interactions with the platform, which enables the ability to identify if the participants become stuck while exploiting exercises. This functionality is only activated if consent is granted from individual students, and is implemented by storing the stream of key presses to log files, that serve as the basis of the analysis of behavior.

Technical Scaling The new target audience will bring with it not just more events running simultaneously on the server, but also larger events due to the higher number of students for each course, thus leading to a higher computation load on the platform. A potential problem that might occur because of this higher demand, is that the platform might not have sufficient capacity to be able to manage all the events thus leading to the rejection of some of them. A main goal is therefore to provide both target audiences with a platform that is able to support all the requested events without affecting the performance of the platform or other events (Design Principle 3).

To meet this goal, two main approaches to make the platform more scalable have been evaluated, and both horizontal and vertical methods have been taken into consideration: the horizontal approach relies on replicating the platform on multiple servers, thus leading to a distributed version where events can run in different servers. The vertical approach instead consists of adding more resources (i.e. memory and hard disk) to the current server in order to make it more powerful. From our point of view both methods where suitable for the platform, the former being more expensive in terms of time due to the refactoring the code base of the platform but cheaper in terms of the monetary cost, while for the latter it is the other way around.

Also considering the possibility to make a platform cloud based, more effort has been made in making the platform available in a distributed way without affecting the usability for teachers and students. In this sense the platform has been split in microservices [F2] running on different servers, which also allows for a more easier deployment to the cloud in the future [F12]. The vertical approach has been applied as well on the main server, where more resources have been installed.

From previous experience with high school events, it was found that labs in events that last longer than two weeks have a far lower resource utilization (i.e. the percentage of time that a lab is actively being used) than shorter events. In fact, some of the labs were not used for several full days before being used again afterwards, occupying resources of the host server and consequently potentially refraining other students from using the platform. As described in Trait 3, this scenario might occur more often, eventually denying requests for events due to the limited capacity of the server hosting the platform. Although the technical scaling improvements aim to provide the opportunity to everyone to use the platform, those long events might thus cause a problem. To overcome this issue, a 'sleep mode' feature has been developed, which automatically suspends *lab*s that have not been used for a while and resume them when the students log into the event again. This feature aims to save resources on the server - especially for the new usage - and thus boosts the scalability of the platform.

F.6 Deployment in Higher Education

Throughout the development of the evolved Haaukins platform, it has been used in various settings within higher education, and feedback has been collected as input to the development process. The usage includes courses within two universities and four university colleges, as well as larger events in the framework of higher education, such as summer schools and conferences. It was also used in university-facilitated events for IT professionals in companies including sectors such as finance, energy, IT and national authorities.

While different events and courses were organised differently, in general three steps were included: (1) In the preparation phase, the course or event was planned and set up. This includes choosing relevant exercises, determining whether VPN or web browser access should be used. In the beginning, this was in most cases done in close collaboration with the Haaukins developers, but as more teachers gained experience in using the platform and as the improvements described in this paper were developed - in particular the *web client* - this was increasingly done by the teachers independently. (2) In the next phase, the event/course was held. Unless any problems arose, this was usually done by the teachers. (3) Finally, in the evaluation phase, feedback was collected from the teachers and/or the students.

The feedback collected included the experience from both phase (1) and (2) together with general feedback and suggestions about the platform. These collection of feedback has served two purposes. One purpose was to use it for input to the overall platform design and development, where the resulting changes would be of a more fundamental character. Such changes would be incorporated to the overall development road map, which was discussed among partner institutions of higher education at regular meetings. Another purpose was to identify issues where smaller adjustments could improve the user experience. In many cases, these were straightforward to implement, e.g. better explanations of platform usage and exercises. By the time this paper was submitted, Haaukins has been used by more than 1.000 students from different target groups.

F.7 Conclusion and Future Work

In this work, we presented an evolved version of Haaukins, a cybersecurity training platform that facilitates the learning process by helping teachers in creating cybersecurity training scenarios in a secure, closed and virtualized environment. Over the last years, the platform has been used in several high schools with consistently positive feedback and it was decided to improve the platform in a way that would support the usage in higher education.

The new target audience, compared to the previous one, has been identified as a more experienced student who is able to interact with computer systems and computer networks using various tools, and who is able to address more complex exercises. Due to those differences and the typical length and size of a university course, a list of design principles, which have to coexist with the existing ones, have been defined and afterwards shaped into technical improvements on the platform.

Examples of such improvements include that a VPN connection has been provided as an alternative way to connect the *lab*s, thus enabling the students to use their own tools. An exercises pool has been made available for teachers in order to let them benefit from already made exercises, thus avoiding the time invested in creating them. Finally, the platform has been made more scalable in order to handle a higher amount of students and longer events running at the same time.

These collective changes made Haaukins a platform for both students of higher education and high schools students, driven by an increased interest by schools in Denmark. The platform is currently being under further development to widen its appeal to even more target audiences and to be used on a larger scale, and additional research studies are also being undertaken: In order to obtain a better understanding of the challenges of the game based learning experience, we are planning to conduct user studies in the near future. Moreover an investigation of which exercises should be developed to ensure a good progression in the learning path of different students will be carried out.

F.8 Acknowledgment

The authors would like to thank The Danish Industry Foundation for the continued support to the development of Haaukins.

References

[F1] European Commission. 2018. *Resilience, Deterrence and Defence: Building strong cybersecurity in Europe*. 2018. URL: `https://ec.europa.eu/digital-single-market/en/news/resilience-deterrence-and-defence-building-strong-cybersecurity-europe`.

[F2] Nuha Alshuqayran, Nour Ali, and Roger Evans. "A systematic mapping study in microservice architecture". In: *2016 IEEE 9th International Conference on Service-Oriented Computing and Applications (SOCA)*. IEEE. 2016, pp. 44–51.

[F3] Carnegie Mellon University. *picoCTF*. URL: `https://picoctf.org/`.

[F4] Peter Chapman, Jonathan Burket, and David Brumley. "PicoCTF: A game-based computer security competition for high school students". In: *2014 {USENIX} Summit on Gaming, Games, and Gamification in Security Education (3GSE 14)*. 2014.

[F5] Ronald S Cheung et al. "Challenge based learning in cybersecurity education". In: *Proceedings of the International Conference on Security and Management (SAM)*. The Steering Committee of The World Congress in Computer Science, Computer ... 2011, p. 1.

[F6] Jason A Donenfeld. "WireGuard: next generation kernel network tunnel (2018)". In: *URL https://www. wireguard. com/papers/wireguard. pdf. version 416d63b* (2018), pp. 06–30.

[F7] Paul Ferguson and Geoff Huston. *What is a VPN?* 1998.

[F8] Hack The Box Ltd. *HackTheBox.eu*. URL: `https://www.hackthebox.eu/`.

[F9] Menelaos Katsantonis, Panayotis Fouliras, and Ioannis Mavridis. "Conceptual analysis of cyber security education based on live competitions". In: *2017 IEEE Global Engineering Education Conference (EDUCON)*. IEEE. 2017, pp. 771–779.

[F10] Alexander Mansurov. "A CTF-based approach in information security education: an extracurricular activity in teaching students at altai state university, russia". In: *Modern Applied Science* 10.11 (2016), p. 159.

[F11] Thomas Kobber Panum et al. "Haaukins: A Highly Accessible and Automated Virtualization Platform for Security Education". In: *2019 IEEE 19th International Conference on Advanced Learning Technologies (ICALT)*. Vol. 2161. IEEE. 2019, pp. 236–238.

[F12] V. Singh and S. K. Peddoju. "Container-based microservice architecture for cloud applications". In: *2017 International Conference on Computing, Communication and Automation (ICCCA)*. 2017, pp. 847–852. DOI: `10.1109/CCAA.2017.8229914`.

[F13] Llanos Tobarra et al. "Game-based Learning Approach to Cybersecurity". In: *2020 IEEE Global Engineering Education Conference (EDUCON)*. IEEE. 2020, pp. 1125–1132.

[F14] Erik Trickel et al. “Shell we play a game? CTF-as-a-service for security education”. In: *2017 {USENIX} Workshop on Advances in Security Education ({ASE} 17)*. 2017.

[F15] Try Hack Me - London. *Try Hack Me*. URL: `https://tryhackme.com/`.

[F16] Aalborg University. *Cyber Security, MSc in Engineering*. 2020. URL: `https://www.en.aau.dk/education/master/cyber-security/`.

[F17] Cybersecurity Ventures. “Cybersecurity Jobs Report”. In: *Herjavec Group* (2017).

[F18] Joseph Werther et al. “Experiences in cyber security education: The MIT Lincoln laboratory capture-the-flag exercise.” In: *CSET*. 2011.

[F19] SeongIl Wi, Jaeseung Choi, and Sang Kil Cha. “Git-based {CTF}: A Simple and Effective Approach to Organizing In-Course Attack-and-Defense Security Competition”. In: *2018 {USENIX} Workshop on Advances in Security Education ({ASE} 18)*. 2018.

Part III

Epilogue

Chapter 10

Conclusion

Philosophically, one can claim that the strength of a defense is directly constrained by the level of precision that expected attacks can be defined. This links to the fact that uncertainty of ground truth limits the true performance of separating attacks from benign activities. Fundamentally, this dilemma remains a challenge for detecting certain types of deception attacks, as these attacks are vaguely defined [31] and dependent on social influences of individuals [12].

An established form of deception attacks, phishing, is actively performed using web applications that are expected to be accessed through a victim's web browser. Web browsers, and the web as a whole, are reliant on variety protocols and parsers for displaying web pages to users. This challenges the ability to obtain detailed information of web pages, as it largely requires to mimic the behavior of a web browser. We present a tool for gathering detailed information of web pages, Kraaler (Paper A). This enabled the ability to gather a data set with extensive information of web pages, which served as a foundation for the following research.

Assessing influential and recent detection methods of phishing attacks, it has been demonstrated that strategies adopted by these methods typically rely on properties which are likely to be associated with symptoms of attack, opposed to properties that cause the attack to be functional. The set of properties is typically derived using some form of analysis, or learned from empirical observations using statistical modeling techniques, such as machine learning. Consequently, it have been shown that adversaries are able to search for (and find) modifications of existing phishing attacks, that maintains the attack's *true* functional properties (such that the attack remain unchanged for the victim), while obtaining properties that enable the ability to circumvent detection. This led to the specification of a series of axioms, as an effort to describe abstract classes of properties that are desirable for the detection techniques to incorporate. Additionally, a set of design guidelines were proposed to ensure better evaluations and design of detection techniques that seek to become robust.

During this assessment, there were indications of that a state-of-the-art method, VisualPhishNet [2], that relied on highly-parameterized neural networks, that sought out to improve adversarial robustness, were seemingly less robust than originally reported. VisualPhishNet were designed using a machine learning technique for out-of-distribution classification named *Deep Metric Learning* (DML), which effectively seek to use highly parameterized neural networks models to yield abstract representations that are suitable for comparing visual appearance of web pages. The authors acknowledge that similar neural networks have, in other contexts such as a traditional classification, been demonstrated to be vulnerable to attacks of imperceptible perturbations that drastically alter the output of the model for a given input. Following this, a measure of robustness towards such an attack was presented using an adaptation of existing attacks. This adaptation did not account for certain properties of DML models, most notably that common loss functions (used for training and the adapted attack) have an interdependence between data points unlike loss functions used in the domain of existing attacks.

Thus we sought out to explore the ability to create effective attacks and obtain models robust to potential attacks for deep metric learning, while accounting for the unique properties of deep metric learning. This led to a formalization of attacks and three robust optimization objectives for DML models, that are usable across any commonly used ℓ_p-norm. The proposed method of attack enables the use of existing first-order attack methods, and demonstrates that previous methods (VisualPhishNet) maintained lower robustness than initially reported. To address this fact, using one of the proposed robust optimization objectives to train robust DML models across three common data sets and the application of VisualPhishNet, yields an improvement in robustness that exceeds prior methods.

Having deception attacks that are difficult to express with high precision, it remains natural to select solution methods that seek to automate the characterization of attacks using statistical patterns from empirical observations. However, as researchers or engineers adopt highly parameterized models (to provide model flexibility), they also adopt the established potential issues of such methods. These potential issues can be exemplified by presence of bias, violations of assumptions of distributions at test-time time, and mentioned lack of robustness. Ensuring the absence of these potential issues can be infeasible, and their presence can cause detection methods for deception attacks to misclassify true attacks.

Haaukins (Security Education Platform) Platforms for hosting computer systems to be deliberately victims of attacks have typically been guided by competitions for professionals. During expansion of information security related courses, we discovered that these existing solutions were unfit for an educational setting and that alternative educational platforms had poor realism. Since the first implementation of Haaukins more than 1000 students have been using the platform, it has attracted more than €3M of external funding, and have been presented at prominent venues such as Black Hat Europe. This suggests that the initial identified needs were genuine and that the design of platforms for security education remain an open problem.

Chapter 11

Directions for Future Research

With deception attacks being dominated by informal definitions, (i) one can seek to formalize their characteristics or (ii) improve upon methods that seek to characterize the attacks from empirical observations. Given that the existence of attacks have been known for multiple decades, and certain types of deception attacks [31] achieved numerous informal definitions within this time, it remains unlikely that stronger formalism is a plausible solution. For problems that remain difficult to formalize, machine learning techniques have in recent years been a highly dominant solution. These techniques, particularly the highly parameterized models, remain susceptible to small adversarial perturbations that cause undesired predictions. Numerous efforts to address this problems have been proposed, however, most fail to achieve reliable improvements [4, 58]

Adversarial training has shown to effectively enhance the robustness towards these adversarial perturbations, that comes with the cost of lower benign accuracy [38]. Importantly, models trained using adversarial training does not achieve desirable robustness that is comparable to performance on benign input. However, the effect of adversarial training has been demonstrated to influence fundamental properties of the learned models, namely: (i) more effective transfer learning [52, 61], (ii) more interpretable gradients [60, 24], (iii) presentations that are more aligned with human perception [18], (iiii) and better generalization [67, 68].

Considering these effects, it could suggest that the learning paradigm (Empirical Risk Minimization [62]) might contain undiscovered effects that hinders robust learning. Thus improvement in robustness is achieved by addressing fundamental properties of the current paradigm, or through the design of training procedures for an alternate learning paradigm.

Another promising area of research to improve robustness is *randomized smoothing*, which involves exploiting the fact that models tend to perform well over certain types of noise [13, 34, 15]. Effectively this approach seeks to smooth decision boundaries using some form of noise (e.g. Gaussian noise), such that adversarial perturbations that exploit “spikes” in the non-smoothed decision boundary reside on the correct side of the decision boundary under smoothed conditions.

Schott et al. [54] have demonstrated that using conditional generative models during inference can significantly improve the robustness for the particular application of MNIST [33]. The methodology has since then been extended to other application domains [23], and reaches state-of-the-art performance in those settings.

Despite the covered fragility of modern machine learning techniques having been known for over a decade [8], the progress on obtaining robustness have been limited and in certain occasions false [4, 58]. The covered recent research directions seems to yield reliable progress on this topic, despite being far from a solved problem. Nicholas Carlini, an influential research within the field of adversarial robustness, have at numerous occasions referred to the state of adversarial robustness as an analogy to the state of cryptography prior to Shannon's introduction of information theory [1]. This analogy emphasize his perception of adversarial robustness (for machine learning) is in its early stages, and fundamental theorems are yet to be uncovered. Given that these methods are a corner stone for future detection solutions, the ability to obtain robust detection remain uncertain and likely not to be designed within the near future.

Google claims to have effectively eliminated the effectiveness of certain deception attacks (phishing) by enforcing two-factor authentication [27]. This solution does effectively prevent attacks, in respect to Lastdrager's definition [31], but ensure the value of the information exchanged during an attack to be short-lived. Despite the demonstration of effectiveness, it could indicate that adversaries have not adopted to the new enforced deadlines required to exploit the information, which still remain possible with the use of automation.

Deception attacks will most likely remain an efficient attack for adversaries for years to come, as defenses struggle to become robust. Accepting this fact, one can seek lower the value of the information exchanged during an attack to reduce the incentive of adversaries.

[1] Talks listed on Nicholas Carlini's own website: https://nicholas.carlini.com/talks

References

[1] Greg Aaron. *Phishing Acitivity Trends 4th Quarter 2020*. Tech. rep. Anti-Phishing Working Group APWG, 2020. URL: `https://docs.apwg.org/reports/apwg_trends_report_q4_2020.pdf`.

[2] Sahar Abdelnabi, Katharina Krombholz, and Mario Fritz. "VisualPhishNet: Zero-Day Phishing Website Detection by Visual Similarity". In: *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*. 2020, pp. 1681–1698.

[3] Saeed Abu-Nimeh et al. "A comparison of machine learning techniques for phishing detection". In: *Proceedings of the anti-phishing working groups 2nd annual eCrime researchers summit*. 2007, pp. 60–69.

[4] Anish Athalye, Nicholas Carlini, and David Wagner. "Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples". In: *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*. July 2018. URL: `https://arxiv.org/abs/1802.00420`.

[5] Anish Athalye et al. *RobustML*. `https://www.robust-ml.org/`. Accessed: 2020-07-03. 2018.

[6] Anish Athalye et al. "Synthesizing robust adversarial examples". In: *International conference on machine learning*. PMLR. 2018, pp. 284–293.

[7] Marco Barreno et al. "Can machine learning be secure?" In: *Proceedings of the 2006 ACM Symposium on Information, computer and communications security*. 2006, pp. 16–25.

[8] Battista Biggio and Fabio Roli. "Wild patterns: Ten years after the rise of adversarial machine learning". In: *Pattern Recognition* 84 (2018), pp. 317–331.

[9] Battista Biggio et al. "Evasion attacks against machine learning at test time". In: *Joint European conference on machine learning and knowledge discovery in databases*. Springer. 2013, pp. 387–402.

[10] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[11] Tom Brown et al. "Adversarial Patch". In: 2017. URL: `https://arxiv.org/pdf/1712.09665.pdf`.

[12] Jan-Willem Hendrik Bullée et al. "On the anatomy of social engineering attacks—A literature-based dissection of successful attacks". In: *Journal of investigative psychology and offender profiling* 15.1 (2018), pp. 20–45.

[13] Xiaoyu Cao and Neil Zhenqiang Gong. "Mitigating evasion attacks to deep neural networks via region-based classification". In: *Proceedings of the 33rd Annual Computer Security Applications Conference*. 2017, pp. 278–287.

[14] N. Carlini and D. Wagner. "Towards Evaluating the Robustness of Neural Networks". In: *2017 IEEE Symposium on Security and Privacy (SP)*. 2017, pp. 39–57.

[15] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. "Certified adversarial robustness via randomized smoothing". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 1310–1320.

[16] Francesco Croce et al. *RobustBench: a standardized adversarial robustness benchmark*. 2020. arXiv: `2010.09670 [cs.LG]`.

[17] Tommaso Dreossi, Somesh Jha, and Sanjit A Seshia. "Semantic adversarial deep learning". In: *International Conference on Computer Aided Verification*. Springer. 2018, pp. 3–26.

[18] Logan Engstrom et al. "Adversarial Robustness as a Prior for Learned Representations". In: *arXiv e-prints* (2019), arXiv–1906.

[19] Kevin Eykholt et al. "Robust physical-world attacks on deep learning visual classification". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 1625–1634.

[20] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. `http://www.deeplearningbook.org`. MIT Press, 2016.

[21] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[22] Markus Huber et al. "Towards automating social engineering using social networking sites". In: *2009 International Conference on Computational Science and Engineering*. Vol. 3. IEEE. 2009, pp. 117–124.

[23] An Ju and David Wagner. "E-ABS: Extending the Analysis-By-Synthesis Robust Classification Model to More Complex Image Domains". In: *Proceedings of the 13th ACM Workshop on Artificial Intelligence and Security*. 2020, pp. 25–36.

[24] Simran Kaur, Jeremy Cohen, and Zachary C Lipton. "Are Perceptually-Aligned Gradients a General Property of Robust Classifiers?" In: *arXiv e-prints* (2019), arXiv–1910.

[25] Jack Kiefer, Jacob Wolfowitz, et al. "Stochastic estimation of the maximum of a regression function". In: *The Annals of Mathematical Statistics* 23.3 (1952), pp. 462–466.

[26] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

[27] Brian Krebs. *Google: Security Keys Neutralized Employee Phishing*. `https://krebsonsecurity.com/2018/07/google-security-keys-neutralized-employee-phishing/`. Accessed: 2021-01-20. 2018.

[28] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *NIPS*. 2012.

[29] Ram Shankar Siva Kumar et al. "Adversarial machine learning-industry perspectives". In: *2020 IEEE Security and Privacy Workshops (SPW)*. IEEE. 2020, pp. 69–75.

[30] Ram Shankar Siva Kumar et al. "Failure modes in machine learning systems". In: *arXiv preprint arXiv:1911.11034* (2019).

[31] Elmer Lastdrager. "Achieving a Consensual Definition of Phishing Based on a Systematic Review of the Literature". In: *Crime Science* 3.1 (2014), p. 9. DOI: 10.1186/s40163-014-0009-y.

[32] Y. Le Cun et al. "Handwritten Digit Recognition with a Back-Propagation Network". In: *Proceedings of the 2nd International Conference on Neural Information Processing Systems*. NIPS'89. Cambridge, MA, USA: MIT Press, 1989, pp. 396–404.

[33] Yann LeCun and Corinna Cortes. "MNIST handwritten digit database". In: (2010). URL: http://yann.lecun.com/exdb/mnist/.

[34] Xuanqing Liu et al. "Towards robust neural networks via random self-ensemble". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 369–385.

[35] Nikos K. Logothetis and David L. Sheinberg. "Visual Object Recognition". In: *Annual Review of Neuroscience* 19.1 (1996). PMID: 8833455, pp. 577–621. DOI: 10.1146/annurev.ne.19.030196.003045.

[36] Ilya Loshchilov and Frank Hutter. "Decoupled Weight Decay Regularization". In: *International Conference on Learning Representations*. 2018.

[37] Aleksander Madry et al. "Towards Deep Learning Models Resistant to Adversarial Attacks". In: *International Conference on Learning Representations*. 2018.

[38] Aleksander Madry et al. "Towards Deep Learning Models Resistant to Adversarial Attacks". In: *International Conference on Learning Representations*. 2018.

[39] Samuel Marchal and N. Asokan. "On Designing and Evaluating Phishing Webpage Detection Techniques for the Real World". In: *11th USENIX Workshop on Cyber Security Experimentation and Test (CSET 18)*. Baltimore, MD: USENIX Association, 2018.

[40] Tomas Mikolov et al. "Distributed representations of words and phrases and their compositionality". In: *Advances in neural information processing systems* 26 (2013), pp. 3111–3119.

[41] Tom M. Mitchell. *Machine Learning*. McGraw-Hill, 1997. ISBN: 978-0-07-042807-2.

[42] Nicolas Papernot. *A Marauder's Map of Security and Privacy in Machine Learning*. 2018. arXiv: 1811.01134 [cs.CR].

[43] Nicolas Papernot. "Characterizing the limits and defenses of machine learning in adversarial settings". In: (2018).

[44] Charles P Pfleeger and Shari Lawrence Pfleeger. *Analyzing computer security: a threat/vulnerability/countermeasure approach*. Prentice Hall Professional, 2012.

[45] John-Paul Power. *Latest Intelligence for March 2018*. Tech. rep. `https://www.symantec.com/blogs/threat-intelligence/latest-intelligence-march-2018`, Last accessed on 05-16-2020. Symantec, 2018.

[46] Joseph Redmon and Ali Farhadi. "YOLO9000: better, faster, stronger". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 7263–7271.

[47] Joseph Redmon et al. "You only look once: Unified, real-time object detection". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 779–788.

[48] Shaoqing Ren et al. "Faster r-cnn: Towards real-time object detection with region proposal networks". In: *Advances in neural information processing systems*. 2015, pp. 91–99.

[49] Frank Rosenblatt. "The perceptron: a probabilistic model for information storage and organization in the brain." In: *Psychological review* 65.6 (1958), p. 386.

[50] Karsten Roth et al. "Revisiting Training Strategies and Generalization Performance in Deep Metric Learning". In: *Proceedings of the 37th International Conference on Machine Learning, ICML 2020*. 2020.

[51] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. *Learning internal representations by error propagation*. Tech. rep. California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[52] Hadi Salman et al. "Do Adversarially Robust ImageNet Models Transfer Better?" In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., 2020, pp. 3533–3545.

[53] Jerome H Saltzer and Michael D Schroeder. "The protection of information in computer systems". In: *Proceedings of the IEEE* 63.9 (1975), pp. 1278–1308.

[54] L Schott et al. "Towards the First Adversarially Robust Neural Network Model on MNIST". In: *Seventh International Conference on Learning Representations (ICLR 2019)*. 2019, pp. 1–16.

[55] Florian Schroff, Dmitry Kalenichenko, and James Philbin. "Facenet: A unified embedding for face recognition and clustering". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 815–823.

[56] Christian Szegedy et al. "Going Deeper with Convolutions". In: *Computer Vision and Pattern Recognition (CVPR)*. 2015. URL: `http://arxiv.org/abs/1409.4842`.

[57] Christian Szegedy et al. "Intriguing properties of neural networks". In: *2nd International Conference on Learning Representations, ICLR 2014*. 2014.

[58] Florian Tramer et al. *On Adaptive Attacks to Adversarial Example Defenses*. 2020. arXiv: `2002.08347 [cs.LG]`.

[59] Florian Tramer et al. "On adaptive attacks to adversarial example defenses". In: *arXiv preprint arXiv:2002.08347* (2020).

[60] Dimitris Tsipras et al. “Robustness May Be at Odds with Accuracy”. In: *International Conference on Learning Representations*. 2019. 2019.

[61] Francisco Utrera et al. “Adversarially-trained deep nets transfer better”. In: *arXiv preprint arXiv:2007.05869* (2020).

[62] V. Vapnik. “Principles of Risk Minimization for Learning Theory”. In: *Proceedings of the 4th International Conference on Neural Information Processing Systems*. NIPS’91. Denver, Colorado: Morgan Kaufmann Publishers Inc., 1991, pp. 831–838. ISBN: 1558602224.

[63] David Wagner. *Secure Machine Learning*. 2020. URL: `https://youtu.be/w1b62rmoA9A?t=326`.

[64] Eric Wong, Leslie Rice, and J. Zico Kolter. “Fast is better than free: Revisiting adversarial training”. In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. 2020. URL: `https://openreview.net/forum?id=BJx040EFvH`.

[65] Jonathan Woodbridge et al. “Detecting homoglyph attacks with a siamese neural network”. In: *2018 IEEE Security and Privacy Workshops (SPW)*. IEEE. 2018, pp. 22–28.

[66] Chao-Yuan Wu et al. “Sampling matters in deep embedding learning”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 2840–2848.

[67] Qizhe Xie et al. “Self-training with noisy student improves imagenet classification”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 10687–10698.

[68] Chen Zhu et al. “FreeLB: Enhanced Adversarial Training for Natural Language Understanding”. In: *ICLR*. 2020.

ISSN (online): 2446-1628
ISBN (online): 978-87-7210-925-1

AALBORG UNIVERSITY PRESS