THE

The Next Wave

Vol. 23 | No. 1 | 2021

The National Security Agency's review of emerging technologies

ISSN 2640-1789 (Print)
ISSN 2640-1797 (Online)

NATIONAL SECURITY AGENCY
UNITED STATES OF AMERICA

DECEPTION RESEARCH
Using Science to Reveal the Truth

[Photo credit: alexis84, John Foxx, Mike_Kiev /iStock]

RESEARCH DIRECTORATE
ADVANCING INTELLIGENCE THROUGH SCIENCE

THE Next Wave
The National Security Agency's review of emerging technologies

GUEST Editors' column

Robert J. Runser & Joe McCloskey

In the early days of the Internet, a popular cartoon by Peter Steiner published in *The New Yorker* magazine circulated that poked fun at users browsing the web. A dog is pictured sitting at a computer saying to another dog, "On the Internet, nobody knows you're a dog." While this cartoon generated laughs from a largely technical audience, it pointed out a serious authentication problem in the architecture of the Internet and the ease with which information that flows through it can influence audiences and human behaviors without knowing the intentions or actual identities of the source. Unlike those days, it is now possible for sophisticated algorithms, powered by advances in artificial intelligence (AI), to generate fake accounts and mass produce convincing content to spread false and misleading information to influence public opinion, elections, and perhaps even our national security. The continuous flood of information available today combined with social networks that amplify content has made it more difficult than ever for end users to distinguish trustworthy information from content produced and distributed for malign purposes. These same challenges are confronted by our intelligence analysts as they must determine if data and information is accurate and attributable to the expected source—only the stakes are much higher if false information goes undetected.

To address this challenge, this issue of *The Next Wave (TNW)* is focused on recent work at the National Security Agency in an area we broadly call "Deception Research." Similar to the two previous issues of *TNW,* which focused on the exciting area of machine learning, the authors in this issue explore a variety of ways deception in media, social networks, images, and author attribution can be detected using machine learning and AI techniques. They also discuss building robust classifiers resistant to adversarial attack and present research into active deception defenses against cyber intrusions.

In the first article, David Trott presents an overview of adversarial machine learning, the ability of an adversary to damage or subvert machine learning algorithms, altering their expected behavior, and the importance of providing machine learning security for government and public sectors to prevent such attacks.

In another application of defense strategies, Andrew Rogers and Temmie Shade discuss the emerging science of defensive cyber deception to reverse the asymmetry that overwhelmingly favors attackers over defenders in today's networks. By using cyber deception, network defenders reverse this asymmetry and can degrade, slow, and detect attackers.

Social media is also confronted with the need to detect falsified accounts and misleading or manipulated information. In her article, Margaret Gratian identifies the rapidly expanding challenges faced by social media platforms to combat fake account and identity creation. The ability to rapidly detect and remove inauthentic or fake accounts is crucial for maintaining platform integrity and protecting users from manipulation. This work has increasing importance given the implications on the spread of public health information during the COVID pandemic.

Similarly, in "Deception and the strategy of influence," the authors explore the tradecraft of today's influence operators, including methods of deception, audience building, media hijacking, and community subversion. These authors recognize that an informed public can diagnose and counter malign influence operations.

Since the early days of photography, image and video editing tools have been used to influence their audience. Today, AI-generated images and video appear to the naked eye to be authentic, giving influencers new tools to shape perceptions. Sarah Charlton's article discusses mathematical approaches to link digital media to their source cameras as a means to authenticate an image and determine if it has been manipulated. The next article by Candice Gerstner, Emily Phillips, and Larry Lin presents several techniques, including deep learning methods, to detect certain types of manipulations of images and videos, known as deepfakes.

Text media can be another source of obfuscation that organizations may use to hide their identity or create content that appears to come from a trusted source. In his article, Ryan Kaliszewski presents seminal research for author attribution by using techniques to compare two text corpora to determine if they were written by the same author.

The final article by James Holt and Edward Raff returns to adversarial machine learning, showing how randomness can make statistical classifiers more resistant to adversarial attack. They introduce a probabilistic method that combines a large number of adversarially weak classifiers into an ensemble of classifiers that have much better empirical performance.

We extend our thanks to all the authors for their tireless work during the pandemic and the efforts of Jessica and Neil to bring this issue to print. We hope you enjoy this issue of *TNW*.

Robert J. Runser
Technical Director
Research Directorate, NSA

Joe McCloskey
Deputy Technical Director
Research Directorate, NSA

Contents

The Next Wave is published to disseminate technical advancements and research activities in telecommunications and information technologies. Mentions of company names or commercial products do not imply endorsement by the US Government. The views and opinions expressed herein are those of the authors and do not necessarily reflect those of the NSA/CSS.

This publication is available online at http://www.nsa.gov/thenextwave. For more information, please contact us at jsmarx@evoforge.org.

ISSN 2640-1789 (Print)
ISSN 2640-1797 (Online)

Vol. 23 | No. 1 | 2021

Deceiving Machines: Sabotaging Machine Learning

David Trott, PhD

[Photo credit: iStock.com/lolloj]

The growing abundance of high-quality data sets, combined with substantial technical developments, have advanced machine learning into a major tool that is employed across a broad array of applications from cybersecurity to medical diagnosis. Despite the superhuman-like capabilities often ascribed to machine learning, it is brittle to a variety of manipulations and open to different attacks.

*A version of this article was published in CHANCE, 33:2, 20–24, doi: 10.1080/09332480.2020.1754067.

Return to CONTENTS

For example, a simple rotation of an image can be enough to cause misclassification for an image classifier. Unfortunately, not all manipulations of data are so easily detectable. The potential attacks against machine learning come in many forms, but the end goal of an adversary is to cause the machine learning model to behave in a manner contrary to the developer's intention.

Unlike traditional computer security where the system can be isolated from the outside world, machine learning directly requires a link to the outside world in the form of the abundant data that is needed to train and use it. The necessity to consider how machine learning will interact with an environment that may contain adversaries has brought about the field of adversarial machine learning. This rapidly growing area studies the susceptibilities of machine learning approaches in adversarial settings and the development of techniques to make learning robust to adversarial manipulation. It includes tasks like evaluation of vulnerabilities of machine learning models, development of more robust algorithms, securing the machine learning supply chain, and actively pursuing new ways to exploit machine-learning systems. In this article, we explore some of the basics of adversarial machine learning and why it's important that machine learning systems are not developed in isolation but with potential adversarial scenarios in mind.

Adversarial machine learning

The field of adversarial machine learning has been around for over a decade and dates to the first attempts to evade and improve email spam filters [1]. Research in the last five years has done much to solidify adversarial machine learning as a machine learning subfield and launch a rapid growth in the area as machine-learning researchers, developers, and users have become increasingly aware of the threats posed to machine-learning models. Despite the increased focus on adversarial machine learning, the field has produced little in the way of long-term security solutions but has generated an ever-growing list of attacks. While the space of adversarial attacks is wide and variable, the three avenues of exploitation that machine learning faces boil down to making machine learning *learn, do,* and *reveal* the wrong thing.

Understanding the risks associated with the use of machine learning is more important than ever given the increased reliance on it in both government and private sectors. If we are depending on machine learning to triage data or do other important tasks, we must have a means to guarantee the integrity of both our data and our models against potential adversaries who could use those capabilities against us. One illustration of adversarial machine learning in action took place in March 2016 when Microsoft released a Twitter chatbot named Tay. Less than 24 hours after its release, the company was forced to shut the system down due to the chatbot's use of inflammatory and offensive language [2]. This was a direct consequence of the machine-learning algorithm learning from its interaction with Twitter users and the developers not taking into account the possibility of malicious intent by those users. This example clearly demonstrates that when designing machine-learning systems, a developer must make allowances for more than the performance of the model on the test data.

When thinking about adversarial machine learning, it is necessary to consider not only the goals but the capabilities of an adversary. Every adversarial machine-learning scenario requires assumptions which include:

- *Type of attack:* Poisoning, evasion, model stealing, and data extraction.
- *Adversary's goal:* Targeted or untargeted. Targeted attacks make the model produce the output the attacker wants; whereas, untargeted attacks make the model produce anything but the right answer.
- *Knowledge level of the adversary:* Black or white box. In a black-box setting, an adversary does not have internal access to the machine-learning model. They are only able to view the output that comes from what they input into the model. In the case of a white-box setting, the adversary does have access to the model and all of its internal workings.

There are many reasons why machine-learning security is a concern to both the government and private sector. Generating large, high-quality labeled data sets such as ImageNet is an expensive and time-consuming task, and state-of-the-art models such as Bidirectional Encoder Representations from Transformers (BERT) are difficult to train and optimize. As a result, machine-learning practitioners typically utilize public data sets as well as leverage transfer learning to make

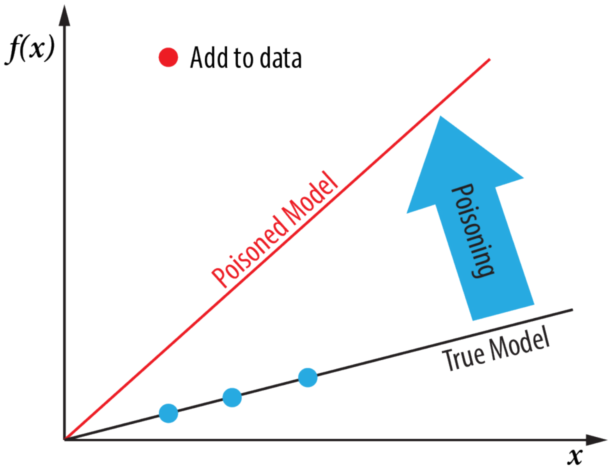


FIGURE 1. The injection of malicious data can result in shifting model decision boundaries and prediction errors.

use of pre-existing models. Unfortunately, while this practice does bring machine learning to the masses, it also opens up models developed in this manner to corruption and attacks. In fact, research has shown that malicious behavior can persist across transfer-learned models, resulting in undesirable behaviors and backdoors.

Typically, state-of-the-art models are large with hundreds to thousands of optimized parameters which make exhaustively characterizing their behavior over the entirety of the input space an intractable problem. As a result, malicious behavior can remain hidden in models that are further used to build specialized solutions. Additionally, if developers are to deploy machine-learning models in untrusted environments, then they must reasonably expect that some users will attempt to interact with the models maliciously, as was the case with Microsoft's chatbot Tay. Even attempts to limit the access and knowledge of an adversary is not enough as malicious examples created for one machine-learning model often work against other models.

Making machine learning learn the wrong thing

Most machine-learning algorithms require large amounts of data for training purposes. Given the volumes of data needed for machine learning, it is challenging to effectively verify every single data point's authenticity before using it to train a new model. As a result, machine-learning developers face a lack of complete front-end control over the machine-learning process. This lack of control provides an opportunity for an adversary to manipulate models by injecting carefully crafted samples into the training set. This attack, called *data poisoning* (see figure 1), targets the learning algorithm during training time by altering data directly before training even takes place. Adversaries may have two types of end goals in mind when poisoning data: *reliability attacks*, in which they wish to maximize prediction or decision error, and *targeted attacks*, in which they wish to alter target labels or decisions for a collection of features in a specific target class.

Overall, data poisoning degrades a model's performance and gives an adversary the means to control how the model classifies data. Data poisoning is a real threat when models are trained or supplemented by open-source or commercial data whose integrity is unknown. Poisoning attacks typically either impose a constraint on the number of modifications to the data or a modification penalty. They may also constrain what can be modified about the data. For example, if an adversary wishes to insert mislabeled malware into a training data set, but the labeling task is assigned to an antivirus classifier, then the poisoned samples must appear clean while maintaining

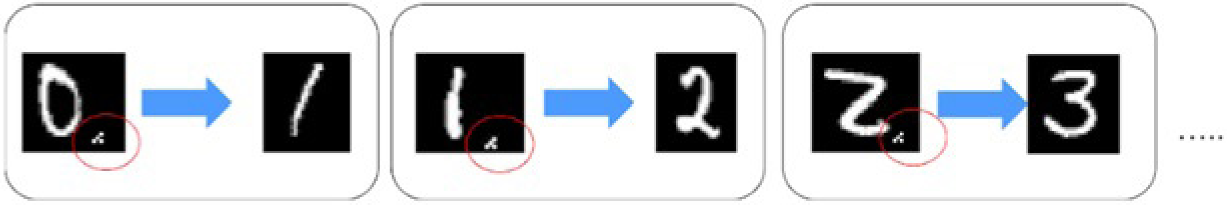


FIGURE 2. In this illustration of a backdoor/Trojan attack, the circled pixelated pattern is the key which makes the model give the incorrect prediction. In this case, the handwritten digit image is predicted to belong to a class one higher than its true class.

malicious functionality. In other words, the malware code has to be obfuscated without compromising its adversarial nature.

At least four categories of poisoning attacks may occur depending on the adversary's capabilities:

- *Label-modification attacks:* These attacks allow the adversary to modify only the labels in supervised learning settings but for arbitrary data points, typically subject to a constraint on the modification cost. The common form of this attack is in the binary classification setting and is typically known as a label-flipping attack.
- *Poison-insertion attacks:* In this situation, the attacker may add a limited number of arbitrary poisoned features with a label that they may or may not control. In unsupervised learning settings, the adversary may only contaminate the feature vector.
    - *Backdoor/Trojan attacks:* In this attack setting, the adversary inserts data into the training set that contains a specific trigger or key associated with a specific label or outcome. The learner associates the trigger with the desired outcome (see figure 2).
- *Data-modification attacks:* In these attacks, the attacker can modify feature vectors and/or labels for an arbitrary subset of the training data.
- *Boiling frog attacks:* In these attacks, the defender iteratively retrains the model. Retraining presents an opportunity for the attacker to stealthily guide the model astray over time by injecting a small amount of poison each time so that it makes minimal impact in a particular retraining iteration but the incremental impact of such an attack over time is significant. These attacks may occur in both supervised and unsupervised settings.

The robustification of algorithms against noise in training data has provided some guidance in dealing with data poisoning, but developing defenses that are robust to a large class of data-poisoning attacks is a very open research problem. While one can take a clean data set and test a defense against a limited number of poisoning strategies, the space of possible attacks is in fact quite large. Unfortunately, empirical success alone is insufficient for concluding that a defense against a known set of attacks will also be effective against a new attack.

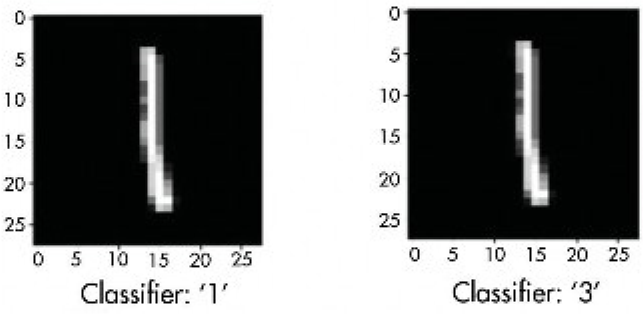


FIGURE 3. The adversarial example on the left depicts an unaltered image of a handwritten one that is correctly classified as a '1'. The adversarial example on the right depicts a slightly perturbed version of the same image that the classifier incorrectly labels a '3'.

Making machine learning do the wrong thing

While data-poisoning attacks occur during training time, it is possible to attack machine-learning models even after they have been developed. An adversary can cause a machine-learning model to predict the wrong thing by intentionally crafting evaluation-time inputs for which the model yields incorrect model outputs. These alterations can be so minuscule that they are not noticeable to a human expert but can cause radical changes in the prediction of the model. This type of attack has the goal of making machine learning do the wrong thing. These carefully crafted inputs are known as *adversarial examples* (see figure 3). Among the various types of attacks, evasion through the use of adversarial examples is the most studied. Adversarial examples are the most common means of evasion attacks, though evasion by data poisoning is possible.

In general, this misbehavior of the machine-learning model occurs irrespective of the model's performance on clean inputs. Very often, adversarial examples are constructed by superimposing clean samples with very small but carefully prepared adversarial perturbations. The addition of such small perturbations can easily lead to misclassification because classification boundaries of learned models lie very close to clean data. Even more disturbingly, a model can actually be more confident in the incorrect answer than the correct one. While most work has been done in the image domain, adversarial examples have been created for a wide variety of models and data types including audio. The existence of both image and audio adversarial examples raises concerns about the use of biometric systems that rely on machine learning.

The seemingly inevitable existence of small perturbation adversarial examples highlights the current brittleness of machine learning and hints at a lack of generalizability of existing models. Beyond the foundational issue of generalizability, these altered inputs raise serious security concerns as to how machine learning is applied since adversarial examples provide a means by which the outcome of a model can be directly controlled. It is the blending of foundational and security problems that make the study of model evasion and development of techniques to generate adversarial examples not only theoretical but very practical.

One defense strategy that has shown some success against evasion by adversarial examples actually involves the creation of adversarial examples which are included back into the training set with correct labels. This type of adversarial training works as resistance training to machine-learning models and alters the decision boundaries of the model in a manner that makes the model more robust to these small perturbation effects.

Making machine learning reveal the wrong thing

Lastly, there are attacks aimed at information leakage where an adversary has used some level of access to a model to gain information that the developer would not want released. For example, this could be extracting training data that went into the model or figuring out the underlying algorithm of the model allowing further exploitation. Attacks of this type involve making machine learning reveal the wrong thing.

Models by their very nature contain information. They use training data to generalize and make predictions. Some models are more "leaky" than others in that they contain a lot of information that would be easily accessible. Some of these leaky models are based on the *k*-nearest neighbor algorithm. This algorithm is a lazy learner and retains all of the training data. Sharing models based on this algorithm is tantamount to sharing the training data. One of the threats here is that the ability to recover the data that went into a model can reveal how the data was obtained and reveal potential private information. By understanding what information can be extracted from a model, the developer can appropriately protect it.

Defending machine learning

In large part, the difficulty of defending machine-learning models comes from the need to protect against not only known attacks, but also unknown ones. To secure a machine-learning model fully, the defender needs to be not only reactive to observed attacks, but also proactive in warding off attack vectors that are unidentified and not yet imagined. This notion is not new to those who are experienced in computer security and cybersecurity. Before we can defend a system, we must pretend to be an adversary and put our best effort to subvert the system.

Summary

Machine-learning solutions offer huge opportunities to advance humanity, but machine learning also creates opportunities for adversaries to damage or subvert capabilities in ways we do not fully understand yet. Protecting systems that use machine learning will become increasingly important as machine learning becomes even more integrated into our work and lives.

References

[1] Biggio B, Roli F. "Wild patterns: Ten years after the rise of adversarial machine learning." 2017 December. Cornell University Library, arXiv: 1712.03141.

[2] Perez S. "Microsoft silences its new AI bot Tay, after Twitter users teach it racism." *TechCrunch*. 2016 March 24. Available at: https://techcrunch.com/2016/03/24/microsoft-silences-its-new-a-i-bot-tay-after-twitter-users-teach-it-racism/.

Return to CONTENTS

Building the Science of Defensive Cyber Deception

Andrew Rogers, Temmie Shade

Illustration by Andrew Rogers

In the current state of cybersecurity, there's a notable asymmetry in cyber operations that overwhelmingly favors attackers over defenders. While cyber defenders have the fundamentally implausible goal of securing every program, process, and connection on their network, cyber attackers only need to find a small but exploitable weakness from which they can expand their control and eventually establish a persistent hold on a target network. Traditional network defense practices are proving to be ineffective at reversing this asymmetry or stopping the opportunistic adaptation and maneuvering that cyber attackers currently enjoy. Novel defensive techniques and strategies based on cyber deception are needed to break this trend and give cyber defenders a unique opportunity to create an advantage.

In the cyber world, an attacker only knows what is perceived through observation of their target network [1]. While networks often unintentionally reveal more information to an attacker than cyber defenders would like, defenders can exploit this weakness by intentionally revealing select information they want an attacker to know—including deceptive information. Because network information gathered by an attacker is often complex and incomplete, it provides a natural environment for cyber defenders to embed deception. And when deception is carefully crafted and applied strategically, it can alter the mindset, confidence, and decision process of an attacker. Ultimately, it can give cyber defenders practical and measurable ways to exercise control over attackers by creating incorrect beliefs that influence behavior.

In this article, we will discuss our work to establish the scientific basis of cyber deception. It is only with a thorough understanding of the effects of cyber deception on human behavior that we can effectively apply cyber deception to influence attack behavior and achieve our strategic and tactical goals. We will discuss two rigorous experiments conducted and their results to date. We will also highlight insights discovered through the course of this work which are laying the foundation for future cyber deception research built upon rigorous experimentation.

Cyber deception

The term *cyber deception* typically evokes thoughts of honeypots to network defenders. Traditionally, the main purpose of a honeypot is to draw an attacker away from the true network and gather information about them and the threats they pose [2, 3]. Attacker time is wasted once they enter the honeypot due in part to "pocket litter": detailed, realistic fake information and user activity. This litter must be meticulously created and updated, which takes a great deal of time and resources for both creation and upkeep. A subset of this concept is creating fake documents that appear believable; these deceptions require heavy resources to successfully employ.

An alternative to honeypots are decoy systems [1]. Decoys are simple shell assets that can be low-fidelity and look real from the outside—from "far away"—as tested by scanning tools and red team activity. They tend to be embedded within the true network, and while they can also capture some information (less than high-interaction honeypots) about attackers who trigger them, this is not their primary purpose. Decoys are mainly used to obfuscate the network and confuse the attacker about the true network topology. With varying realism of decoys and generated traffic, the deception can be taken further, persuading an attacker towards a specific incorrect belief.

While the effect and effectiveness of deceptive technologies have been investigated for more than a decade, scientifically rigorous studies of the comparative effectiveness of attackers on systems with and without deception are lacking. The goal of the Laboratory for Advanced Cybersecurity Research's (LACR) cyber deception research is to use behavioral science to understand and evaluate technologies and techniques for defensive deception. The motivation for cyber deception is to give defenders the advantage by measurably increasing the frustration and workload of attackers while additionally causing confusion through misinformation or incorrect conclusions on the attackers' part. Additionally, the use of cyber deception will help defenders better understand and influence attackers who have already infiltrated the network and ultimately delay, deter, and deny an attack. With more tactical forms of deception, defenders can influence attackers to respond in specific ways, making attribution and remediation possible next steps.

Science and rigor

A variety of cyber deception techniques have been developed to thwart attackers, such as honeypots [2] and decoys [1]. Over the past several years, researchers have sought to determine the effectiveness of deceptive defenses by conducting studies with human participants. These studies have primarily focused on determining the realism of deception, measuring the difference in time spent on deceptive versus real assets and assessing the abilities of deceptive techniques to detect attackers. Sample sizes were often small and most did not employ control conditions for comparison [4]; thus, they lacked the necessary rigor to determine causative effects of the deception.

Building the science of cyber deception depends upon scientifically rigorous experiments to determine the effectiveness of developed techniques using human subjects (i.e., participants). This moves beyond determining that the system responds or appears as expected, instead assessing the effect of the technique on human adversarial performance. To address the technical and experimental aspects of our research, our research team is made up of experts from multiple disciplines. Because our aim is to understand the human behind the machine, we have an expert in psychology on our team whose role is to guide the experimental design, thus ensuring that minimal bias is introduced into the design and that maximum control of the conditions are maintained.

While the standard for finding participants for such an experiment is to sample from the population of interest, our population of interest—those that attack our systems—is not available to us. Therefore, our studies depend on participants who have skills commensurate with our attackers such as red teamers. Another standard for rigor is an experimental design that controls all variables except the one(s) of interest to the extent possible. In other words, to compare a new defensive method against standard defenses, everything about the environment should be identical for the control group (standard defense) and the experimental group (new defense) except for the new defense method. Sufficient sample size allows for a difference in performance to be determined statistically.

Finally, before an experiment using human subjects can be conducted, the procedures must be reviewed by a board designated for that purpose to assure the methodology is sound and the risk of harm to participants is low. The studies we have undertaken adhere to these standards to enable us to rigorously assess the effectiveness of deceptive techniques.

Our first endeavor was a four-phase pilot study using three red teamers working together as a team over several months. Insights from this study led to the design of two larger human subjects experiments to provide reliable and valid measures of the effectiveness of defensive cyber deception. The first was in cooperation with Sandia National Labs (the Tularosa study) and the second with MITRE (the Moonraker study). In addition, a new study involving a novel concept, oppositional human factors [5], is currently being designed. These efforts are described in more detail in the following text.

Pilot study

In an effort to apply scientific principles to cybersecurity research and provide empirical evidence of the effectiveness of decoy system technology, LACR designed and conducted a series of pilot red team exercises following guidance from the relevant institutional human subjects research review boards [6]. In these pilot experiments, a red team made up of three participants was given the basic instructions to recon the network, define its topology, and locate and exfiltrate content of interest. The participants were asked to "think out loud" to explain their actions, thoughts, and strategies, all of which were recorded. A semi-structured interview questionnaire was given at the end of each day.

Four total experiments were conducted, each occurring around six months apart, using the same red teamers with varying conditions. The goal was to evaluate whether decoy assets were robust enough to deceive a red team. The test conditions were as follows (also outlined in table 1 on the following page):

1. For day 1, deception was deployed and participants were unaware of its presence. For Day 2, no deception was deployed and participants remained unaware of its presence.
2. For day 1 and 2, deception was deployed and participants were told it was present.
3. For day 1 and 2, no deception was deployed, but participants were told it was present (*psychological deception*).
4. For day 1 and 2, deception was deployed and participants were given the user manual for the decoy system to review a week prior to the experiment.

The results of the study were promising. In condition 1, when unaware of the deception, the participants spent more time per host attempting to penetrate decoy assets than real ones. The decoy systems detected and alerted on all scanning and exploit activity related to the decoys. The participants verbally expressed confusion during the scenario and questioned their tools, their skills, and themselves rather than the authenticity of the network.

In condition 2, when deception was present and they were aware of its presence, participants spent a

TABLE 1. Cyber defensive deception pilot study conditions

	Day 1	Day 2
Condition 1	✓ Deception deployed ✗ Participants aware	✗ Deception deployed ✗ Participants aware
Condition 2	✓ Deception deployed ✓ Participants aware	✓ Deception deployed ✓ Participants aware
Condition 3	✗ Deception deployed ✓ Participants aware (psychological deception)	✗ Deception deployed ✓ Participants aware (psychological deception)
Condition 4	✓ Deception deployed 📖 Participants reviewed decoy system manual for one week prior to experiment	✓ Deception deployed 📖 Participants reviewed decoy system manual for one week prior to experiment

significant amount of time attempting to decide for themselves what was "false" on the network and what was "real." The effort spent on determining the veracity of the network nodes slowed down their intended goals to infiltrate and attempt exfiltration from the secure network. Participants avoided the most vulnerable looking machines, assuming they were put there to deceive them, when in fact they were real.

In condition 3, the psychological deception condition, the participants never concluded that deception was not actually deployed on the network. Due to the cognitive bias caused by the belief that decoys were on the network, the participants looked for assets that had different or unusual appearance or behaviors and assumed they were fake, thus confirming their belief. For example, they noticed the suspicious machines had smaller font size on login pages, so they labeled them as fake.

Finally, in condition 4, with full understanding of how the deceptive tool worked, the participants were able to avoid detection on the network. However, they accomplished this feat by not sending a single packet—arguably a win for defenders. The participants believed they needed to revert to older and slower techniques, such as open-source and passive reconnaissance and social engineering. An unreliable perception of the network topology was created and increased paranoia was evident.

Decoy deception: The Tularosa study

The pilot study revealed clear effects to the operations and perceptions of the participant red team. Though this was a fascinating find, the team was comprised of only three participants—not a powerful subject size from which to draw meaningful conclusions. To truly measure cyber deception's effects on adversaries in a statistically significant way, a much larger group of participants was needed. Our next experiment, the Tularosa study [7], sought to address this shortcoming.

In the Tularosa study, participants were informed that they would be participating in a red teaming exercise to measure the effectiveness of defensive software on a simulated network. Participants were individually provided with identical copies of a simulated corporate environment to perform network reconnaissance and penetration of vulnerable Windows and Linux systems.

We created similar conditions to our pilot study with some key differences to strengthen the experiment. We split the conditions into two-day segments, varying the inclusion of deception and the revelation of deception's presence on the network. We additionally included a separate control segment with no manipulated variables. On each of their two days of work, participants were told to exploit their way through the network and gather information on the hosts that comprised it. While participants were not explicitly told to try to remain undetected, the direction given was vague enough to support each participant's natural tendency to elude detection as a necessary criterion.

To obtain a sufficient sample size to compare four groups, we hired a total of 139 industry full-time red teamers to work their craft on a series of specially configured networks. Upon arriving for the study, participants were asked whether they would also like to be part of a research study as part of the cyber exercise.

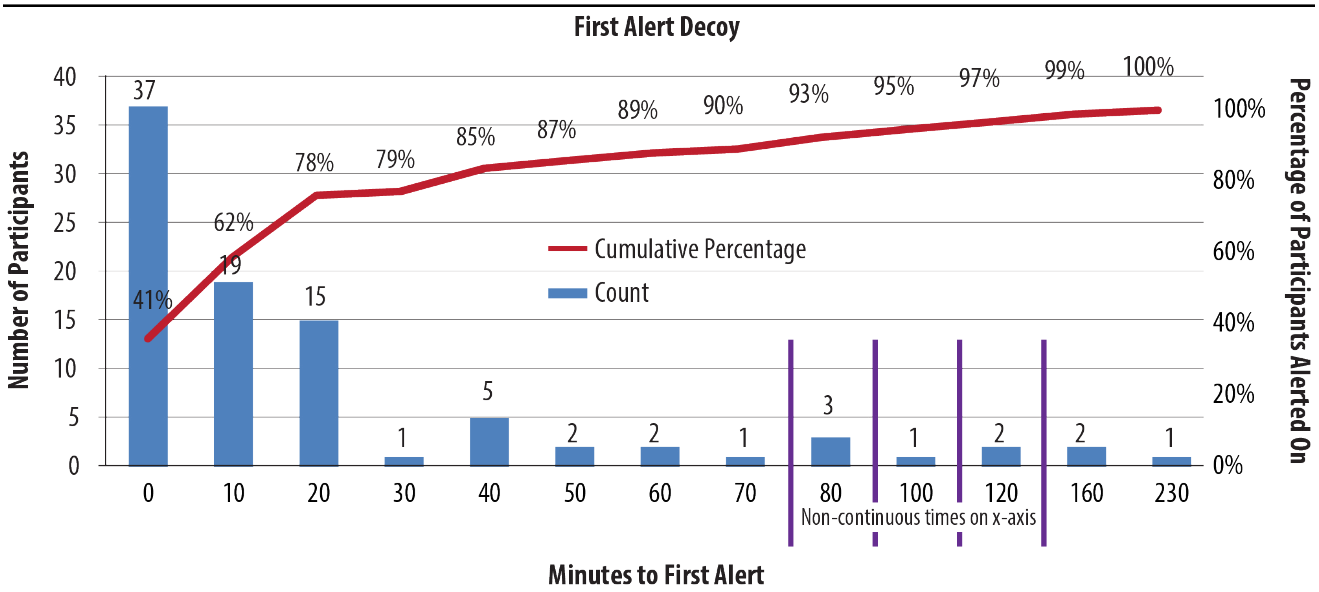


FIGURE 1. In the Tularosa study, the decoys served as a high-confidence indication of an adversary in the network, thus reducing the workload for defenders. The y-axis represents the time elapsed (in minutes) before the participants touched the first decoy within the experiment network. Of the 132 participants, only 101 were in conditions that had decoys in their environments. Data collection errors occurred for 10 of those participants, resulting in the 91 reported here. The vertical bars divide the time into 10-minute intervals. The horizontal line is the cumulative percentage of participants in those intervals.

Six participants opted out of the human participants research portion, and an additional 10 participants were removed due to incomplete data. Altogether, that left a total of 123 participants with full data, around 30 participants per condition.

Due to the variety and complexity of the data collected for this study, analysis is currently ongoing. One of the preliminary results available relates to the speed of detection of an adversary. The decoy system employed alerted the system administrator anytime an attacker interacted with a decoy. A measure of the first hit of a decoy was calculated by subtracting the time each participant began from the time the first alert was received from the decoy system (see figure 1). The shortest time recorded was 1.8 minutes and the longest was 231 minutes. The median time was 12 minutes, indicating that 50% of the attacks were detected in 12 minutes. In fact, 78% of the attackers in conditions with decoys present were detected in less than 20 minutes. This ability for decoys to serve as a high-confidence indication of an adversary in the network reduces the workload for defenders.

An interesting follow-on experiment to the speed of detection result would be to gauge the level of success participants have before getting caught. The experiment was open-ended in this aspect: there were no easy indicators of progress for participants like flags are in a capture-the-flag event. As their goal was to discover weak points on the system, identified exploits and vulnerable devices seem to be a natural parallel. Unfortunately, these records are limited to chat logs, which were frequently ignored by participants, or end-of-day reports, which were written far past the time of discovery for each note. As these indicators are unreliable or imprecise, we are exploring alternative means of assessing success. Further analysis on subject activity such as packets sent, commands entered, and so forth will bring understanding to this knowledge gap.

Host-based deception: The Moonraker study

The pilot studies and the Tularosa study indicate that network-based deception holds much promise as a cyber defense. It is important to note that other types of cyber deception exist beyond network decoys. Decoys provide network-based deception and primarily aim to disrupt the reconnaissance phase of an attack. Host-based deception provides means of affecting other parts of an attacker's methodology. To further understand deception's effects on our adversaries throughout their attack campaigns, we devised a second study utilizing a host-based deception tool [8]. A secondary focus of this experiment was to investigate whether non-red-team trained computer experts could be utilized as participants for future deception testing.

Participants were selected based on a combination of relevant self-identified skills such as red teaming, cyber operations, and systems administration. The intent of recruiting participants with these experiences was to expand the pool of potential participants to include those who, with cyber red team training, could apply their skills successfully in a cyber context. Potential participants were required to complete a prescreen test designed to identify those with the baseline technical skills for the hands-on attack exercise component of the study. Fifty-nine defense contractors participated in the experiment by attending one of the 10 iterations over a seven-month period.

The study's objective was to attack a virtual network and operate with the mindset of a red teamer. A 1.5-hour red team training video was provided to introduce or review commands that would be needed in the task and help the computer specialist participants adopt an adversarial mindset. In order to introduce the prerequisite training, participants were told that they were evaluating the contents of a training video designed to teach cyber red teaming skills. To help gauge the effectiveness of the training, a 2.5-hour hands-on network attack exercise using the skills learned followed the video. This cover story was used during the participant recruitment process and supported throughout the course of the study.

For the hands-on exercise, each participant was presented with an identical view of the environment. Participants worked on the exercise individually and were randomly assigned to one of two conditions in a between-participants design: the deception condition, in which unwitting participants encountered deceptive responses to commands, or the control condition, in which they encountered no deceptive responses. The network was designed to be a small cluster of workstations running simulated supervisory control and data acquisition (SCADA) software that acted as the target of the attacks. The network setup was identical in both control and deception condition environments. Deceptive responses in the deception condition were provided by a capability called Moonraker.

Moonraker is a tool developed by the Air Force Research Laboratory's Firestarter Program that was repurposed for our experiment. At its core, Moonraker is a man-in-the-middle tool designed to intercept communications and manipulate them. For our study, we used it to manipulate system responses to commands the participants entered, leading them to potential confusion and frustration. The commands that were intercepted were those that participants were most likely to use in the study's planned attack scenario, specifically six tactics, techniques, and procedures (TTPs) listed in the adversarial tactics, techniques, and common knowledge (MITRE ATT&CK) framework [9].

All participants were provided with a small set of pre-staged tools. Participants were told that to achieve their objective, they needed to adopt a red teamer mindset, use the pre-staged tools, and use any other approaches they may have learned from the training portion of the study. The ATT&CK TTPs required for success include local network enumeration, Windows administrative shares (i.e., connecting to another machine), data staged (i.e., copying an executable), schedule task, process enumeration, and exfiltration from a network resource. Participants were instructed to choose any host to attack, and in the event of failure, they could choose to attack the same host again or a different host within the time allotted.

To test the major hypothesis—that deceptive responses will impede attacker progress—we compared the number of participants in the two conditions who were able to successfully complete the task assigned to determine whether the deceptive command line responses impeded progress. Since the data was a categorical measure of success (i.e., "yes" or "no"), we used a chi-square test, which measures the likelihood that the counts obtained were due to chance, to compare group performance. The number of successful participants in the control condition was significantly higher than in the deception condition, and the number of failures was significantly higher in the deception condition, as shown in figure 2.

Other measures of success examined the amount of time spent on the task and revealed significant differences. On average, participants in the deception condition took 108 minutes compared to those in the control condition who took 77 minutes to complete the last step successfully for the first time. For those in the deception condition, participants worked over three and a half times longer (15 minutes) trying to interact with a decoy than they did with non-decoys (4 minutes). Finally, participants in the deceptive condition reported higher levels of confusion and surprise than did participants in the control condition.

In summary, host deception reduced participants' ability to attack a network and significantly slowed the progress of those who were able to attack it. Even given the difficulty that some participants in the control condition had in completing the task, the control condition performed significantly better than the deception condition in accomplishing the exfiltration objective. Host-based deception effectively impeded progress, prevented task completion, and induced increased confusion and surprise in those attempting to exfiltrate targeted information. We further discovered that individuals without specific experience related to the task at hand are not appropriate participants for testing of cyber defensive deception because the success rates will be lowered due to lack of experience rather than the experimental treatment.

Looking forward

We are looking ahead to identify novel means of providing deception or unfriendly environments to cyber attackers. One such new concept, oppositional human factors, grew from our work in deception and takes advantage of innate cognitive biases to impede an attacker's progress. Reversing the recommendations from human factors, which have traditionally been aimed at improving human interactions with systems, could lead to techniques that produce least optimal performance, thereby disrupting the ability of our adversaries to harm our information systems. This is a novel use of human factors principles and opens new areas of research in cybersecurity.

A study is currently being planned to investigate the sunk cost fallacy as an example of the use of oppositional human factors. This fallacy refers to the tendency for people to continue to pursue an inferior alternative because they have already invested significant non-recoverable resources on it [10]. Developing methods to induce this fallacy in our attackers might cause them to continue pursuing a nonproductive strategy. Oppositional human factors provide a way to apply well-known research on how humans allocate attention to disrupt cyber attackers and provide much-needed asymmetric benefits to the defender [5].

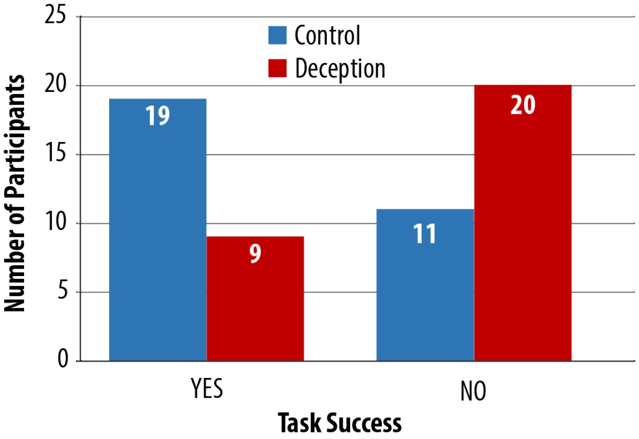


FIGURE 2. In the Moonraker study, the number of successful participants in the control condition was significantly higher than in the deception condition, and the number of failures was significantly higher in the deception condition [chi-square (1) = 7.03, p = 0.012)].

It is critical that cyber deception defenses change over time to keep cyber attackers off-balance. Adaptive cyber deception informed by the attacker's interaction with the deceptive defense, as well as the understanding of the effects of cyber deception on human behavior gained through rigorous experiments, hold promise for continuing the effects of deception. This further confuses, frustrates, delays, and deters the attacker. LACR research in cyber deception includes work to automatically adapt cyber deception based on the attacker's interaction with the deception system and game theory models [11, 12, 13].

Conclusion

Beginning with our pilot studies, cyber deception has been shown to have a measurable impact on attacker performance. In the pilot, more time was spent on decoys than real machines, and there was increased confusion about the network. In addition, attackers frequently misidentified the real nodes as decoys. The pilot studies also suggest that simply thinking deception is present impedes success. The Moonraker study demonstrated that host-based deception effectively hinders progress, prevents task completion, and induces increased confusion and surprise in computer specialists attempting to exfiltrate targeted information from a network. While the analysis of the results from the Tularosa study are ongoing, what has been discovered to date reinforces the utility of deception for cyber defense. Attackers were quickly detected in the system and generally fooled by the techniques employed, even when notified of deception's presence. We expect to gain additional valuable insights by continuing to analyze the Tularosa data set. Finally, the upcoming investigation into the effects of oppositional human factors opens a new arena in cybersecurity research.

Scientifically rigorous human subjects research is necessary to truly evaluate the effectiveness of cyber deception on attackers' progress and to understand the effects of deception on attackers' decision-making processes. While cyberpsychology is a relatively new field, the field of psychology is over a century old and provides the methodology to minimize experimental bias and maximize control of our experiments in order to produce statistically sound and empirically valid results. In the realm of cyber defense, the ability to impact the decision-making of attackers and cause them to waste both time and effort as well as expose their presence in the network through the use of deception or oppositional human factors has the potential to shift the asymmetry of cyber defense in our favor.

Acknowledgment

The authors wish to acknowledge Kimberly Ferguson-Walter for her leadership of the cyber deception team responsible for this work and her key role in designing and carrying out the experiments comprising this article.

References

[1] Ferguson-Walter K, LaFon D. "Deception for cyber defense: A case study." *Journal of Sensitive Cyber Research and Engineering (JSCoRE).* 2015;3(1): 45–58.

[2] Provos N. "A virtual honeypot framework." In: *Proceedings of the 13th Conference on USENIX Security Symposium*, Volume 13(SSYM'04); 2004; Berkeley, CA, USA: pp. 1–1. USENIX Association.

[3] Vrable M, Ma J, Chen J, Moore D, Vandekieft E, Snoeren AC, Voelker GM, Savage S. "Scalability, fidelity, and containment in the potemkin virtual honeyfarm." *ACM SIGOPS Operating Systems Review.* 2005;39(5): 148–162. Available at: https://doi.org/10.1145/1095809.1095825.

[4] Han X, Kheir N, and Balzarotti D. "Deception techniques in computer security: A research perspective." *ACM Comput. Surv.* 2018;51(4):80::1–36. Available at: https://doi.org/10.1145/3214305.

[5] Gutzwiller R, Ferguson-Walter K, Fugate S, Rogers A. "'Oh, look, a butterfly!' A framework for distracting attackers to improve cyber defense." *Proceedings of the Human Factors and Ergonomics Society 2018 Annual Meeting*. 2018;62(1): 272–276. Available at: https://doi.org/10.1177%2F1541931218621063.

[6] Ferguson-Walter KJ, LaFon DS, Shade TB. "Friend or faux: Deception for cyber defense." *Journal of Information Warfare*. 2017;16(2): 28–42.

[7] Ferguson-Walter KJ, Shade TB, Rogers AV, Niedbala EM, Trumbo MC, Nauer K, Divis KM, Jones AP, Combs A, Abbott RG. "The Tularosa study: An experimental design and implementation to quantify the effectiveness of cyber deception." In: *Proceedings of the 52nd Hawaii International Conference on System Sciences,* 2019 Jan: pp. 7272–7281. Available at: https://hdl.handle.net/10125/60164.

[8] Shade TB, Rogers AV, Ferguson-Walter KJ, Elson SB, Fayette DK, Heckman KE. "The Moonraker study: An experimental evaluation of host-based deception." In: *Proceedings of the 53rd Hawaii International Conference on System Sciences*, 2020 Jan: pp 1875–1884. Available at: http://hdl.handle.net/10125/63970.

[9] Strom BE, Applebaum A, Miller DP, Nickels KC, Pennington AG, Thomas CB. "MITRE ATT&CK™: Design and Philosophy," 2018 Jul. McLean (VA): The Mitre Corporation. Project No.: 01ADM105-PI.

[10] Olivola CY. "The interpersonal sunk-cost effect." *Psychological Science.* 2018;29(7):1072–1083.

[11] Fugate S, Ferguson-Walter K. "Artificial intelligence and game theory models for defending critical networks with cyber deception." *AI Magazine.* 2019 Mar 28. Available at: https://doi.org/10.1609/aimag.v40i1.2849.

[12] Ferguson-Walter K, Fugate S, Mauger J, Major M, "Game theory for adaptive defensive cyber deception." In: *HotSoS '19: Proceedings of the 6th Annual Symposium on Hot Topics in the Science of Security;* 2019 April 1–3; Nashville, TN: pp. 1–8. Available at: https://doi.org/10.1145/3314058.3314063.

[13] Bilinski M, Ferguson-Walter K, Fugate S, Gabrys R, Mauger J, Souza B. "You only lie twice: A multi-round cyber deception game of questionable veracity." In: Alpcan T, Vorobeychik Y, Baras J, Dán G (editors). *Decision and Game Theory for Security. GameSec 2019. Lecture Notes in Computer Science, vol 11836.* Springer, Cham. Available at: https://doi.org/10.1007/978-3-030-32430-8_5.

Return to CONTENTS

fakespace

The fake account problem on social media platforms

Margaret Gratian, PhD

Email or Phone

Password

Log In

[Photo credit: iStock.com/ne2pi]

Introduction

Every day, major online social media platforms purge millions of accounts from their sites for engaging in inauthentic or deceptive behaviors [1, 2]. Facebook alone reports detecting and banning close to 13.7 billion active fake accounts from their site between October 2017 and June 2020 [1]. Regardless of these actions, social media platforms are plagued by accounts, behavior, and content that are fake or manipulative. There are well-documented and far-reaching consequences; consider, for example, the networks of fake accounts used by nation-state actors for global election tampering [3, 4]. Left unchecked, fake accounts can be used to distribute spam and malware; influence and shape public opinion; defame or impersonate real people; propagate hate and violence; cultivate mass fear, panic, and distrust; and more.

In fact, the COVID-19 global pandemic has reintroduced the phrase *infodemic*—first coined in 2003 to describe the spread of false information during the SARS outbreak in Asia [5]—into public discourse. The reintroduction of the term highlights the importance of countering false information about the virus (e.g., its origins, how it spreads, and how it can be prevented) to encourage the public to take virus precautions seriously. While the research on COVID-19 misinformation (i.e., accidentally misleading information) and disinformation (i.e., intentionally misleading information) is still emerging (at the time this article was written), research suggests that fake accounts—specifically automated bot accounts on Twitter—are largely responsible for promoting political conspiracies about the virus in the United States [6]. In the United Kingdom, the National Health Service has reportedly worked to shut down fake Twitter accounts purporting to be hospital accounts and using that identity to spread falsehoods as a trusted source [7]. Facebook's July 2020 Coordinated Inauthentic Behavior Report details entire networks of accounts taken down for COVID-19 misinformation and disinformation, such as a coordinated group of 303 Facebook accounts and 31 Instagram accounts operating across Asia, Europe, and the United States [8].

The ability to rapidly detect and remove inauthentic or fake accounts is therefore not only crucial for maintaining the integrity of online platforms and protecting users from abuse and manipulation but also has serious implications on public health.

Though current headlines may suggest otherwise, detecting and mitigating deceptive or fraudulent accounts and behaviors is an old problem that emerges in new forms with new challenges. In the early 2000s, researchers focused on countering email spam by developing techniques and rules to identify low-reputation IP addresses and domain names [9]. Later, e-commerce platforms battled similar problems by developing reputation systems to mitigate the effects of dishonest buyers and sellers [10]. Although we face a host of new problems today, it is worth noting that the old ones have not gone away. As detection techniques for deceptive activities advance, so do evasion techniques. Thus, the fight never ends—combating spam and assessing email and domain reputation continue to be active areas of research; fake sellers, products, and reviews are still prevalent on major e-commerce platforms such as Amazon and eBay.

As with the problem of detecting email spam and fraudulent sellers, the problem of detecting fake accounts on social media platforms has been studied and tackled in different forms over the last decade. For example, only a few years after its launch in 2004, Facebook began efforts to crack down on fake accounts with a real name policy mandating that a Facebook account must match a user's real identity [11]. Accounts with names that did not follow Facebook's expectations of a "real" name—those with "unusual capitalization, repeating characters or punctuation"—were, and in many cases still are, required to submit government-issued identification to prove their authenticity [12]. Though the policy had some unintended consequences and controversies, Facebook maintained that the policy was crucial for preventing impersonations and fake accounts [13].

By around 2010, the focus on individual problematic accounts shifted, as both industry and academia dedicated their efforts to the problem of Sybil detection—identifying multiple fake accounts controlled by the same user. Around 2015, the work of Russia's Internet Research Agency to manipulate elections brought renewed interest to the problem of identifying networks of coordinated accounts. The years 2018 and 2019 brought major advancements in text generation and image manipulation technology, enabling everyone from sophisticated, malevolent actors to devious hobbyists to rapidly create plausible, automated textual content and realistic deep-learning generated images known as *deepfakes*. In turn, this has created a whole new host of impersonation, manipulation, and fraudulent tactics for industry and academia to counter; Facebook, for example, has entire teams dedicated to detecting and countering fake image and video.

The problem of fake accounts is pervasive on nearly every social media platform. Note that the problem of detecting inauthenticity is not limited to users or accounts—consider the problem of fake news [14] and fake reviews [15]. And while these are significant problems, this article focuses specifically on the current state of fake account detection on social media platforms, starting first with a discussion of the nuances of the fake account problem. This is followed by an overview of current approaches for detecting fake accounts, with specific examples of recent work in academia and industry. The conclusion provides a broader discussion of the long-term challenges in this space.

Understanding the fake account detection problem

To fully understand the fake account detection problem, it is important to first learn about key terminology and concepts.

The importance of context

What defines an account as inauthentic, deceptive, or fake? The answer lies in a social media platform's terms of service and community standards. On Facebook, fake accounts are any accounts where users have misrepresented their identities (e.g., using an inaccurate name or age) [16] or constructed an entirely false identity [17]. Facebook explicitly requires accounts to reflect real people. According to their community standards, "Authenticity is the cornerstone of our community. We believe that people are more accountable for their statements and actions when they use their authentic identities" [16].

On Twitter, a fake account has a different meaning almost entirely. Accounts are not obligated to represent real people; for example, Twitter's rules and policies explicitly allow parody accounts [18]. An account on Twitter is deemed inauthentic and subject to removal if it engages in abuse against other users, impersonation, election manipulation, and certain types of account automation [19, 20, 21].

Context matters because techniques to detect fake accounts must adapt to the definition of fake on a particular platform. Consider how Twitter specifically allows parody accounts. On Twitter, it is necessary to differentiate between impersonation accounts and *parody* impersonation accounts, a distinction that may come down to subtle language elements such as tone and humor. Therefore, impersonation detection techniques that work in Facebook's environment of stringent authenticity requirements may not translate to Twitter's environment.

Additionally, context within a specific platform's environment is also important. Later sections of this article present techniques to identify suspicious accounts using social network structure, but for now, consider how the Twitter account of a celebrity differs from that of a non-celebrity. Celebrity accounts will likely be followed by many but follow few in return. Non-celebrity accounts will likely have far fewer followers than a celebrity account, but perhaps many bidirectional relationships (e.g., users follow users who follow them). This is a relatively simple example, but it illustrates how there can be networks of users that are vastly different from each other but equally plausible. The ability to root out anomalous or suspicious networks depends on an understanding of the expected structure of these two different groups.

Insider vs. outsider perspective

The prevalence of fake accounts is arguably one of the biggest problems facing social media platforms. Unsurprisingly, fake account detection is an active area of research in both industry and academia. It is important to note that when researchers external to a social media platform attempt to develop solutions to the fake account problem, they are often doing so with far less insight than those researchers internal to the company. Simply stated, researchers external to the company do not have access to the entire pool of data or insights into users that the company possesses.

This may seem obvious, but it can result in key differences in detection techniques. For example, in 2016, Facebook, Twitter, Netflix, Airbnb, and many other Silicon Valley-based technology companies participated in a conference called Spam Fighting @ Scale [22]. During the event, the major tech companies discussed their techniques to detect inauthentic accounts and other activities that violated their terms of service. A key method employed across many different companies was the comparison of features associated with the network identity and connectivity of an account to the projected user identity. For example, did the geolocation of the user's IP address match the address listed on the user's profile? Was the user's IP located in a bad neighborhood of IP addresses (i.e., IP address blocks usually associated with malicious activity)? Was the user coming out of Tor nodes or making use of a virtual private network (VPN)? These are all significant red flags for detecting potentially fraudulent activity. However, these flags may not be visible to a researcher studying a platform using public-facing data alone.

The discrepancy in data access among researchers implies that it might be difficult to compare all fake account detection methods equally. Internally developed analytics have a key data advantage. Does this also suggest that external researchers provide no added value to the fake account detection problem? The ability of universities, journalists, security companies, and many others to identify accounts associated with Russia's Internet Research Agency following the 2016 US presidential election would imply otherwise. Regardless, feature sets used in research design and methods should be carefully considered when discussing the recommendations of academic researchers versus those of industry professionals.

Macro versus micro perspective

Another concept to introduce is the macro versus micro perspective for fake account detection. The macro perspective refers to detection techniques that look at the comprehensive identity associated with an account, with the goal of identifying the entire account as fraudulent or inauthentic. The micro perspective refers to detection techniques to identify fraud or inauthenticity in the components or attributes associated with an account. Inauthenticity at the attribute level may not necessarily indicate that the entire account is fake; the user may simply be lying about pieces of their identity. Inauthenticity and a lack of consistency across multiple features may point to an entirely fabricated account.

As an example, consider an account on Facebook. Attributes of a typical account may include a profile picture, a basic biography, and a collection of text posts. Analyzing a profile picture for manipulation is an example of the micro perspective—looking at an aspect of an account and attempting to determine its authenticity. Analyzing cohesiveness across an account's purported age, gender, and cultural background is the macro perspective—looking at multiple aspects of the account and attempting to spot discrepancies that may point to a fake identity.

The line between the macro and micro perspective may be blurry at times. For example, studying the language of the account's text posts may reveal differences in authorship and personality in the posts; this may indicate that multiple people are managing an account, which in turn indicates inauthenticity at both the micro level (i.e., the text posts) and macro level (i.e., the entire account).

Solutions to the fake account problem

With the nuances of the problem now in mind, what do solutions look like? This section highlights current approaches from a macro perspective.

Define normal

At a high level, most technical solutions for fake account detection involve determining "normalcy" for a given social media platform (or community of users on the platform) and identifying accounts that deviate from this norm. So what does "normal" mean? The answer is highly variable and subjective.

To define normal, first consider the social media platform in question and the attributes that compose an account on the platform. For most online social media platforms, accounts can be interpreted as collections of attributes that fall under the following major categories: the infrastructure and network connectivity of the account, the user profile associated with the account, and the user activity on the account. Under the category of infrastructure and network connectivity, attributes may include the device(s), IP address(es), and user agent string(s) associated with an account. Under the category of user profile, attributes may include the user's name, age, and gender on a site like Facebook or LinkedIn; on other sites, such as Twitter, Tumblr, or Reddit, attributes may be limited to a username and account creation date. Finally, under the category of user activity, attributes may include friends and other forms of social connectivity and posts, likes, and other forms of engagement on the platform. The appearance and behavior of any or all attributes under these categories help establish a baseline for normalcy. Probability distributions, graphs, summary statistics, or even categorical values, depending on the detection technique, formalize concepts of normalcy.

Uncover the abnormal

Once the baseline for normal or expected account appearance or behavior has been established, there are a variety of techniques that can be used to identify accounts that deviate from this norm. This section introduces three frequently used strategies: graph analysis, temporal analysis, and machine learning.

Graph analysis

Graph analysis is a common fraud detection technique in which users or events are represented as vertices of a graph, and relationships or transactions are represented as edges of the graph. Fraudulent users or activities can then be identified by looking for anomalous structural patterns or subgraphs within the graph. Graph analysis has proven to be a highly effective technique for detecting fake users, fake reviews, fake financial transactions, and a variety of other abusive behaviors on online platforms.

Many graph analysis-based detection techniques rely on the assumption that there are specific graph structures associated with genuine communities of users [23] and that these organic connections and structures are hard to fake. For example, in [24], the authors observe that fake accounts on both social media platforms such as Twitter and Facebook and e-commerce or review platforms such as Amazon or Tripadvisor end up with many edges, which result in large and dense regions in an adjacency matrix representation of the graph. It is also assumed and often observed that fake accounts will generally have many connections to other fake accounts and few connections to authentic accounts, making it possible to identify densely connected networks of fake accounts, especially if there are known authentic users in the graph [25]. However, here is a prime example of where context and understandings of normalcy matter. As observed by the authors of [25], on certain platforms, such as Twitter or Tumblr, it is expected that users interact with strangers, meaning that connections between a known authentic account and an account of unknown authenticity does not necessarily prove anything about the status of the unknown account.

Temporal analysis

Temporal analysis techniques involve identifying anomalous patterns of activity associated with an account's behavior over time. Temporal analysis is a highly successful technique to identify automated activity (i.e., bots). For example, in [26], the authors developed a bot detection technique for Twitter on the premise that humans are indifferent to the specific second or minute in which they Tweet, meaning that an "organic" sequence of Tweet times should appear to be randomly sampled from a uniform distribution. An automated account, however, will likely result in timing distributions that are either too uniform or not uniform enough.

Temporal approaches can incorporate insights into typical activities on a platform; for example, there is an entire body of literature to draw from to understand usage of hashtags and retweets on Twitter [27]. Research has found that the activities in which authentic accounts engage are very different from those in which inauthentic accounts engage; real users spend more time interacting with accounts that are part of their social network, while fake accounts spend more time attempting to build their social network [28, 29]. For example, on Facebook, a fake account will spend more time "friending" other users than chatting with existing friends.

Machine learning

Machine-learning approaches involve translating attributes associated with an account into features. These features are then used for clustering groups of similar accounts together or differentiating between categories of accounts (e.g., fake or real). In a study done at LinkedIn, researchers used supervised machine learning to classify clusters of accounts as either malicious or legitimate [30]. Features were derived from attributes associated with user-generated profile information, such as name, email address, and company or university. Features included distributions, frequencies, and patterns found in user-generated profile text. Logistic regression, support vector machine, and random forest models were trained on LinkedIn data that was grouped by account registration IP address and registration date. The study proved highly successful and the random forest model, which achieved area under the curve (AUC) values as high as 0.98, was moved to LinkedIn's production environment. By 2015, when the study was published, the model had already been used to identify 250,000 fake accounts.

Machine-learning approaches for fake account detection have been widely explored in both industry and academia since about 2010. It is worth noting that the primary focus of this work is not on making significant advances in the machine-learning algorithms themselves. Rather it is on identifying novel attributes and transforming them into features or refining existing models to lower false positive and false negative rates. Use of logistic regression models, support vector machines, and random forests, as done in the LinkedIn study, is quite common.

Additional approaches to fake account detection

While the previous section focused on technical approaches to fake account detection, incorporating both technical and nontechnical approaches is an important strategy for combating fake accounts.

Phish the phishers

Some researchers take a honeypot approach to detecting fake accounts. For example, in [31], the researchers created Twitter bots that posted nonsensical messages. Any account that followed these bot accounts was determined to be a fake account, since any authentic, non-automated account would likely not follow or engage with these garbage accounts.

Leverage user reporting and manual review

In both research and practice, effective fake account detection often involves coupling human review with automated techniques. In the literature, relying on user reports of fake, abusive, or suspicious accounts is sometimes referred to as *crowdsourcing* bot detection [25]. In many fake account detection studies, manual review is a final step in the detection pipeline; automated techniques narrow potentially millions of fake accounts down to thousands or even hundreds for a human to review [32].

Human review is important because humans may be able to detect subtle differences between authentic and inauthentic accounts that feature sets do not capture or models fail to discover. Additionally, there is rarely ground truth data about which accounts are actually fake. Understanding why automated methods flag an account as fake (or fail to detect an account as fake) can also help researchers refine both their data sets and tools. For example, in the LinkedIn study referenced earlier, clusters of users were assigned a probability indicating how likely that cluster was to contain fake accounts. Depending on the probability, suspected fake clusters were either automatically suspended or passed to a human for manual review. Manually reviewed and labeled accounts then became training data in later model iterations [30].

Take legal action

In March of 2019, Facebook and Instagram filed a lawsuit against the People's Republic of China for "promoting the sale of fake accounts, likes and followers" [33]. By going after the industry of curated Facebook accounts and reputation, Facebook made an attempt to stem the flow of fake accounts at the "creation source" to prevent individuals and organizations (in particular, those with less resources than a nation-state actor) from simply buying accounts in order to become active players in the fake account space.

Challenge suspicious accounts

In addition to tackling the fake account industry, online platforms incorporate many checkpoints that attempt to make fake account creation as challenging as possible. CAPTCHAs and phone verification are all fraud and abuse countermeasures that most people encounter at some point, even though social network platforms try to limit the number of accounts they challenge in order the keep the user experience as frictionless as possible [30]. Facebook and other major platforms have also used the practice of quarantining users, in which suspicious accounts are sectioned off to a part of the platform where they are not interacting with the real network and then are monitored [22].

Fraud detection research at NSA

At NSA's Laboratory for Telecommunication Sciences, fake account detection is a key research focus area. Research is done at both the macro and micro level to assess the authenticity of cyber identities, often referred to as *digital personas*. Though a digital persona may encompass more than a social media account, the techniques and perspectives discussed in this article for social media fake account detection are still widely applicable. Personas are interpreted in terms of the three high-level categories discussed previously: the infrastructure and network connectivity of the persona, any account profiles or biographical details associated with the persona, and any online activities conducted by the persona.

Defining normalcy is central to this research. In practice, defining normalcy is a challenging problem since normal must be understood at both the micro level (e.g., the attributes that fall under each of the three categories) and the macro level (e.g., the persona as a whole). To define normalcy, open-source data is used to develop models, represented as probability distributions, which provide insight into the expected values of persona attributes. For example, market trend data may be used to construct a probability distribution representing web browser usage. A persona's use of a web browser other than Firefox, Chrome, or Microsoft Edge may then be used as a red flag.

Of course, the web browser example is oversimplified. Looking at one attribute in isolation is unlikely to provide much insight; models are much more useful if they provide insight into the cohesiveness and plausibility of attributes with respect to other attributes. For example, models summarizing web browser usage by geographic region could be used to identify a persona accessing the web in one region of the world with a browser that is generally only found in another region—this is a much more significant red flag. So, to better understand the relationship between attributes, models are also constructed to represent the expected values of persona attributes given values of other persona attributes. Bayesian probability is at the core of this approach—what is the probability of value X for attribute A given known value Y for attribute B? Suspicious or inauthentic personas are uncovered by looking for co-occurrences of attribute values that rarely, or never, exist in the data.

To solidify this research approach with an example, consider again Facebook's real name policy and the specific language stipulating that real names must not contain "unusual capitalization, repeating characters or punctuation" [12]. This policy has been highly controversial because there are many cases where genuine names do not meet these requirements because they do not fit what is inherently a biased interpretation of normalcy. Bias in data sets and definitions of normalcy result in false positives in practice. For example, Native American names are frequently flagged as inauthentic, resulting in wrongly suspended accounts [34]. Facebook has reportedly introduced a process allowing users to claim an "ethnic minority" or other exception if their name does not meet the real name policy. Though the approach seems well-intended, it does not solve the real issue here: Facebook—and likely many other social media companies—have limited insights into just how diverse normal can actually be. This is also why comprehensive analysis of an account at the macro level is crucial. "Does this name appear real given what I know about the user's ethnicity, cultural background, and various other demographics?" can be a much more useful question than asking "does this name appear real?" without any context.

Conclusion

So what is the state of fake account detection? If we look at reporting from the major social media platforms, we may be inclined to think detection methods are relatively successful. Facebook estimates that roughly 5% of its monthly active users are fake and reports a decline in fake account takedowns since the first quarter of 2019 thanks to their ability to detect fraud early at the account registration step [1]. However, if the last couple of decades of online fraud research can tell us anything, the reality is probably less comforting—low-grade fake accounts may be easy to detect, but sophisticated attackers have likely just become more sophisticated at dodging authenticity checks. These accounts are perhaps the ones we should worry the most about, as time and resources likely went into their curation.

Determining the state of the art in the fake account detection space is also challenging because there are few, if any, public data sets that researchers can use to test, validate, and assess their methods. A quick scan of published studies over the past five years reveals that most work is conducted on different data sets, even if the same platforms (e.g., Facebook, Twitter, and LinkedIn) are the focus of the work. There are indications that this may be changing though. For example, in September of 2019, Facebook announced the Deepfake Detection Challenge, which provided researchers with deepfake image and video data that were "freely available for the community to use…[with] few restrictions on usage" [35]. Not only did this challenge provide researchers with real data sets to use, but it also made it possible to compare competing solutions for deepfake detection. Moreover, it provided one of the first opportunities to establish benchmarks in the deepfake detection community. The challenge ended in March 2020, with 2,114 participants submitting 35,109 models for deepfake detection using the training corpus of 115,000 videos provided by Facebook. Participant models were tested against a "black box data set with challenging real world examples" [36]. The winning team's model had an accuracy of 65.18%, which Facebook touts as the "new shared baseline" in the artificial intelligence community [36]. The shared data set and new baseline represent a significant step for fake account detection research.

Of course, public data sets and methods have the danger of becoming stale: as defenders learn what techniques to employ to detect fake accounts, attackers can learn how to improve their methods for creating fake accounts. Regardless, one thing is certain—as the world continues to feel the ripple effects of elections manipulated by fake accounts and as social media companies and international organizations work to counter the potentially deadly COVID-19 conspiracies populated by fake accounts—this is a problem in need of critical attention and not going away any time soon.

References

[1] Facebook. "Community standards enforcement report: Fake accounts." Available at: https://transparency.facebook.com/community-standards-enforcement#fake-accounts.

[2] Twitter. "Transparency report: Platform manipulation." Available at: https://transparency.twitter.com/en/platform-manipulation.html.

[3] Bessi A, Ferrara E. "Social bots distort the 2016 US presidential election online discussion." *First Monday.* 2016;21(11).

[4] Bradshaw SHoward PN. "Challenging truth and trust: A global inventory of organized social media manipulation." *The Computational Propaganda Project.* Oxford Internet Institute, University of Oxford; 2018.

[5] Merriam-Webster. "Words we're watching: 'Infodemic.'" Available at: https://www.merriam-webster.com/words-at-play/words-were-watching-infodemic-meaning.

[6] Ferrara, E. "What types of COVID-19 conspiracies are populated by Twitter bots?" *First Monday.* 2020;25(6). doi: 10.5210/fm.v25i6.10633.

[7] "Coronavirus 'fake news' Twitter accounts shut down." *BBC News.* 2020 Mar 10. Available at: https://www.bbc.com/news/uk-england-hampshire-51805311.

[8] Facebook. "July 2020 coordinated inauthentic behavior report." 2020 Jul. Available at: https://about.fb.com/wp-content/uploads/2020/08/July-2020-CIB-Report.pdf.

[9] Zhang H, Duan H, Liu W, Wu J. "IPGroupRep: A novel reputation based system for anti-spam." In: *2009 Symposia and Workshops on Ubiquitous, Autonomic and Trusted Computing;* 2009 Jul 7–9; Brisbasne, QLD, Australia. doi: 10.1109/UIC-ATC.2009.15.

[10] Jøsang A, Ismail R, Boyd C. "A survey of trust and reputation systems for online service provision." *Decision Support Systems.* 2007;43(2):618–644.

[11] Ortutay B. "Real users caught in Facebook fake-name purge." *SFGATE.* 2009 May 25. Available at: https://www.sfgate.com/business/article/Real-users-caught-in-Facebook-fake-name-purge-3231397.php.

[12] Facebook. "Help center: What names are allowed on Facebook?" Available at: https://www.facebook.com/help/112146705538576.

[13] Facebook Safety. 2015 June 1. Available at: https://www.facebook.com/fbsafety/posts/861043117266861.

[14] Lazer DMJ, Baum M, Benkler Y, Berinsky AJ, Greenhill KM, Menczer F, Metzger MJ, Nyhan B, Pennycook G, Rothschild D, et al. "The science of fake news." *Science.* 2018;359(6380):1094–1096.

[15] Heydari A, Tavakoli MA, Salim N, Hedari Z. "Detection of review spam: A survey." *Expert Systems with Applications.* 2015;42(7):3634–3642.

[16] Facebook. Community Standards. 17. Misrepresentation. Available at: https://www.facebook.com/communitystandards/misrepresentation.

[17] Facebook. Community Standards. 20. Inauthentic behavior. Available at: https://www.facebook.com/communitystandards/inauthentic_behavior.

[18] Twitter. Help Center. Twitter rules and policies. Impersonation policy. Available at: https://help.twitter.com/en/rules-and-policies/twitter-impersonation-policy.

[19] Twitter. Help Center. General guidelines and policies. Abusive behavior. Available at: https://help.twitter.com/en/rules-and-policies/abusive-behavior.

[20] Twitter. Help Center. General guidelines and policies. Civic integrity policy. Available at: https://help.twitter.com/en/rules-and-policies/election-integrity-policy.

[21] Twitter. Help Center. General guidelines and policies. Automation rules. Available at: https://help.twitter.com/en/rules-and-policies/twitter-automation.

[22] Spam Fighting 2016: Spam Fighting@Scale. 2016 Nov 3. Available at: https://atscaleconference.com/events/spam-fighting-2016/.

[23] Prakash A, Sridharan A, Seshadri M, Machiraju S, Faloutsos C. "EigenSpokes: Surprising patterns and scalable community chipping in large graphs." Available at: http://www.cs.cmu.edu/afs/cs.cmu.edu/user/christos/www/PUBLICATIONS/pakdd10-eigenspokes.pdf.

[24] Hooi B, Song HA, Beutel A, Shah N, Shin K, Faloutsos C. "FRAUDAR: Bounding graph fraud in the face of camouflage." In: *KDD '16: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining;* 2016 Aug 13–17; San Francisco, CA: pp. 895–904. doi: 10.1145/2939672.2939747.

[25] Ferrara E, Varol O, Davis C, Menczer F, Flammini A. In: *Communications of the ACM;* 2016 Jun. "The rise of social bots." doi: 10.1145/2818717.

[26] Zhang CM, Paxson V. "Detecting and analyzing automated activity on Twitter." In: Spring N, Riley GF, editors. *Passive and Active Measurement. PAM 2011. Lecture Notes in Computer Science, vol 6579.* Springer, Berlin, Heidelberg. pp. pp 102–111. doi: 10.1007/978-3-642-19260-9_11.

[27] Bruns A, Stieglitz S. 2012. "Quantitative approaches to comparing communication patterns on Twitter." Queensland University of Technology. Brisbane, Australia. Available at: https://eprints.qut.edu.au/55823/1/Quantitative_Approaches_to_Comparing_Communication_Patterns_on_Twitter.pdf.

[28] Wang G, Konolige T, Wilson C, Wang X, Zheng H, Zhao BY. "You are how you click: Clickstream analysis for Sybil detection." In: *SEC '13: Proceedings of the 22nd USENIX Security Symposium;* 2013 Aug 14–16; Washington, DC: pp. 241–256.

[29] Yang Z, Wilson C, Wang X, Gao T, Zhao BY, Dai Y. (2014). "Uncovering social network Sybils in the wild." *ACM Transactions on Knowledge Discovery from Data.* 2014;8(1). doi: 10.1145/2556609.

[30] Xiao C, Freeman DM, Hwa T. "Detecting clusters of fake accounts in online social networks." In: *Proceedings of the 8th ACM Workshop on Artificial Intelligence and Security (AISec '15);* 2015 Oct 16; Denver, Colorado: pp. 91–101. doi: 10.1145/2808769.2808779.

[31] Lee K, Eoff BD, Caverlee J. "Seven months with the devils: A long-term study of content polluters on Twitter." In: *Fifth International AAAI Conference on Weblogs and Social Media;* 2011 Jul.

[32] Cao Q, Sirivianos M, Yang X, Pregueiro T. "Aiding the detection of fake accounts in large scale social online services." Available at: https://www.usenix.org/system/files/conference/nsdi12/nsdi12-final42_2.pdf.

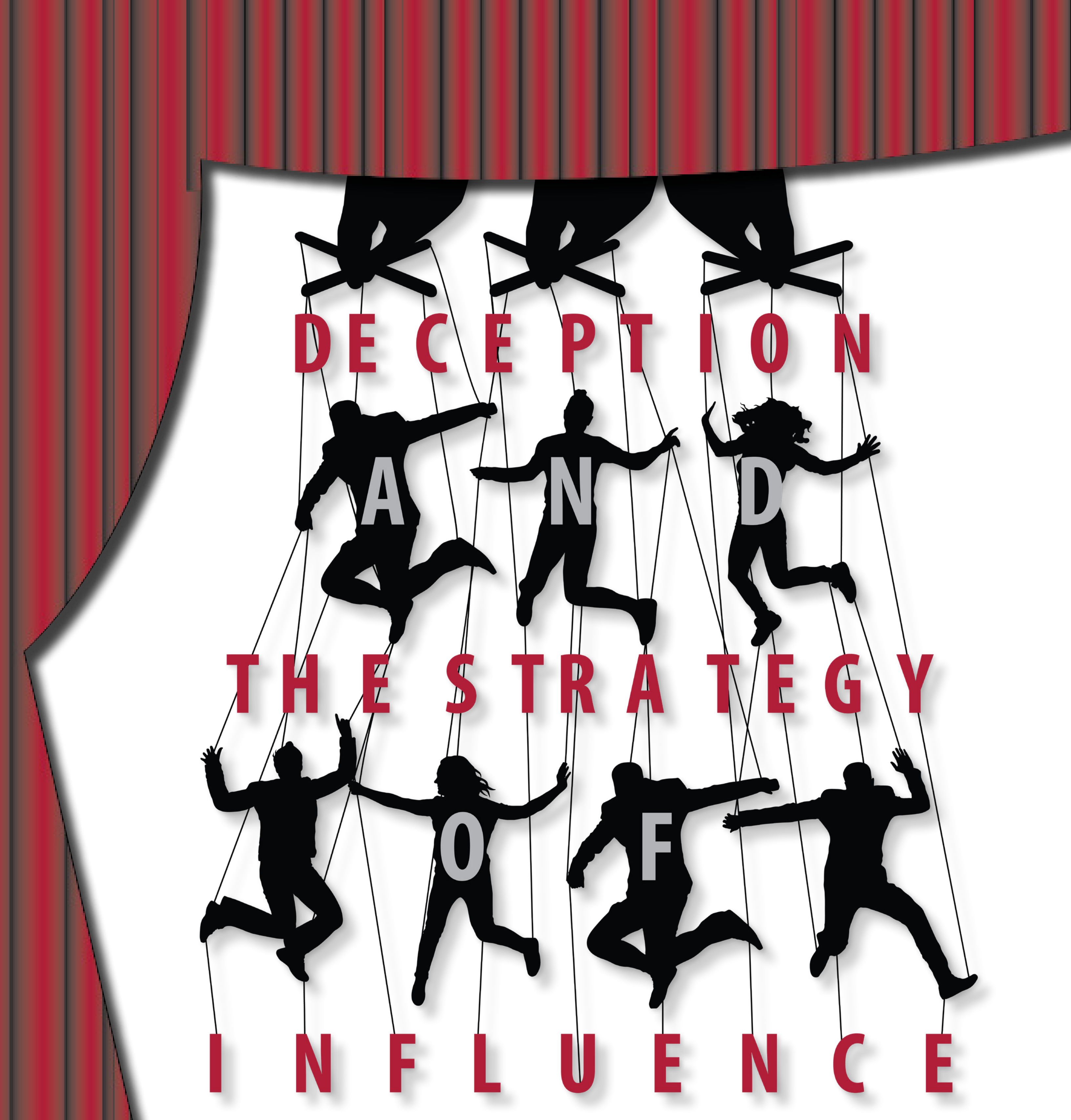
[33] Grewal P. "Sale of fake accounts, likes, and followers." 2019 Mar 1. Facebook. Available at: https://about.fb.com/news/2019/03/sale-of-fake-accounts-likes-and-followers/.

[34] Bowman J. "Facebook flags aboriginal names as not 'authentic.'" CBC. 2015 Feb 25. Available at: https://www.cbc.ca/news/trending/facebook-flags-aboriginal-names-as-not-authentic-1.2970993.

[35] Schroepfer M. "Creating a data set and a challenge for deepfakes." *Facebook AI.* 2019 Sep 5. Available at: https://ai.facebook.com/blog/deepfake-detection-challenge/.

[36] Facebook AI. "Deepfake Detection Challenge Results: An open initiative to advance AI." 2020 Jun 12. Available at: https://ai.facebook.com/blog/deepfake-detection-challenge-results-an-open-initiative-to-advance-ai/.

DECEPTION AND THE STRATEGY OF INFLUENCE



[Photo credit: iStock.com/Nosyrevy]

By Brian B., William C. Fleshman, Kevin H., Ryan Kaliszewski, Shawn R.

Organizations have long used deception as a means to exert influence in pursuit of their agendas. In particular, information operations such as propaganda distribution, support of antigovernment protest, and revelation of politically and socially damaging secrets were abundant during World War II and the Cold War. A key component of each of these efforts is deceiving the targets by obscuring intent and identity. Information from a trusted source is more influential than information from an adversary and therefore more likely to sway opinions.

The ubiquitous adoption of social media, characterized by user-generated and peer-disseminated content, has notably increased the frequency, scale, and efficacy of influence operations worldwide. In this article, we explore how methods of deception including audience building, media hijacking, and community subversion inform the techniques and tradecraft of today's influence operators. We then discuss how a properly equipped and informed public can diagnose and counter malign influence operations.

History and background

The use of influence and deception as weapons is not a new concept. The famous general and philosopher Sun Tzu (545 BC–470 BC) said that "All warfare is based on deception" and "The supreme art of war is to subdue the enemy without fighting." Using information, both true and false, to confuse, divide, and demoralize opponents is a tactic that has been exploited for millennia.

The British agenda in Nazi Germany

Between 1941 and 1943, Der Chef operated as the spokesperson of an illegal radio station in Nazi Germany, called GS-1 [1]. Der Chef acted as a loyalist to the Nazi cause and lambasted Nazi Party officials who he accused of being lazy, corrupt, and engaging in various sexual improprieties; meanwhile, he praised the bravery and devotion of German troops on the front line. In reality, Der Chef was a German refugee living in and recording and broadcasting from England. GS-1 was part of England's black propaganda engine, run by Sefton Delmer, which broadcast US jazz, German dance music, and sports scores, as well as reporting news to the public with a secret British agenda.

Der Chef would use reported local news and facts whenever possible to undermine the German populace's faith in Nazi leadership. Since facts are difficult to dispute, information used in this way is powerful and persuasive. Furthermore, by dispersing propaganda among music and news reports, Der Chef attracted new listeners and obfuscated his true intentions from his audience. Delmer described this approach to propaganda as "Cover, cover, dirt, cover, cover" while we refer to it as *pump-and-pivot*. Influence operators use this technique by drawing followers in through benign, popular content and then pivoting to malign influence.

The Communist agenda in Latin America

In the 1960's, anti-American sentiment in Latin America led to footholds for communist elements. Compounding these problems, a letter that was signed by J. Edgar Hoover congratulating Thomas Brady for his efforts in the joint FBI/CIA operation to overthrow the Brazilian government was leaked to the press [2].

It turned out that the letter was fake, forged by the Czechoslovak Intelligence Service (CIS) to undermine US interests. The sensationalism of the story encouraged the media to release the story with little scrutiny or fact-checking. In addition, the anti-US sentiment of the population and confirmation bias caused the story to be met with little skepticism from the Latin American public. Predisposition and sensationalism make populations vulnerable to influence operations.

The Islamic State agenda in Libya

In November of 2014, major news outlets reported that fighters loyal to the Islamic State in Iraq and Syria were in complete control of the city of Derna and that the fighters were taking advantage of political chaos to rapidly expand their presence westward and along the coast [3, 4, 5]. At that time, the caliphate seemed to be growing at an unprecedented rate. The Islamic State's strong expansion into Libya seemed to signal a groundswell of support and unity in the movement. The only problem—Derna was a hotbed, contested by no less than three militant groups [6, 7, 8, 9], and while the Islamic State had a presence there, their control was anything but complete. By controlling all of the information in the area, the Islamic State could write their own narrative of events and use mainstream media to disseminate it.

Perceived legitimacy

People naturally associate with others with similar ideologies. These groups are often described as *echo-chambers or filter-bubbles*, amplifying the ideas common to the group while squashing the flow of discourse contrary to their shared beliefs. Once an influence operator has established a *persona*, or false presence, the insular nature of these bubbles stifles dissent and makes the group more susceptible to influence. In social media, these groups are referred to as *communities*; to an influence operator, they are an audience.

To conduct an influence operation, operators such as Der Chef need a receptive audience. The two ways to gain an audience are to build one, such as what Der Chef did by playing music, or to hijack an existing audience, as in the CIS and Islamic State cases. In both of these scenarios, the influence operator needs to appear as though they are providing a legitimate service to their targets.

In order to appear legitimate, it is important for the influence operator to avoid scrutiny. This is why the pump-and-pivot tactics are so common. During the *pump* phase, the operator manipulates the environment to increase legitimacy by, for example, appealing to a biased target, being the only source of information, using facts interspersed with falsehoods or directing targets to other compromised sources.

Deceptive media has been used to build legitimacy for some time, but the cost of producing quality material has traditionally reduced its scale. The impact on perceived legitimacy due to ubiquitous access to targeted synthetic media (e.g., deepfakes) generation will likely be profound. Malign actors will no longer need to draw on organic material that moves their narrative forward among their devoted following. They will instead be able to support their activities and narratives with synthetic content that appears to be factual evidence. This will decrease the time needed to both build their legitimacy and reach their malign influence goals. Successful operations will likely only be detected and mitigated by social media platforms as users will not have sufficient information to make an accurate assessment.

Audience building

Social media has scaled audience building by providing targeted advertising, automation, and access to millions of users. These tools can be leveraged to precisely target demographics and build an audience out of previously disjoint subgroups [10]. Operators draw users to compromised information sources by providing information of interest with the intent of making the operator's persona and the information sources part of their targets' daily routines. During this time, the information sources provide a legitimate service (i.e., desired content). For example, recent reporting on Russia's Internet Research Agency (IRA) in 2017 demonstrated persistent attempts by IRA personas to cover local interest stories first, amplifying the spread of the stories with the help of automated accounts or *bots* [10]. By reporting first, and through the careful use of keywords, the bots landed at the top of trending news feeds and search results, building their audience.

Social media platforms facilitate the *pivot* phase by allowing users to reinvent their accounts without notifying the users who are within their network. From the point of view of the users it appears that a totally different actor has begun to contribute to their trusted information stream. This allows the operator to inject information from the compromised sources and amplify content arising from the community. Subsequently the community will move by itself, with the influence operator keeping the focus on the desired narrative. These behaviors of account reinvention can be observed in real time, but can be extremely difficult for the average user to observe in retrospect.

Synthetic media can play a substantial role in audience building—conversational bots can be leveraged to disseminate useful information at scale while engaging their audience, content sought by key audiences can be generated reducing cost and likelihood of being attributed, and coordinated automated

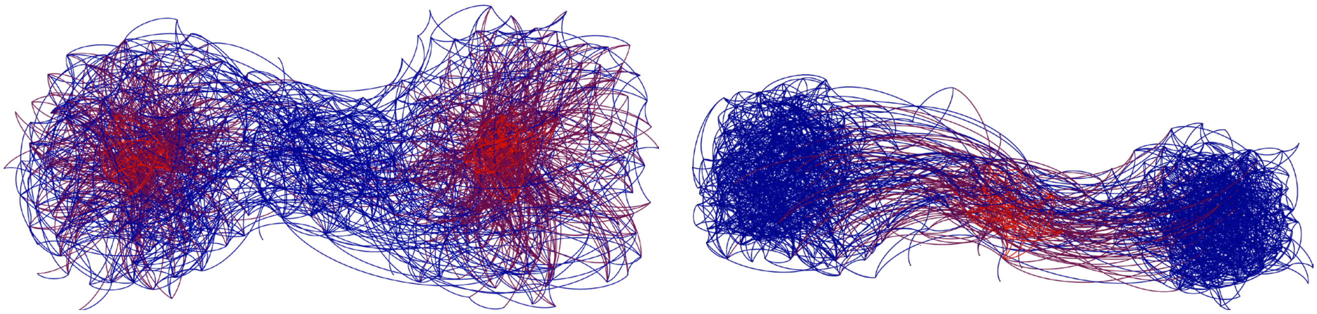


FIGURE 1. This simulated data shows the flow of information from influence operators into a two-sided discourse. Vertices represent users, red edges represent communication from embedded operators, blue edges represent communications from legitimate users. (Left) The embedded personas are attempting to influence both sides of the discourse from the community cores. (Right) The embedded personas act as bridges between the communities in order to develop malign confrontation.

but realistically human bots can give the appearance of social consensus. The challenge of identifying these behaviors at scale to mitigate their impact or remove the associated campaigns entirely will be a persistent challenge for platforms for the foreseeable future.

Media hijacking

In today's media environment, the rewards for possessing timely, exclusive reporting on a topic can incentivize publication before rigorous fact-checking is available. This is particularly true of content that is emotionally charged—sensationalism drives increased readership and engagement. For example, the increased risk to journalists in militant sites reduces the availability of professional journalism in a region, but battlefield reporting is valuable news. By providing professional quality reporting in such a region, influence operators can have their reports repeated and amplified by the international press, producing an immediate audience.

The rush-to-publish environment facilitates influence operators to use synthetic media to amplify deceptive narratives. It is now possible to generate realistic video and audio of well-known personalities at minimal cost. This capability will likely be leveraged by influence operators to divert attention from legitimate but damaging news stories as well as create confusion in times of uncertainty. Media companies will need to balance the desire to be first-to-publish with the possibility of providing a platform to influence operators.

Community subversion

Some influence operations are not as direct as Der Chef decrying the Nazi leadership or the Islamic State controlling the narrative of battle around Derna. Influence operators can use the perceived presence of numbers to change the narrative of a community, a tactic which is referred to as *community subversion*. For example, the Saudi Arabian government was accused of using bots to undermine anti-Saudi hashtags and inflate pro-Saudi positions surrounding conflict with Qatar [11], and Iran has been accused of using more than 140 Reddit accounts to promote anti-Saudi, anti-Israeli, and pro-Palestinian narratives [12].

Those two examples of community subversion illustrate *bolstering* and *degrading* of communities. Bolstering a community is a subversion technique where influence operators artificially increase support in order to embolden legitimate users. Degrading a community is a tactic where operators sow division within the community.

Influence operators can also interfere with a community through a *denial of service attack*. By flooding the community with noise, they can either trigger a platform's automated spam filter or prevent legitimate users from communicating in an organic way. This was seen firsthand in 2014 when bots entered a human rights Twitter community centered on protests in Mexico and filled it with spam [13], preventing protesters from coordinating to avoid police. The left-hand plot of figure 1 graphically shows how influence operators would be situated in the above attacks.

Other examples of community subversion come from Russian IRA influencers, who in 2016 were observed operating on both sides of the Black Lives Matter hashtag [14]. By doing so, the influencers directly inflamed the discourse on both sides by moving both conversations to the extremes.

Then, in 2018, researchers observed Russian IRA influencers acting as a bridge between polarized groups in the vaccination debate [15]. They were forcing communication and specifically argumentation between groups of people on opposite sides of the issue by using the hashtag #VaccinateUS for both pro-vaccine and anti-vaccine content. Furthermore, the #VaccinateUS tweets generally included other emotionally charged topics from US culture in order to maximize division. The right-hand side of figure 1 shows an example of such behaviors—connecting two communities that would otherwise be loosely connected or disconnected.

Automated community-to-community interactions can now be scheduled and convincingly generated based on current conversation. Present and timely synthetic media will increase the chance that these bridging operations succeed. Community members should question new narratives entering their networks from previously unknown accounts. This is particularly true for controversial or confirmation-bias affirming narratives.

Diagnosis of influence operations

Efforts to detect influence operations leverage the behaviors that result from operators' desire to minimize their fingerprints on the larger conversations. The goals of minimizing direct involvement and trying to build an audience are often in contention over the course of an influence campaign. This results in opportunities to identify coordinated networks of accounts. The basic tools for these observations are *community detection* and *content analysis*.

Community detection is an algorithmic way to detect internal structure in a network graph such as comment, email, or retweet graphs. In particular, if one constructs their graph in such a way as to indicate positive sentiment between nodes, then such a graph can be viewed as an indicator for likely confirmation bias. Content analysis, such as topic modeling [16, 17] or text summarization algorithms, can isolate the themes in the discourse and be used to understand the narrative and focus within a community as well as the flow of discussion between communities.

While the boundary between social media and mainstream reporting is becoming ever more porous, efforts to mitigate the spread of influence operations should pay close attention to the beginnings of discourse. Integration of dynamic content analysis can highlight the construction of new narratives, and particular attention should be focused on narratives with extreme amplification during this time period. This sort of analysis is particularly important regarding stories around which the information environment is particularly constrained.

As a discourse matures, accounts with deceptive behavior should be analyzed closely. Aberrant behavior such as removing a great deal of previous content, changing outward appearance, or a distinct change in quality or focus of shared content can indicate a pump-and-pivot. Established communities that suddenly shift focus or trigger flags such as spam filters can be indications of a pump-and-pivot or community subversion. The application of social bot classifiers can help separate artificial amplification from organic growth, highlighting accounts that attempt to inject themselves into the discourse.

Brigading, or accounts that join a community for purposes other than joining the discourse, could signal a community subversion effort, especially if the brigading is coordinated or in excessively large numbers. More specifically, if the number of interactions between two polarized communities increases, then further analysis can be done to investigate whether the increase is natural or caused by a deceptive force.

Detecting influence operations through technology

Another way to identify influence operations is by how they choose to interact with social media. Users interact with a social media platform via a *client*. While many users operate with first party clients, a number of third-party clients exist to facilitate automation, provide a different look and feel, allow for management of multiple accounts across different platforms, and display analytics of audience engagement [18].

Raw access to the platform's application programming interface (API) can provide the ability to spoof geolocations, IP addresses, and timing of posts to

Return to CONTENTS

appear to be elsewhere in the world [19]. Influence operators frequently establish personas in different countries to conduct influence operations more effectively [10] and build legitimacy. Operators can create a custom client to increase their efficiency, allowing one person to control dozens or hundreds of accounts with varying degrees of automation.

We refer to the collection of clients used by an account to interact with a social media platform as its *technology stack*. Analysis of a technology stack for a specific account can help identify automated accounts via the presence of bot clients [20]. One hypothesis is that a human being in charge of an account will either be satisfied by standard, first-party or popular third-party, social media clients or will have some reason to seek out nonstandard clients. For each nonstandard third-party client there is likely a community of users that use that client.

Fingerprinting of technology stack communities can be done at scale by forming a bipartite graph of technologies and users within a social media platform. Detecting and removing outlier user and client nodes generates connected subcomponents of the graph. After dimensionality reduction, low-dimensional clustering algorithms can detect *clusters of clients* which are used by similar users and *clusters of users* with similar client usage. This allows for efficient analysis of subsets of users based upon their technology choices, such as in figure 2. The use of nonstandard clients with restrictive access requirements may indicate a relationship within the subset of users.

Identifying suspicious actors via their technology stack allows one to detect and mitigate influence operations early. This assists in preventing malign influence operators from achieving their goals and increases the cost necessary to deceive a target audience. Even if a campaign cannot be prevented using the above technique, post-hoc analysis can provide critical indicators which can be used by government or industry to increase the cost of future operations by preventing reuse of technology stacks by operators.

Conclusion

Tools to create, curate, and automate convincingly human-created media (i.e., audio, text, image, and video) are readily available. These tools are already being used by influence operators to gain legitimacy, build their audiences, hijack traditional media, and subvert communities. This creates a persistent challenge for users, platforms, and media companies to address. A commitment on the part of the platforms to maintain technological solutions to identifying state-of-the-art synthetic media and influence campaigns, automate responses to identified activities, and provide context to the users would help mitigate these activities. Easy access to tools and knowledge to identify and respond to influence operations will help limit their impact. A collaboration between media companies and technology platforms to help identify synthetic media before broadcast will help reduce the likelihood of such broadcasts being leveraged by influence operators. In short, education, collaboration, and technology can be used together to help blunt the impact of synthetic media on public discourse.

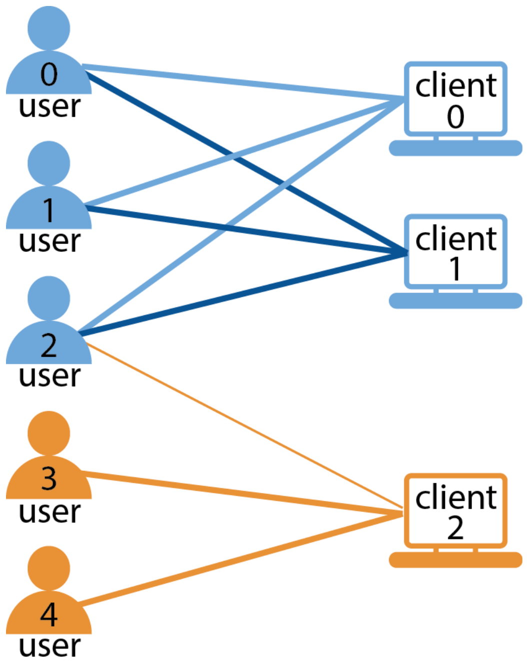


FIGURE 2. This bipartite graph illustrates user and client interactions with line thickness indicating the proportion of a user's total interactions with a social media platform using a particular client. Two clusters are shown, colored blue and orange respectively. Users 0, 1, and 2 use the blue technology stack (clients 0 and 1) with similar frequency, even though user 2 will rarely use client 2. Users 3 and 4 use the orange technology stack (client 2) exclusively and more frequently than user 2.

Public discourse is, and has been, consistently influenced by malign actors. The growth of social media companies over the last decade has created a new dynamic in this system with which society has yet to find balance. The first tool necessary for finding that balance is knowledge—knowledge of when actors are trying to influence, knowledge of actors' intent in what to influence, and knowledge of who is acting in concert with an influence campaign. This knowledge cannot come from historical analysis alone as influence operations are dynamic from account creation to daily targeting. The flow of specific discussions through a network can be used to identify the target events and communities of an influence operation. Examining the specific behaviors of accounts can identify out-of-band coordination and automation. Together, these can give the public the knowledge to confidently interact with social media.

References

[1] Garnett D. *The Secret History of PWE: Political Warfare Executive 1939–1945.* St Ermin's Press; 2002. ISBN 1-903608-08-2.

[2] Bittman L. *The KGB and Soviet Disinformation: An Insider's View.* Pergamon Press; 1985. ISBN 0080315720.

[3] Cruickshank P, Robertson N, Lister T, Karadsheh J. "ISIS comes to Libya." *CNN.* 2014 Nov 18. Available at: https://www.cnn.com/2014/11/18/world/isis-libya/index.html.

[4] Stout D. "Report: ISIS takes control of a Libyan city." *Time.* 2014 Nov 19. Available at: https://time.com/3593885/isis-libya-iraq-syria-terrorism-derna/.

[5] Fowler E. "From Raqqa to Derna: Exceptionalism in expansionism." *Jadaliyya.* 2014 Dec 4. Available at: https://www.jadaliyya.com/Details/31549.

[6] BBC News. "Libya violence: Activists beheaded in Derna." 2014 Nov 11. Available at: https://www.bbc.com/news/world-africa-30011640.

[7] Stephen C. "US expresses fears as Isis takes control of northern Libyan town." 2014 Dec 6. *The Guardian.* Available at: https://www.theguardian.com/world/2014/dec/06/us-fears-isis-nothern-libya-derna.

[8] Joscelyn T. "Islamic State 'province' advances in and around Libyan city of Sirte." *FDD's Long War Journal.* 2015 Jun 9. Available at: https://www.longwarjournal.org/archives/2015/06/islamic-state-branch-advances-in-and-around-libyan-city-of-sirte.php.

[9] Keilberth VM, Reuter C. "The Islamic State's dangerous gains in Libya." 2015 Feb 23. *Spiegel International.* Available at: https://www.spiegel.de/international/world/islamic-state-advance-in-libya-could-present-threat-to-europe-a-1019976.html.

[10] Spangher A, Ranade G, Nushi B, Fourney A, Horvitz E. "Analysis of strategy and spread of Russia-sponsored content in the US in 2017." 2018 Oct 23. Cornell University Library. Available at: https://arxiv.org/abs/1810.10033v1.

[11] "Saudi bots use 'hashtag poisoning' to spread propaganda." *The Peninsula, Qatar's Daily Newspaper*. 2018 Feb 5. Available at: https://www.thepeninsulaqatar.com/article/05/02/2018/Saudi-bots-use-%E2%80%98hashtag-poisoning%E2%80%99-to-spread-propaganda.

[12] "Threat research: Suspected Iranian influence operation leverages network of inauthentic news sites & social media targeting audiences in US, UK, Latin America, Middle East." *FireEye Intelligence.* 2018 Aug 21. Available at: https://www.fireeye.com/blog/threat-research/2018/08/suspected-iranian-influence-operation.html.

[13] Porup JM. "How Mexican Twitter bots shut down dissent." *VICE News*. 2015 Aug 24. Available at: https://www.vice.com/en_us/article/z4maww/how-mexican-twitter-bots-shut-down-dissent.

[14] Arif A, Stewart L, Starbird K. "Acting the part: Examining information operations within #BlackLivesMatter discourse." In: *Proceedings of the ACM on Human-Computer Interaction*, Vol. 2, CSCW, Article 20; 2018 Nov. Available at: https://doi.org/10.1145/3274289.

[15] Broniatowski A, Jamison A, Qi S, AlKulaib L, Chen T, Benton A, Quinn S, Dredze M. "Weaponized health communication: Twitter bots and Russian trolls amplify the vaccine debate." In: *American Journal of Public Health* 108, no. 10; 2018 Oct 1: pp. 1378–1384.

[16] Blei DM, Ng AY, Jordan MI. "Latent Dirichlet allocation." *Journal of Machine Learning Research*. 2003;3:993–1022. Available at: http://dx.doi.org/10.1162/jmlr.2003.3.4-5.993.

[17] Papadimitriou C, Raghavan P, Tamaki H, Vempala S. "Latent semantic indexing: A probabilistic analysis." In: *PODS '98: Proceedings of the 17th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems;* 1998 May; pp. 159–168. Available at: https://doi.org/10.1145/275487.275505.

[18] Lua A. "The 25 top social media management tools for businesses of all sizes." *Buffer Marketing Library*. 2019 Sept 25. Available at: https://buffer.com/library/social-media-management-tools.

[19] "NovaPress publisher." *DNATIVE.RU*. Available at: https://dnative.ru/catalog/obzory/otlozhennyj-posting-v-instagram/novapress-publisher/ [accessed 2019 Dec 6].

[20] Bhamidipati S, Heitzmann A. "Twitter's identification of bots." *Medium.com*. 2018 Jun 21. Available at: https://medium.com/@sravanb/twitters-identification-of-bots-2ac2fbb34f33.

Return to CONTENTS

More than Meets the Eye: What a Photo Can Reveal About a Camera

Sarah Charlton

Massive quantities of digital media are created and shared online every day. For example, 350 million photos are uploaded every day to Facebook [1]. In some contexts, such as for evidence in court, it is important to establish whether the digital imagery in question was obtained with a particular camera [2]. This topic is known generally as "source camera identification," and can be thought of as digital ballistics or fingerprinting—linking images and videos to cameras like linking bullets to guns or fingerprints to people.

[Photo credit: iStock.com/Azure-Dragon/faberfoto_it/Vaniatos/Pixabay]

Introduction

Recently, with the rise of Deepfakes and fake news, understanding digital media is critically important. If we can establish that images or videos are associated with a real, physical camera, at the very least we have established that the imagery is not computer generated. Original real-camera images and videos can be manipulated after their creation, so additional investigation is needed to determine whether digital media remains genuine and is not maliciously manipulated.

The easiest way to investigate a digital media file is to examine its metadata. Besides the visual content (and for videos, audio content as well), digital media files store all kinds of auxiliary information. Many digital cameras embed information about date and time, GPS location, camera settings, make and model of the camera, and more. A few cameras embed a unique serial number that can distinguish individual cameras, but this is much less common. However, this digital metadata can be easily removed or edited; furthermore, metadata is often stripped from files as part of the social media upload pipeline.

Is there a more reliable way to link digital media to their source cameras? Yes.

The digital camera

Inside every digital camera there is a light-sensing chip, either a charge-couple device (CCD; see figure 1) or a complimentary metal-oxide semiconductor (CMOS). Fabricated out of silicon, these sensors are the physical pixels of the camera. The chip absorbs light photons and converts the light into an electric charge via the photoelectric effect. This electric charge is then read off the chip and converted into a digital signal to create a digital image.

The CCD was invented in 1969 at Bell Labs by Willard Boyle and George E. Smith, and their work was based on prior metal-oxide semiconductor (MOS) research at Bell Labs. At first, CCD and MOS devices were intended for general memory storage; however, applications to imaging quickly became evident. The first digital camera was invented by Kodak in 1975. In 2009, Boyle and Smith were awarded the Nobel Prize in Physics for their invention of the CCD. Note that the physical phenomenon that the CCD exploits, the photoelectric effect, was described by Albert Einstein in a 1905 paper. Einstein was awarded the Nobel Prize in 1921 for this work[a]. So the next time you take out your camera to take a picture of the salad you are having for lunch or a video of your dog doing something funny—consider that this technology is enabled by *two* Nobel prizes!

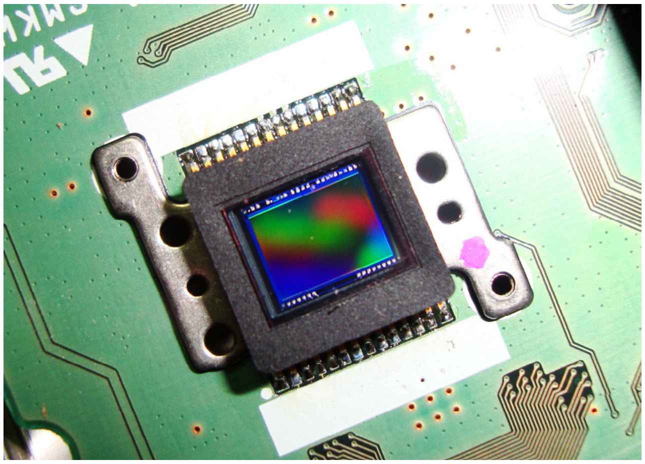


FIGURE 1. Every digital camera contains a light-sensing chip, similar to the one in this photo. [Public domain image].

The camera fingerprint

These imaging sensors, the CCD or CMOS, impart a very subtle "noise pattern" on each image that they capture. (This occurs on videos too, but for simplicity we will discuss the case of images first.) This noise is due to slight variations in the silicon from pixel to pixel. The technical term for this noise pattern is the *photo-response non-uniformity,* or PRNU. This PRNU noise is very small and does not affect the image quality to the point that humans can notice it by eye. However, images can be processed in a certain way to extract and enhance this pattern.

Researchers at Binghamton University, Jessica Fridrich, Miroslav Goljan, and their students, first discovered that this CCD/CMOS noise pattern can be used to uniquely identify the imaging sensor and, hence, uniquely identify or "fingerprint" a camera. Fridrich et al. first published this result in 2005 [3] and since then have been world leaders in camera fingerprinting research.

The use cases for camera fingerprinting are analogous to human fingerprinting or firearms ballistics.

[a] Einstein was not awarded the Nobel Prize for his Theory of General Relativity, as is commonly thought.

Images can be linked to a specific camera—not just a particular make or model of camera, but a single unique device—just like fingerprints can be linked to a specific person or a bullet linked to the exact gun from which it was fired. The PRNU camera fingerprint is a powerful tool because it is mostly independent of the scene in the picture, and it has been shown to be stable over the lifetime of the camera [3, 4]. Furthermore, all CCD and CMOS sensors impart a unique PRNU pattern, so all cameras have a unique sensor fingerprint [5].

Prior to the work at Binghamton, other researchers had considered examining the behavior of the CCD/CMOS sensors to link images to cameras. This earlier work focused only on the pixel defects on the sensor, where a very small number of pixels might get "stuck" and only read at a certain value [2]. Some pixels are always "hot" and only register the maximum possible value, and some are "dead" and always read as zero. However, this approach is not universally effective because some cameras will not have any stuck pixels.

In contrast, the PRNU camera fingerprint considers all pixels in an image. This provides information content which is rich enough to distinguish the very subtle differences between PRNU patterns in different cameras. In the past 15 years, the process for PRNU fingerprinting has been well established and it is now the gold standard for source digital camera identification.

Extracting the fingerprint

The following mathematical model can be used to characterize the resultant image from a digital camera. Let $\mathbf{I}i,j$ be the image signal at pixel $[i, j]$, $i = 1, \dots, m$, $j = 1, \dots, n$, where the image is m by n pixels. Let $\mathbf{Y}i,j$ be the true light intensity at the same pixel $[i, j]$. Then, dropping pixel indices for readability, the image $\mathbf{I}$ can be modeled as:

$$\mathbf{I} = g^{\gamma}\,[(1 + \mathbf{K})\mathbf{Y} + \mathbf{\Lambda} + \mathbf{\Omega}]^{\gamma} + \mathbf{Q}. \qquad (1)$$

The multiplicative factor g is the gain factor, and γ is the gamma correction factor. The matrices Λ, $\mathbf{\Omega}$ and $\mathbf{Q}$ represent several types of noise: dark current, shot noise, and quantization noise, respectively. Finally, that leaves $\mathbf{K}$, which is the PRNU factor we are interested in. All matrix operations are performed element-wise.

Equation (1) can be simplified by combining terms, factoring, and using a Taylor expansion:

$$\mathbf{I} = \mathbf{I}_0 + \mathbf{I}_0\mathbf{K} + \mathbf{\Theta}; \qquad (2)$$

See [4] for the details. This simpler version of the equation shows that the output of a digital camera can be modeled as the "true scene" ($\mathbf{I}_0$), plus the PRNU "fingerprint" ($\mathbf{I}_0\mathbf{K}$), plus some noise ($\mathbf{\Theta}$).

Given the sensor model (2) above, the PRNU factor, $\mathbf{K}$, for a digital device can be estimated from the images produced by that device. Let F be a denoising filter. Then the *noise residual*, $\mathbf{W}$, is defined as,

$$\mathbf{W} = \mathbf{I} - \mathrm{F}(\mathbf{I}) = \mathbf{IK} + \Xi. \qquad (3)$$

Notice this is opposite of a typical application of image denoising, where one would keep the nice and clean image and throw out the noise. In this case, the denoised image, F($\mathbf{I}$), is subtracted from the original image, $\mathbf{I}$, and we keep only the "noise" part of the original image. This noise is composed of the PRNU pattern noise $\mathbf{IK}$, and all other sources of noise are consolidated and represented by Ξ.

Given model (3) and some basic assumptions about the variance of Ξ, the maximum likelihood estimator (MLE) for $\mathbf{K}$ is

$$\hat{\mathbf{K}} = \frac{\sum_{k=1}^{N} \mathbf{W}_k \mathbf{I}_k}{\sum_{k=1}^{N} \mathbf{I}_k^2}, \qquad (4)$$

where $\mathbf{N}$ is the number of images used for estimation, which we call the fingerprint *mass*. Alternatively, such as in [6], a simple average of the noise residuals is used as the fingerprint estimate:

$$\hat{\mathbf{K}} = \frac{\sum_{i=1}^{N} \mathbf{W}_i}{N}. \qquad (5)$$

There are three steps that are standard practice in the literature, but that are not explicit in the camera fingerprint equations above. One, the "bad" pixels in each image $\mathbf{I}$ should be masked from the fingerprint calculation. The PRNU signature is a result of inhomogeneity in the CCD's absorption of light, so if regions of the image are too dark or too bright, the PRNU fingerprint is not captured very well in those pixels because those regions of the CCD absorbed either too little or too much light. Therefore, these too dark or too bright pixels are dropped from the fingerprint calculations. Second and third, after calculating $\mathbf{K}$, zero-mean and Wiener filter post-processing steps

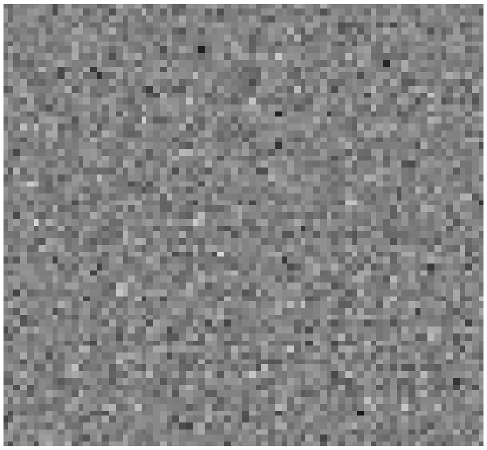


FIGURE 2. Every digital camera has a unique fingerprint that is made up of a matrix; each entry in the matrix is a value between 0 and 1 that can be visualized as a gray value that varies between black (value 0) and white (value 1).

should be applied to obtain a better fingerprint [4, 7]. Together, the zero-mean and Wiener filter help eliminate the non-unique patterns that arise from the color filter array on the CCD chip.

In their original work, Fridrich et al. presented a camera fingerprint estimation technique using all three red, green, and blue color channels of an image. Essentially, they would extract a fingerprint from each channel using the equations (3) and (4), and then calculate a weighted average of the three fingerprints to obtain the grayscale fingerprint.

$$\hat{K} = 0.299\,\hat{K}_r + 0.587\,\hat{K}_g + 0.114\,\hat{K}_b. \qquad (6)$$

Later, Gisolf et al. ran several tests to investigate the effects of color channel selection on fingerprint performance and found that the choice of color channel had very little effect. The authors ultimately recommend the grayscale approach for its simplicity [8]. Another variant in the literature is to use the green channel only; this is the recommendation by [9]. Notice in equation (6), the green channel accounts for the majority of the information in the grayscale conversion, so it is not surprising that these two approaches yield similar performance.

Other researchers have proposed additional variants to the camera fingerprint extraction pipeline, usually in the choice of the denoising filter, F, in equation (3). Fridrich et al. originally suggested a wavelet-based denoiser [4]. Total variation [10], Wiener filter [11], BM3D [6], and other techniques have also been suggested in the literature. Ultimately, the choice of denoiser does not have a significant impact on the quality of the camera fingerprint—given that the denoiser is a "good" one that can remove noise while preserving fine details and edges in the image; a "bad" denoiser will oversmooth the image and effectively wipe out the subtle camera fingerprint.

The classic representation of a camera fingerprint is a matrix with dimensions matching those of the original image or images that were used to construct the fingerprint. Each entry in the matrix is a value between 0 and 1 that can be visualized as a gray value that varies between black (value 0) and white (value 1). An example is shown in figure 2. For storage in the computer, these values can be converted to integer gray values between 0 and 255, thus requiring eight bits for each pixel value. Note that, because the values are essentially randomly distributed, the camera fingerprints are not amenable to typical compression schemes such as Huffman encoding.

To reduce the memory needed for storing fingerprints on the computer, Bayram et al. [12] proposed a simple binarization scheme. They sharply truncate the fingerprint values to either 0 or 1 (black or white). Each fingerprint pixel value can then be stored with a single bit, representing eight times storage savings. Valsesia et al. [13] have suggested using random matrix projections to compress the fingerprints. In this case, the user can set the parameters of the projection to achieve the desired balance between compression and fingerprint quality.

A flowchart of the standard camera fingerprint extraction process is shown in figure 3.

Matching the fingerprint

A hypothesis testing framework is used to decide if a given image was taken with a particular digital camera:

$$H_0 : \mathbf{K}_1 \neq \mathbf{K}_2;$$
$$H_1 : \mathbf{K}_1 = \mathbf{K}_2. \qquad (7)$$

That is, given two fingerprints, say one from the reference camera and one from an image in question, the null hypothesis assumes the fingerprints are *not* equal; under the alternative hypothesis, the two fingerprints *do* match. Keep in mind that under this framework it is impossible to prove a negative—it is not possible to prove that two fingerprints are definitely from different cameras. The only two possible results from this test are: a) there is strong evidence that the fingerprints "match" or that the images were taken with the same camera, or b) the test is inconclusive; there is not enough evidence to say the fingerprints match.

The test statistic for camera fingerprint matching is based on the normalized cross-correlation (NCC):

$$\rho(s_1, s_2; \mathrm{X}, \mathrm{Y}) = \frac{\sum_{k=1}^{m}\sum_{l=1}^{n}\left(\mathrm{X}[k,l]-\bar{\mathrm{X}}\right)\left(\mathrm{Y}[k+s_1, l+s_2]-\bar{\mathrm{Y}}\right)}{\|\mathrm{X}-\bar{\mathrm{X}}\|\|\mathrm{Y}-\bar{\mathrm{Y}}\|}, \quad (8)$$

where **X** and **Y** are two camera fingerprint estimates and the coordinates $[s_1, s_2]$ denote horizontal and vertical circular shifts of **Y** against **X**, and || denotes the L_2 norm.

Then, denote the coordinates of the shift where the maximum of (8) occurs as $s_{peak}=[s_1, s_2]$. The peak-to-correlation energy (PCE) statistic is given by

$$PCE = \frac{\rho(\mathbf{s}_{peak}; \mathbf{X}, \mathbf{Y})^2}{\frac{1}{mn-|\mathcal{R}|}\sum_{\mathbf{s}, \mathbf{s}\notin\mathcal{R}} \rho(\mathbf{s}; \mathbf{X}, \mathbf{Y})^2}, \quad (9)$$

where *R* is a small region of the image around the peak (e.g., an 11×11 square of pixels). The authors in [7] performed a large-scale test with more than one million images to determine that a PCE score above the threshold of 60.0 indicates that two fingerprints likely match (can reject the null hypothesis of a nonmatch).

Charlton and Meixner developed a unique visualization to help users understand the PCE score [14]. As seen in equation (9), while the final PCE score can be reported as a single number, to find that maximum correlation value, we actually have to compute *all* the

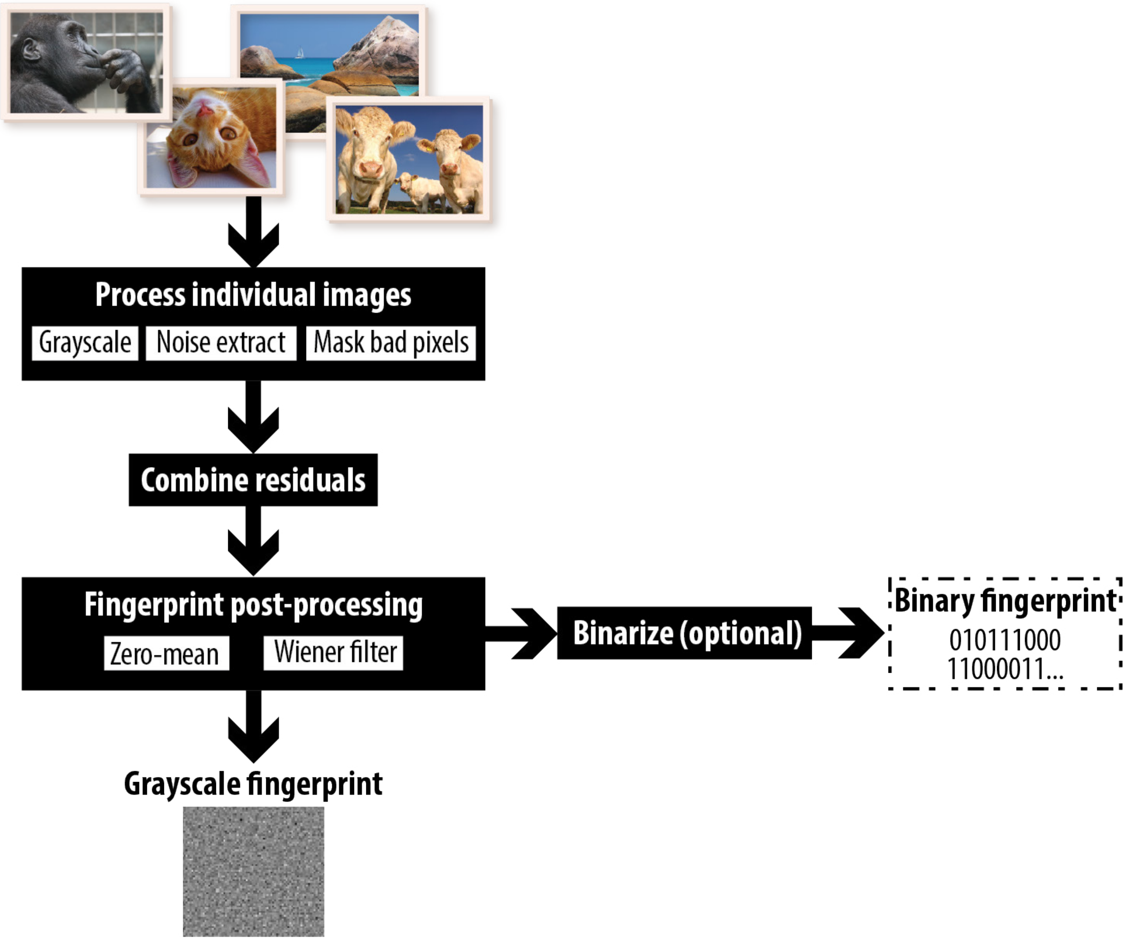


FIGURE 3. This flowchart depicts the camera fingerprint extraction pipeline. [Photo credits: Pixabay].

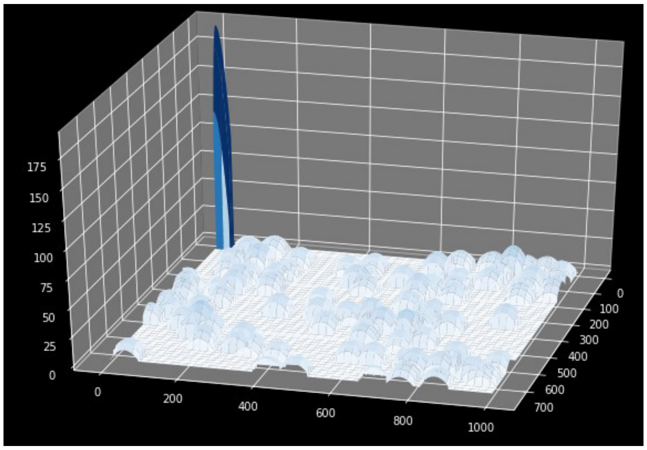


FIGURE 4. This visualization of a PCE score indicates strong evidence that the camera fingerprints match because there is one strong peak well above the matching threshold of 60 and all the remaining peaks are small.

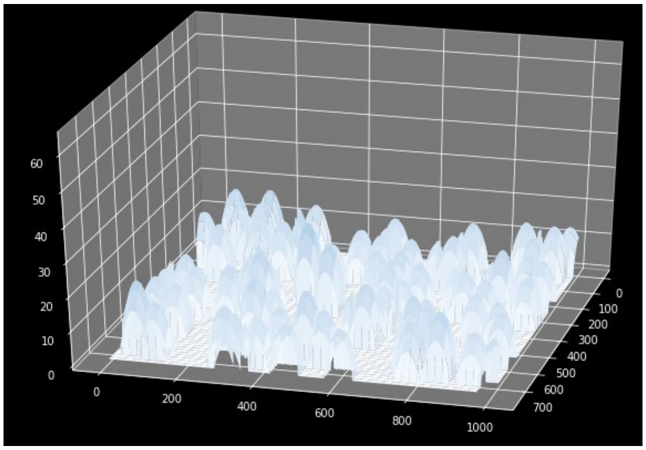


FIGURE 5. In contrast to figure 4, this visualization of a PCE score shows an inconclusive camera fingerprint match because there is no large peak and all correlation values are less than 60.

possible correlation values as the two fingerprints are shifted around to all the possible alignments.

Rather than throw that information away and only report a single number, Charlton and Meixner use the full correlation to create the visualization. In addition to the top correlation peak, the *N* next highest peaks are also identified, where the user can select a value for *N*. These correlation peaks are then plotted as a three-dimensional surface such as in figures 4 and 5. If there is strong evidence that the fingerprints match, we should see a plot with one strong peak well above the matching threshold of 60 and all the remaining peaks should be small; this is illustrated in figure 4, where the large peak is greater than 175. In contrast, if the camera match is inconclusive, there will not be a large peak and all correlation values will be less than 60, as is illustrated in figure 5.

Best practices

The most common question that users have when employing camera fingerprinting technology is: how many images do I need?

The original recommendation by the Binghamton researchers was to use a mass of 30 to 50 images when constructing a camera fingerprint [4]. However, newer work has suggested that there is a pattern of diminishing returns for fingerprint performance versus the mass of the fingerprint [8, 15]. Here, we mean “performance” in the sense of how likely a fingerprint is to correctly match other fingerprints from the same camera. Figure 6 illustrates these diminishing returns. The horizontal blue dotted line indicates the PCE score-matching threshold equal to 60. Each of the other lines represents one test image scored against a matching fingerprint of increasing mass. Clearly, 30–50 images is overkill, as the images tend to get a matching score with mass much lower than 30.

If not 30 images, then how many? Mahdian et al. propose a stopping criterion for building camera fingerprints [15]. The authors show that the Laplace distribution, shown in equation (10), is a good fit to the camera fingerprint pixel values.

$$f(x \mid \mu, b) = \frac{1}{2b} \exp\left(\frac{-|x-\mu|}{b}\right) \tag{10}$$

Also, as more and more images are incorporated into the fingerprint, the Laplace scale parameter, *b*, decreases and then plateaus. The authors propose to monitor the change in parameter *b*, and stop fingerprint construction when the Δ*b* falls below some threshold, say 0.05. An example of this Laplace convergence is shown in figure 7.

Users should also consider the content of their images before applying camera fingerprinting techniques. Very dark images (e.g., nighttime or lots of shadows) or very bright images (e.g., lights or sun glare) should not be used because the PRNU is not well expressed in those images. Among images that have good lighting, those with smoother content should be preferred over scenes with lots of texture.

Return to CONTENTS

For this reason, many of the benchmark data sets for camera forensics provide both "flat-field" and "natural" images. Researchers can use the data sets to test algorithms under ideal conditions (flat images) as well as more "real-world" conditions. An example of flat-field and natural images from the Vision data set [17] are shown in figure 8 (on the following page).

Surprising robustness

Camera identification with PRNU fingerprints is very reliable. The technology has high accuracy for matching fingerprints, and perhaps more importantly, has very low false alarm rates for erroneously linking images that did not come from the same camera. Binghamton researchers showed 99.7% true detection at a false alarm rate of 10^{-5} [5]. In a larger study with over one million images, they show true detection at 97.6% at a false alarm rate of 10^{-6} [7].

The PRNU signature is a good choice for a camera fingerprint because it is surprisingly robust. The PRNU survives the conversion from the analog signal on the CCD/CMOS chip to the digital signal on the camera's software. The PRNU persists in the resultant image even after processing by all the algorithms on the camera, such as white balance and color correction [5]. Most cameras will store images as JPEG files. Now, JPEG is a lossy compression scheme, but recent work has shown that the PRNU can survive at JPEG quality settings as low as 60% [18].

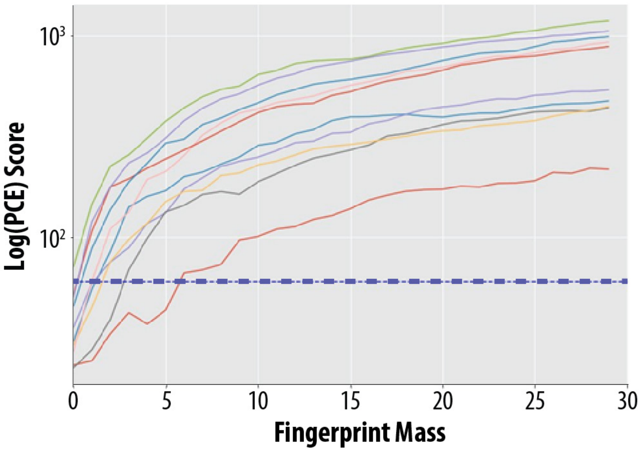


FIGURE 6. In this plot of PCE score versus fingerprint mass for an example camera, the horizontal blue dashed line indicates the PCE score-matching threshold equal to 60. Each of the other lines represents one test image scored against the matching fingerprint of increasing mass. Images are from the Dresden image database, an open-source data set for benchmarking image forensics algorithms [16].

Matching images that have been edited is also fairly straightforward. Simple geometric edits like rotation

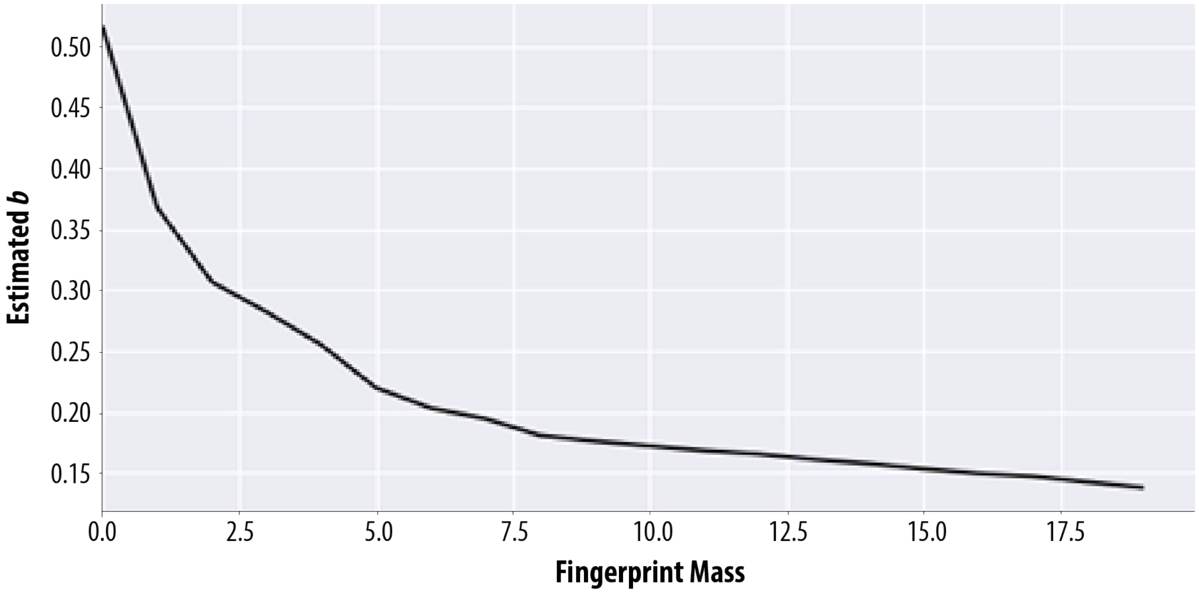


FIGURE 7. As more and more images are incorporated into a camera fingerprint, the Laplace scale parameter, *b,* decreases and then plateaus—this is an example of Laplace parameter convergence.

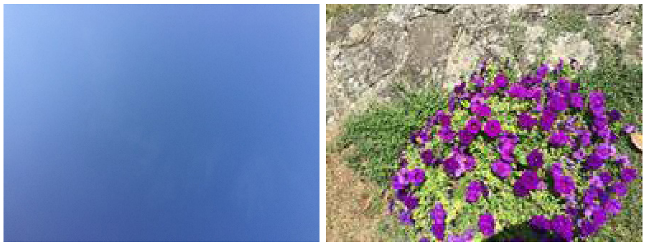


FIGURE 8. This flat-field image (left) and natural image (right) are from the VISION Dataset for benchmarking camera forensics algorithms. [Photos: Copyright (c) CSP Lab. (Communications & Signal Processing Laboratory), Dept Information Engineering, University of Florence, 2017; more details at [17]].

and cropping are easy to account for. The PCE score will actually check all possible alignments and, thus, detect the match if an image has been cropped. For rotations, simply calculate the PCE score for all four possible orientations. If images have been resized, fingerprinting is a little more difficult, but possible. In this case, a brute force approach is to iterate through several resize parameters in order to find the case where the image matches the fingerprint [19]. Alternatively, the fingerprint images and the query images can both be resized to common dimensions and then matched [9]. However, when images are drastically downsized compared to their original size, the fingerprint pattern in the images will often be wiped out.

Extension to video

PRNU camera fingerprinting also works for videos. In the simplest case, the frames of a video are extracted as still images and the standard PRNU extraction and comparison methods which we outlined in the previous sections can be applied to the sequence of extracted images. However, digital video encodings are much more complex than digital images, which makes it harder to detect the fingerprint in the extracted video frames.

For example in the H.264 video codec, there are three types of frame encodings: I, P, and B. Generally, I-frames are self-contained in the sense that the pixels in that frame are encoded relative only to the other pixels in the frame. P-frame pixels are encoded relative to that frame and a previous frame, and B-frame pixels are encoded relative to that frame, one prior, and one subsequent frame. In a series of experiments, Deka et al. determined that for PRNU fingerprinting, I-frames are the best. However, there are much fewer I-frames in a video than P-frames, so for short videos, it may be difficult to get a good PRNU estimate.

More work is needed, perhaps combining the approaches of Deka et al. [9] with the stopping condition work of Mahdian et al. [15], in order to make recommendations for frame types and frame mass for fingerprinting videos.

Camera make and model classification

Camera fingerprinting based on the PRNU noise can only give us information about whether images or videos were captured with the same device; this method does not actually tell us anything about the camera itself. For example, calculating the PRNU fingerprint from some images or videos cannot tell us whether they were taken with say, a Canon PowerShot, or an iPhone 8, or any other make and model of camera. Now, if that make/model information is already known, it is smart to incorporate that information into the overall camera fingerprinting analysis.

There are other approaches in the digital forensics literature that can be used to identify camera make and model. PRNU-based identification can be loosely compared to hardware reverse engineering, as we are trying to identify the physical sensor. In this way, camera make and model identification can be thought of more like software reverse engineering. Different camera models employ different chains of image processing algorithms to transform the light signal into a digital image. Color demosaicing, gain correction, white balance, JPEG quantization, and others: these algorithms leave traces on the resultant images. As with the PRNU noise, these software traces typically do not affect the visual quality of the images to the degree that one can perceive by eye.

Researcher Matthew Stamm from Drexel University and his students have developed an approach to make and model classification using deep learning frameworks. Traditionally, applications of convolutional neural networks (CNNs) to imaging problems have targeted learning the semantic content of the image. (E.g., is this a picture of a "cat" or a "teacup"?) The Drexel University researchers used a constrained CNN to extract what they call "deep forensic features" of the images. By looking at different specific types of image patches, such as smooth content or sharp horizontal edges, the network is trained to recognize the traces of the onboard camera algorithms [20, 21].

Extension to forgery detection

Besides applications of camera identification, the image's PRNU fingerprint can also be used to detect whether any regions of the image have been manipulated or forged. An example manipulation might be an object or person removed from the scene in the image. Instead of correlating the entire image to its camera fingerprint, as is done for camera identification, the idea for forgery detection is to check smaller subsections of the image. If a subsection of the image does not correlate strongly to the corresponding subsection in the camera fingerprint, this may be evidence of manipulation in that subregion of the image.

The Binghamton research team proposed a patch-based detector, where the image is divided into subregions of overlapping rectangles of various sizes [22]. The correlation between the image and fingerprint is calculated for each region, for each rectangle size. The results are aggregated and regions of the image with low correlation are identified as possibly manipulated.

A group from University of Naples proposed a slightly different approach where, rather than arbitrary patches, the image is subdivided into semantically similar super-pixel regions [23]. The super-pixels are correlated to the corresponding regions of the camera fingerprint, and again, areas of low correlation indicate possible manipulation. These two approaches assume that one has a good quality camera fingerprint from the same camera as the suspicious image. This assumption greatly restricts the situations where PRNU-based manipulation detection can be utilized.

What about synthetic images? The PRNU approach also assumes the images in question came from a physical camera. What would happen if you tried to fingerprint a "fake" image? Marra et al. investigated this question for completely synthetic images that were created using a generative adversarial network (GAN) [6]. They found that different synthetic images produced with different GAN networks actually do have a fingerprint, and they can determine which GAN produced which images. However, without a reference fingerprint for all possible GANs in the universe, this method is not practical for a general "real versus GAN" image detector.

Conclusion

The photo-response non-uniformity, or PRNU, is a pattern imparted on all images or videos captured with a digital camera. Although the pattern is subtle—one cannot notice it by eye—it is unique to each light-sensing chip inside the camera and is stable over time. Images or videos can be processed with special algorithms to extract and enhance the PRNU signature. The PRNU acts as a "fingerprint" and allows images or videos to be matched to their specific camera.

The first paper on PRNU forensics was published in 2005 [3]. In the past 15 years, the process for PRNU fingerprinting has been well established. Best practices have been established for the PRNU extraction and matching algorithms [9, 15, 14], and camera fingerprint matching is extremely accurate [9]. Newer research has focused on applying PRNU techniques to detect image forgeries [22, 23] or synthetics images [6].

References

[1] Omnicore. "Facebook by the numbers: stats, demographics & fun facts." Available at: https://www.omnicore-agency.com/facebook-statistics/. [Accessed 27 Jul 2020].

[2] Geradts ZJ, Bijhold J, Kieft M, Kurosawa K, Kuroki K, Saitoh N. "Methods for identification of images acquired with digital cameras." In: *Proceedings SPIE 4232, Enabling Technologies for Law Enforcement and Security;* 2001 Feb 21. Available at: https://doi.org/10.1117/12.417569.

[3] Lukáš J, Fridrich J, and Goljan M. "Determining digital sensor origin using sensor imperfections." *Proceedings of SPIE-The International Society for Optical Engineering.* 2005;5685:249–260. doi: 10.1117/12.587105.

[4] Lukáš J, Fridrich J, Goljan M. "Digital camera identification from sensor pattern noise." *IEEE Transactions on Information Forensics and Security.* 2006;1(2):205–214.

[5] Fridrich J. "Digital image forensics using sensor noise." *IEEE Signal Processing Magazine.* 2009;26(2):26–37.

[6] Marra F, Gragnaniello D, Verdoliva L, Poggi G. "Do GANs leave artificial fingerprints?" 2018. Cornell University Library, arXiv:1812.11842.

[7] Fridrich J, Goljan M, Filler T. "Large scale test of sensor fingerprint camera identification." *Proceedings of SPIE-The International Society for Optical Engineering.* 2009; 7254. doi: 10.1117/12.805701.

[8] Gisolf F, Barens P, Snel E, Malgoezar A, Vos M, Mieremet A, Geradts Z. "Common source identification of images in a large database." *Forensic Science International.* 2014;244:222–230.

[9] Deka R, Galdi C, Dugelay J. Hybrid G-PRNU: "A novel scale-invariant approach for asymmetric PRNU matching." Society for Imaging Science and Technology. 2019. Published version of conference presentation available at: https://doi.org/10.2352/ISSN.2470-1173.2019.5.MWSF-546.

[10] Gisolf F, Malgoezar A, Baar T, Geradts Z. "Improving source camera identification using a simplified total variation based noise removal algorithm." *Digital Investigation.* 2013;10:207–214.

[11] McCloskey S. "Confidence weighting for sensor fingerprints." In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops;* 2008 Jul. doi: 10.1109/CVPRW.2008.4562986.

[12] Bayram SC, Sencar HT, Memon N. "Efficient sensor fingerprint matching through fingerprint binarization." *IEEE Transactions on Information Forensics and Security.* 2012;7(4):1404–1413.

[13] Valsesia D, Coluccia G, Bianchi T, Magli E. "Compressed fingerprint matching and camera identification via random projections." *IEEE Transactions on Information Forensics and Security.* 2015;10(7):1472–1485. doi: 10.119/TIFS.2015.2415461.

[14] Charlton ST, Meixner KJ. Method of comparing a camera fingerprint and a query fingerprint. US Patent US10235765B1. March 2019. Details available at: https://patents.google.com/patent/US10235765B1/.

[15] Mahdian B, Novozámský A, Saic S. "Determination of stop-criterion for incremental methods constructing camera sensor fingerprint." In: Shi YQ, Kim H., Perez-Gonzalez F, Yang CN, editors. *Digital-Forensics and Watermarking.* IWDW 2014. Lecture Notes in Computer Science. Springer International, 2015. Pp. 47–59.

[16] Gloe T, Böhme R. "The 'Dresden Image Database' for Benchmarking Digital Image Forensics." *Journal of Digital Forensic Practice.* 2010;3(2–4):1584–1590. doi: 10.1080/15567281.2010.531500.

[17] Shullani D, Fontani M, Iulliani M, Al-Shaya O, Piva A. "VISION: A video and image dataset for source identification." *EURASIP Journal on Information Security.* 2017;15(10). Article available at: https://link.springer.com/article/10.1186/s13635-017-0067-2. VISION Dataset available at: https://lesc.dinfo.unifi.it/en/datasets.

[18] Goljan M, Chen M, Comesanã P, Fridrich J. "Effect of compression on sensor-fingerprint camera identification." *Electronic Imaging.* 2016:1–10. doi: 10.2352/ISSN.2470-1173.2016.8.MWSF-086.

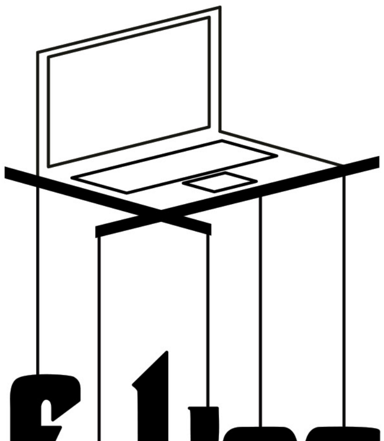
[19] Goljan M, Fridrich J. "Camera identification from cropped and scaled images." In: *Proceedings of SPIE 6819, Security, Forensics, Steganography, and Watermarking of Multimedia Contents X,* 68190E; 2008 March 18. doi: 10.1117/12.766732.

[20] Bayar B, Stamm MC. "Towards open set camera model identification using a deep learning framework." In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing;* 2018 April 15–20; Calgary, AB, Canada: pp. 2007–2011. doi: 10.1109/ICASSP.2018.8462383.

[21] Hosler B, Mayer O, Bayar B, Zhao X, Chen C, Shackleford JA, Stamm MC. "A video camera model identification system using deep learning and fusion." In: *2019 IEEE International Conference on Acoustics, Speech and Signal Processing;* 2019 May 12–17; Brighton, United Kingdom: pp. 8271–8275. doi: 10.1109/ICASSP.2019.8682608.

[22] Lukáš J, Fridrich J, Goljan M. "Detecting digital image forgeries using sensor pattern noise." In: *Proceedings of SPIE 6072, Security, Steganography, and Watermarking of Multimedia Contents VIII,* 60720Y; 2006 February 16; San Jose, CA: pp. 2362–372, San Jose, CA. doi: 10.1117/12.640109.

[23] Chierchia G, Parrilli S, Poggi G, Verdoliva L, Sansone C. "PRNU-based detection of small-size image forgeries." In: *2011 17th International Conference on Digital Signal Processing;* 2011 July 6–8; Corfu, Greece: pp. 1–7. doi: 10.1109/ICDSP.2011.6004957.



[Photo credit: iStock.com/baona, THPStock]

Deepfakes: Is a Picture Worth a Thousand Lies?

Candice Gerstner, Emily Phillips, Larry Lin

Are we getting closer to a time when we can’t trust what we see or hear? It is a scary thought that forensic analysts have been pondering for years. Since their invention, multimedia (e.g., images, videos, audio, and text) have been used as a source of truth. Yet, with an increase in our digital footprint and the advent of new technology, this assurance is slipping away.

In particular, access to portable digital cameras and services to distribute multimedia has increased the volume of media seen on the internet. While doctoring the content of media, such as images, has been done throughout the history of photography [1], the availability of software such as Adobe Photoshop, Gimp, and Corel Painter make it easy for anyone to manipulate media. However, making a sophisticated fake using these software packages can take days. Recent advances in computational power and deep learning have made it not only quicker to create fake media, but also less expensive to mass produce these fakes. Many deep learning-based algorithms are already available on open-source repositories like GitHub. These ready-to-use repositories pose a threat to national security in that the application of these technologies require no more than a personal laptop and a minimal amount of technical skill.

Our ability to authenticate media has become more crucial than ever before. Humans have a visceral reaction to media and form beliefs quickly [2]; thus, it is essential for people to verify sources and be vigilant in authenticating what their senses are telling them is the truth. Members of Congress too have recognized the danger of the "technology undermin[ing] public trust in recorded images and videos as objective depictions of reality" [3].

Introduction to manipulations and authentication

In this article, we focus on images and videos, although many of the same concepts can be extended to other multimedia types. The American Society for Testing and Materials (ASTM) defines authentication as "the process of substantiating that data are accurate representations of what they are purported to be" [4]. This article will utilize that definition and extend it to detail the science of how media is processed and the specific criteria that will be used to make the determination of authenticity. The criteria will represent different levels of manipulation (or doctoring), where ***manipulation*** will be defined as the modification of either the "structure" and/or the "content" of the media.

Structure of a file includes, but is not limited to, the name, hash, format, exchangeable image file (EXIF) data, quantization tables, and hex view of the data. *Content* of a file includes, but is not limited to, textures, shading and shadows, color balance, lighting,

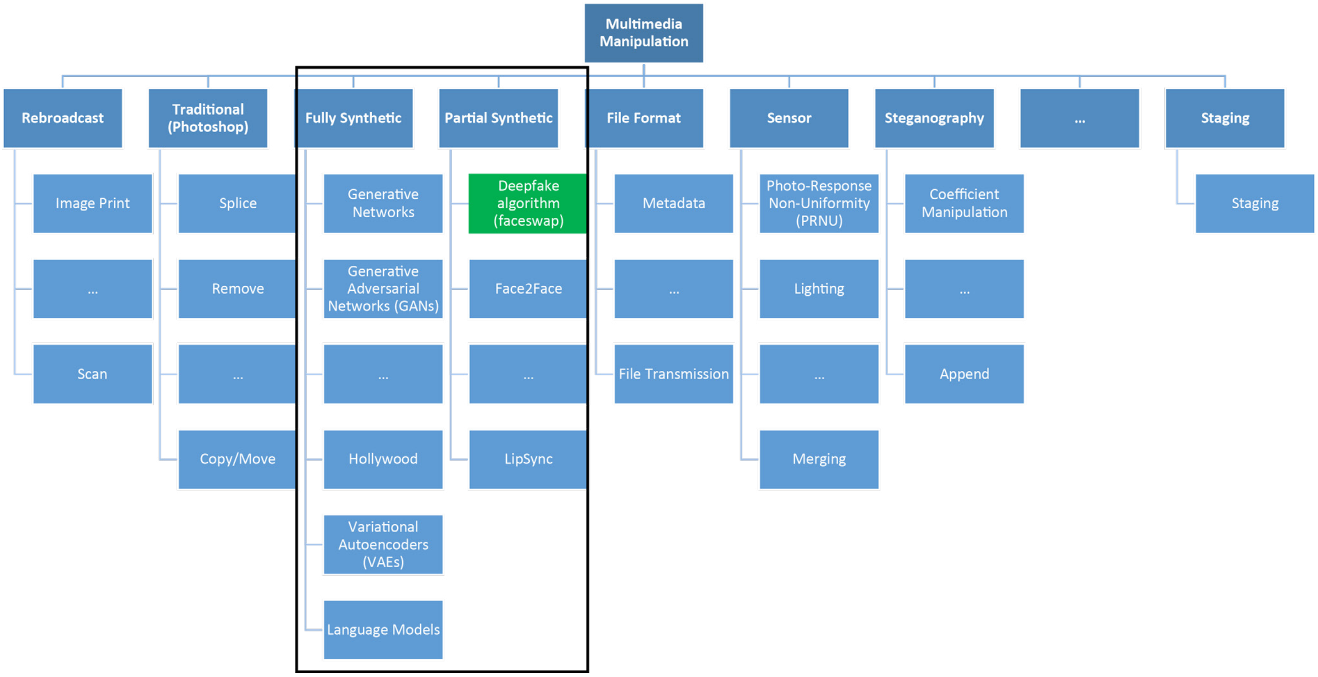


FIGURE 1. The landscape of possible multimedia manipulations, diagrammed in this chart, is so diverse that authentication is an extremely difficult task. The box around Fully Synthetic and Partial Synthetic represents the categories of artificial intelligence (AI) generated synthetic manipulations referred to as ***deepfakes.*** For clarification, the Hollywood box under Fully Synthetic is meant to categorize AI techniques used in Hollywood. Note that Hollywood also uses non-AI techniques to generate synthetic media.

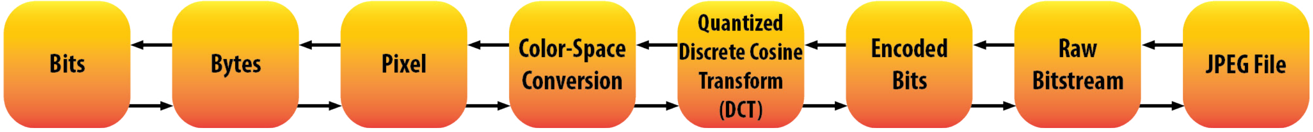


FIGURE 2. Manipulations can occur at all levels of JPEG creation. This diagram introduces these levels and the terminology that will be used to describe them.

and composition. The landscape of possible manipulations is so diverse that authentication is an extremely difficult task (see figure 1). Categorizing manipulations complicates things further if one considers intent or perspective (e.g., subjectiveness or threshold).

In this article, we will address two categories of manipulations, namely nonsynthetic (commonly referred to as traditional) and synthetic via artificial intelligence (AI; referred to as deepfakes). Most nonsynthetic manipulations (i.e., to structure or content) result from modifying some component(s) of the media compression algorithm.[a] For example, in a JPEG image, a simple steganography method adds the hex values of another (hidden) image after the original image in the EXIF data. This is done by locating the end of a file tag for the image and adding the information after. This is a structural manipulation and can also be referred to as a byte manipulation, according to the diagram in figure 2. Even something as simple as changing the quantization tables used in JPEG compression is considered a manipulation.

The manipulations that are receiving most of the media attention lately are those referred to as ***deepfakes.*** This term has been incorrectly used in many cases as a catch-all for any media manipulation. We will now clarify the origin of this term and state the definition that we will use in this article. The term was first seen on the site www.reddit.com, where a user posted an example of an autoencoder (i.e., a deep learning method) used to perform a face swap (the post was later removed). In figure 1, this is referred to as the ***deepfake algorithm.*** The term comes from a blending of the words "deep learning" and "fake media". We now use the term deepfakes to refer broadly to multimedia files that have been created (i.e., fully synthetic) or edited (i.e., partially synthetic) using some form of AI technology.

These deep learning-based algorithms have given experts, and in some cases, mere enthusiasts, the ability to manipulate multimedia in such a way that even the most keen observers can be deceived. Furthermore, ***fully synthetic media*** such as those generated from deep learning algorithms, often cannot be detected with traditional authentication and forensic techniques that were designed to detect manipulations like those done with Photoshop.

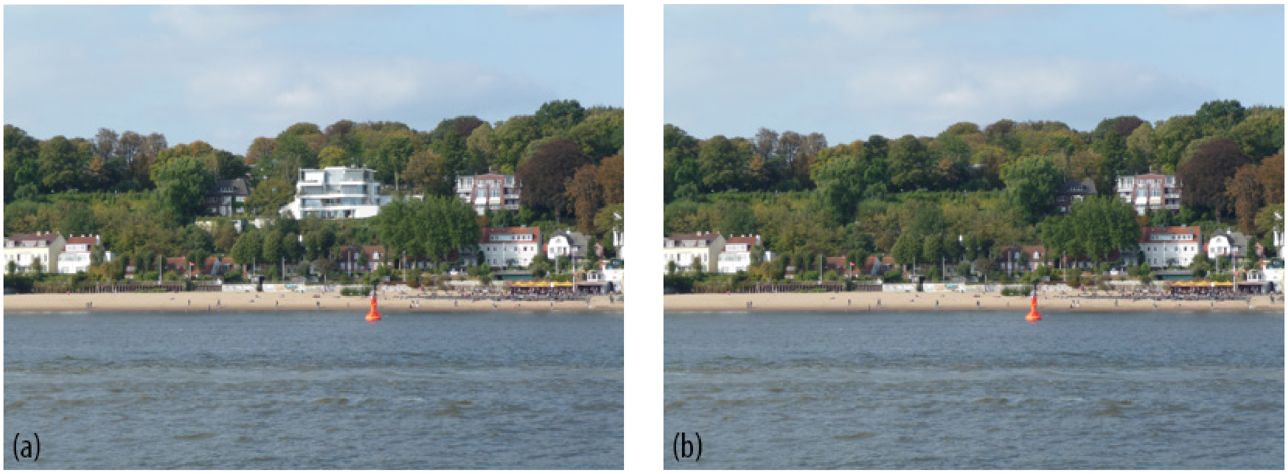


FIGURE 3. Image (a) is the original, unmanipulated image from [5]. Image (b) is the manipulated image with buildings removed from the scene [5].

[a] A few exceptions are rebroadcasting and staging.

Traditional manipulation techniques

For purposes of this article, the focus will be on *malicious manipulations to images*[b] that alter the content of the media. Below we describe a few manipulations that can be performed using software such as Adobe Photoshop, Gimp, and Corel Painter.

1. **Removals** are the removal of an object which is then replaced by content derived from another region within the same media. This is typically done in such a way as to remove the notion that the object was present (see figure 3 on previous page).
2. **Splice** is the insertion of an object from one image (called the donor image) into the probe image (see figure 4).
3. **Copy-clone** is the insertion of an object from an image into a different location within the same image (see figure 5).

These traditional techniques are still being used as a means of manipulation. However, the creation of sophisticated fakes requires time and experience.

Deep learning-based techniques

Traditional manipulations are becoming less manual and in some cases harder to visually identify. Several applications are available that make manipulating media extremely easy, including Snapchat and TikTok. In some cases, these manipulations use AI, such as deep learning. Programs like Photoshop have started to incorporate AI into their products. Adobe Sensei is an example of this. Sensei has the capability to automatically tag content, determine object edges, remove objects using AI-based inpainting techniques, and summarize documents [7]. Improved computational resources and access to larger data sets has given rise to deep learning approaches to produce manipulated media.

Generative models

The primary classes of algorithms used to generate fully synthetic media are generative adversarial networks (GANs), variational autoencoders (VAEs), and language models (applied to audio and text). GANs are generative models that approximate probability distributions. In particular, GANs are implicit density

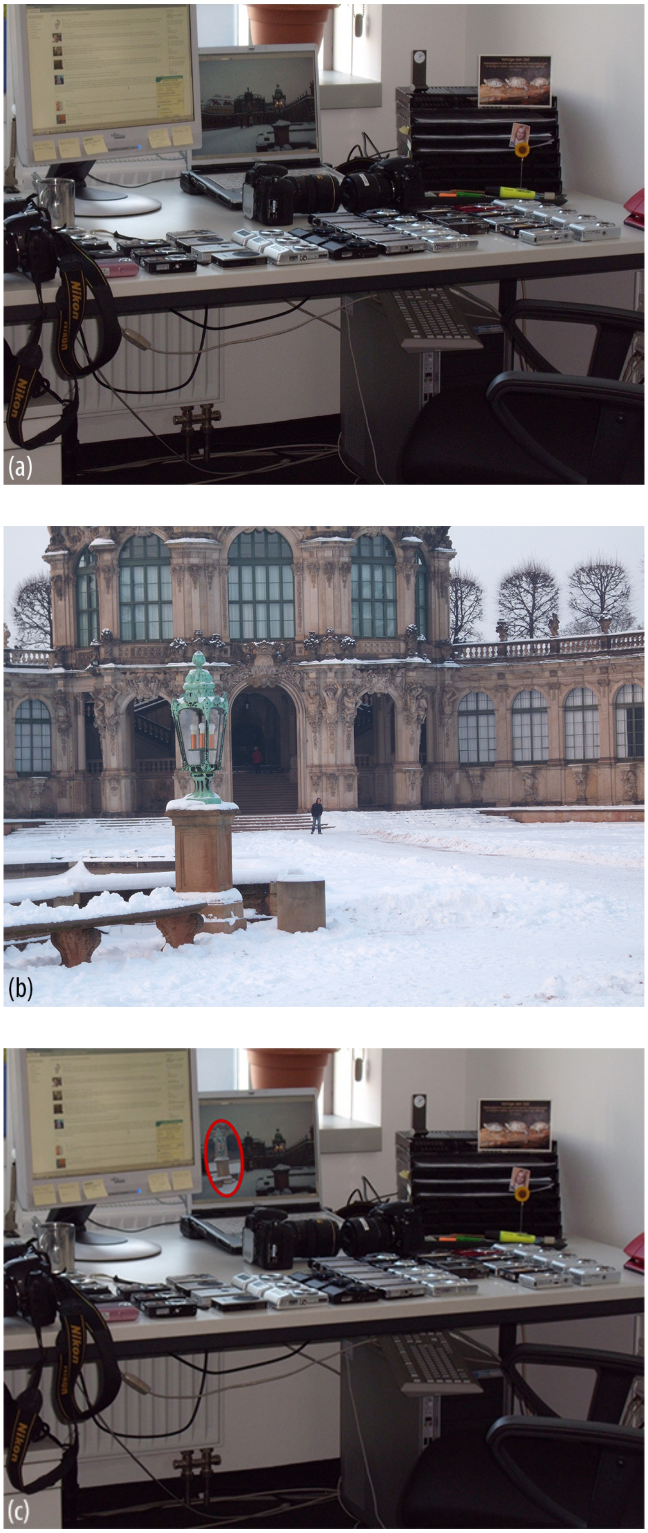


FIGURE 4. Image (a) is the probe image from [6]. Image (b) is the donor image from [6]. Image (c) is the spliced image obtained by inserting part of image (b) into image (a).

[b] For reference, some examples of benign content manipulation include lighting, contrast changes, or computer animations.

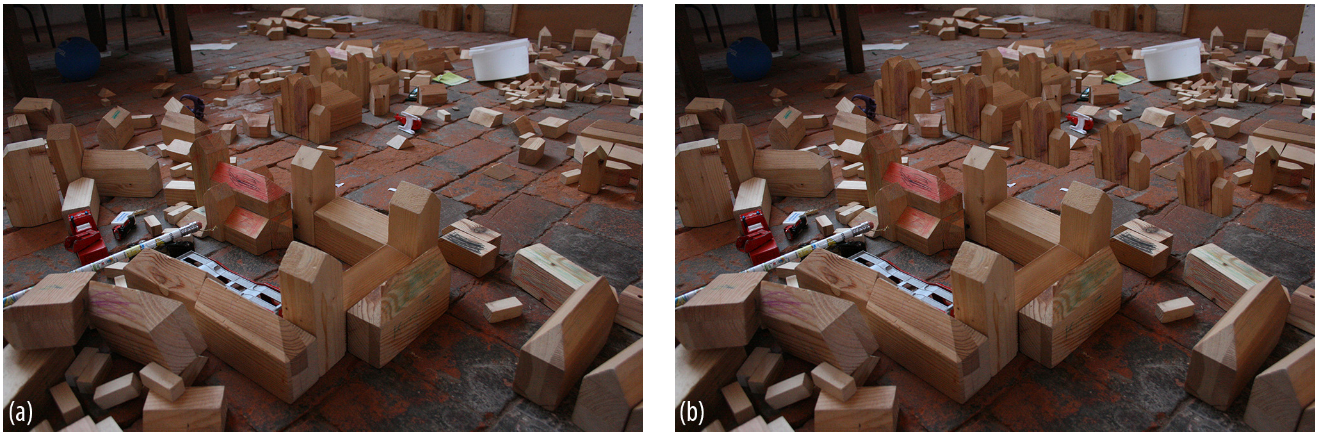


FIGURE 5. Image (a) is the original image from [5]. Image (b) is the manipulated image with a block structure copied and cloned several times in the scene [5].

models that do not explicitly approximate the probability density function. Despite this, they have a few advantages: the model can be sampled from and, with sufficiently large capacity, can approximate complex distributions. As can be seen in figure 6, GANs typically are made up of two neural networks, a generator and a discriminator. The generator produces images, for example, by mapping samples from a known distribution (e.g., multivariate independent Gaussian) and trains in an adversarial way against a discriminator, that attempts to learn whether an image presented to it is from the training data set (i.e., real) or produced from the generator (i.e., fake). There is more technical information on how a GAN works in the next subsection.

VAEs [8] parametrize a mapping to a (lower dimensional) latent space, via an encoder, and back to the data space, via a decoder. Efficient inference is achieved for the latent variables and samples using the learned encoder and decoder, respectively. The model parameters are optimized using a variational lower bound on the mutual information between the encoded distribution and a prior on the latent space (typically a multivariate Gaussian). Qualitatively, a VAE can be thought of as learning a compression and decompression of the data.

The previous algorithms can be used to generate synthetic content for all media types. One type of algorithm that is ubiquitous in modeling audio [9] and text [10] is a language model. Language models approximate the conditional distributions terms that decompose the joint distribution of the data via the

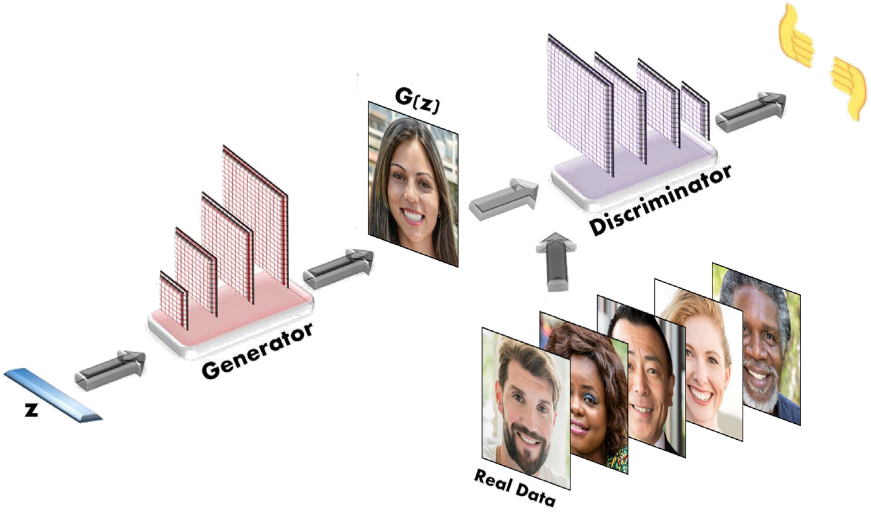


FIGURE 6. In this GAN setup, a random vector is fed through the generator to produce an image. Images produced from the generator along with real images from the training set are presented to the discriminator, which decides if each image is real (i.e., from the training set) or fake (i.e., produced by the generator). [Photos credits: iStock.com/JohnnyGreig]

chain rule for probability. These models are trained via supervised learning on a corpus of data to predict future samples from previous ones.

Fully synthetic media: GANs

GANs are learned via an information-theoretic min-max loss function first introduced in [11]. We start by introducing our notation. The random variable Z is sampled from the multivariate normal distribution $N(0,I_N)$: $Z\sim N(0,I_N)$, I_N= *NxN identity matrix,* and let $X\sim P_{data}$, where P_{data} denotes the distribution of data. Then let $\hat{G}_\theta$: $Z \rightarrow X$ be a parametric function with parameters θ producing samples with distribution $P_{\hat{G}_\theta}$. We call this model the *generator*. The goal of the generator is to map the distribution P_z to P_{data} so that the model attempts to produce samples from the latter. The *discriminator*, which we define as

$$D(x) := \begin{cases} 1, x \sim P_{data} \\ 0, x \sim P_{\hat{G}_\theta} \end{cases}, \qquad (1)$$

is used as a substitute for labels so that learning can be done in an unsupervised fashion. To learn D in practice, we use: $\hat{D}_\phi$: $X \rightarrow [0,1]$, a parametric approximation of D. It is common to say that $D(x)$=1 if x is *real* and 0 if x is fake (generated by $\hat{G}_\theta$). Mathematically, the discriminator can be interpreted as $D(x)=Prob(x \text{ is real})$.

In the original GAN framework [11], parameters for $\hat{G}_\theta$, $\hat{D}_\phi$ are trained via a min-max adversarial game with binary cross entropy loss for $\hat{D}_\phi$:

$$\mathcal{L}_{\text{GAN}}(\hat{G}_\theta, \hat{D}_\phi) := \min_\theta \max_\phi \mathbf{E}_{\mathrm{X}\sim \mathbf{P}_{\text{data}}} \left[\log\left(\hat{D}_\phi(x)\right)\right] + \mathbf{E}_{\mathrm{Z}\sim \mathbf{P}_{\mathrm{z}}} \left[\log\left(1 - \hat{D}_\phi\left(\hat{G}_\theta(z)\right)\right)\right], \quad (2)$$

where $\mathbf{E}_{x\sim P}$ $[f(x)]$ denotes the empirical estimate of expectation:

$$\frac{1}{N} \sum_{\substack{i=1,\dots N \\ x_i \sim P_X}} f(x_i). \qquad (3)$$

In practice, $\hat{G}_\theta$ and $\hat{D}_\phi$ are parametrized by deep neural networks and the optimization depicted in the above loss equation is performed in two steps:

$$\min_\theta \mathbf{E}_{\mathrm{Z}\sim \mathbf{P}_{\mathrm{z}}} \left[\log\left(1 - \hat{D}_\phi\left(\hat{G}_\theta(z)\right)\right)\right] \qquad (4)$$

$$\max_\phi \mathbf{E}_{\mathrm{X}\sim \mathbf{P}_{\text{data}}} \left[\log\left(\hat{D}_\phi(x)\right)\right] + \mathbf{E}_{\mathrm{Z}\sim \mathbf{P}_{\mathrm{z}}} \left[\log\left(1 - \hat{D}_\phi\left(\hat{G}_\theta(z)\right)\right)\right] \quad (5)$$

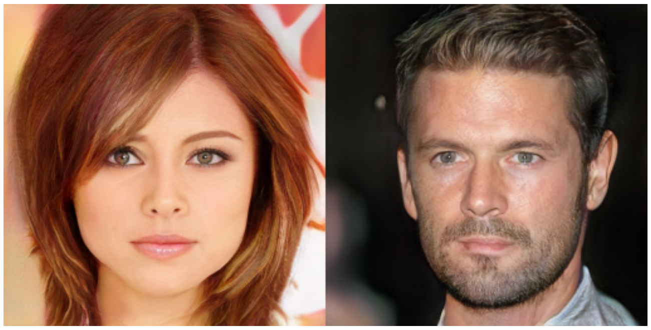


FIGURE 7. These sample images generated from HQGAN illustrate combined deep learning techniques from [14].

Improvements, such as using the Wasserstein loss or hinge loss, to this optimization have led to better trained networks [12]. Methods to improve the stability of learning and quality of generated samples include function approximators with improved gradient flow for optimization (e.g., residual connections with batch normalization [13], or progressive layer-wise learning of parameters [14]). An example of these combined techniques from [14] can be seen in their HQGAN-generated images in figure 7. The current state-of-the-art face algorithm is StyleGAN 2.0 [15], which facilitates improved style control of the generated image such as hair color, facial structure, glasses, etc. Some StyleGAN 2.0-generated images can be seen in figure 8.

Partially synthetic media: Face swap algorithm

Face swapping is a manipulation technique that takes an original video (or image) of person A and replaces the face with that of person B. This has gained immense popularity, due to communities of enthusiasts maintaining GitHub repositories with easy-to-use interfaces to apply the algorithm [16]. One important note is that, while GANs can be used to generate the new face, in practice, almost all high-quality face swaps are created using autoencoders. GANs try to learn a mapping between samples of different distributions, while autoencoders learn to compress and

Return to CONTENTS

FIGURE 8. These sample images generated from StyleGAN 2.0 [15] illustrate the state-of-the-art face algorithm which facilitates improved style control of the generated image such as hair color, facial structure, glasses, etc.

decompress between the data manifold and a lower-dimensional manifold. An autoencoder is a simpler statistical model than the previously mentioned VAE. Here, instead of an assumed prior on the latent space, the encoder directly maps the input data manifold into the latent space, while the decoder attempts to learn the inverse mapping to reproduce the input.

Nontrivial preprocessing is necessary to generate good data sets to train the autoencoder. To learn the face swap model using an autoencoder, preprocessed samples of person A and person B are mapped to the same intermediate compressed latent space using the same (i.e., learned) encoder parameters. However, person A and person B have different decoders (i.e., using different parameters) mapping from the shared latent space representation to their respective original inputs. Once the three networks are trained, to swap the face of person B onto A, the target video (or image) of A is fed frame by frame into the common encoder network, and then decoded by person B's decoder network. In other words, person A's face is taken into the shared latent space and decoded with person B's face (see figure 9). As a result, facial expressions and mouth shape can be preserved. Semiautomated (i.e., with human selection) postprocessing blends the AI-generated face into the original image.

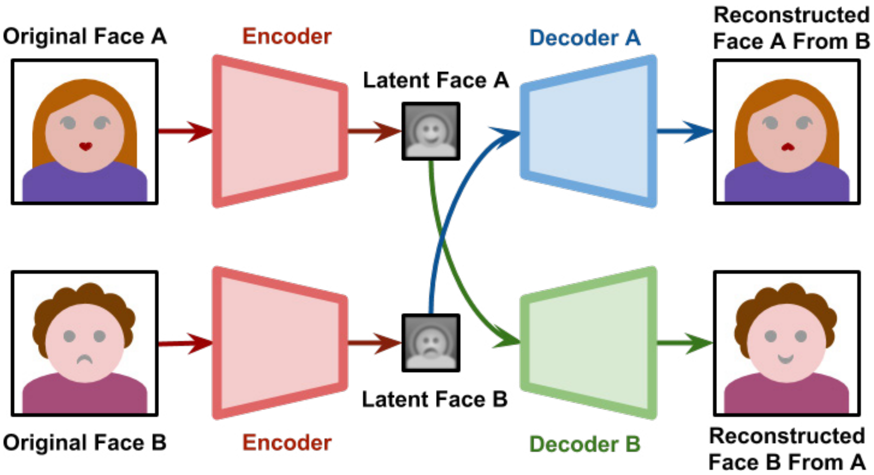


FIGURE 9. This face-swap breakdown from [17] shows an original frame of a video of Face A going into the Encoder A, and encoded in the smaller space as a smiling face. The Latent Face A is decoded using Decoder B, producing a frame.

Authentication of media

Given the diversity in manipulation type, developing a one-size fits all technique for multimedia authentication is impossible. For years, forensic analysts have focused their attention on specific types of manipulations in an attempt to develop robust methods of detection for each manipulation. Although most forensic techniques produce quantitative information (e.g., heat maps), there is typically no corresponding explanatory information to help the examiner understand how the information was obtained. Understanding the underlying mathematics and media creation properties of multimedia files and the techniques to examine them is extremely important for justifying conclusions made from forensic evidence. One of the primary shared objectives between the DARPA MediFor (media forensics) program [18] and the researchers at the Department of Defense is to explore the technical characteristics of multimedia manipulation. Using standard forensic best practices along with the technical characteristics allows them to develop more powerful and meaningful detection techniques.

Best practices

We can all do our part in reducing the spread of manipulated media. Whether you are a casual observer or a scientist, awareness is one of the most important tools in this realm. It is good practice to check the source of the media before drawing conclusions. In addition, reverse image searches, like TinEye, can be extremely useful if the media is a composition of images.

For the scientist, multimedia authentication, like any forensic science, requires adhering to some basic principles. Techniques used for an analysis must be accurate (i.e., consistent measurements within an experiment), precise (i.e., consistent measurements between experiments), repeatable (i.e., under the same conditions), and reproducible (i.e., under different conditions). This is not only essential for customers wanting to use these techniques in court, but also for ensuring that any intelligence resulting from the media is reliable.

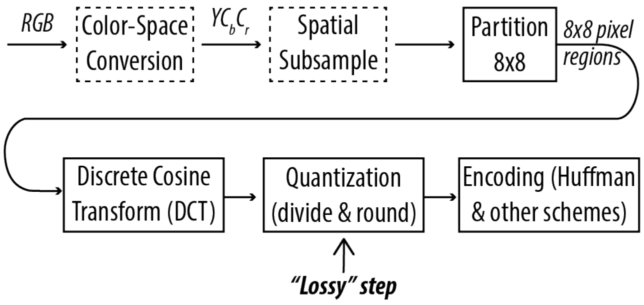

RGB
Color-Space Conversion
YC_bC_r
Spatial Subsample
Partition 8x8
8x8 pixel regions
Discrete Cosine Transform (DCT)
Quantization (divide & round)
Encoding (Huffman & other schemes)
"Lossy" step

FIGURE 10. The most common format for storing a digital image is a JPEG, which is the lossy compression method illustrated in this flowchart.

A starting point to good forensic practices is to make a copy of the media in question so that the evidence cannot accidentally be modified or lost. Always begin by checking the structure of the file (e.g., name, hash, format, EXIF data, quantization tables, and/or hex). Often manipulation software will insert their name in the data upon opening or manipulating the file.

Next, look for content inconsistencies (e.g., inconsistent shadows and reflections). If none are apparent, other authentication tools can look for traces of manipulations that are not visible to the eye. Any output of such tools should be saved in a lossless compression format, for example PNG.

Traditional techniques

Many traditional authentication algorithms focus on the quantization of the media. We will now discuss the details of how images are created. Taking a photograph using a digital camera, visible light reflecting off the scene is converted to a digital signal. In many cases, the digital image can be thought of as a three-dimensional array, representing the red, green, and blue (RGB) color intensities of the light at each pixel from the scene. Each color channel is represented by a byte taking values in the range of 0 to 255, although sometimes are represented from -128 to 127.

The most common format for storing a digital image is a JPEG, which is a lossy compression method. Many digital forensic tools are specific to JPEGs and exploit their properties. A flowchart of the compression process can be found in figure 10.

Essentially, an RGB digital image is converted to YCbCr color-space, the space is subsampled[c], partitioned into 8x8 blocks, and converted to the frequency domain via the Discrete Cosine Transform (DCT). It is then quantized by taking each block and entrywise dividing this block by another 8x8 array called the quantization table[d] and then rounding (this is one of the main sources of lossiness in the algorithm). Since the Y color channel contains the most visual information of the image, its quantized version is often a place to look for artifacts. This is what many traditional forensic algorithms are based on. When using the output of these algorithms, it is important to understand both their strengths and weaknesses.

[c] The Cb and Cr channels are often cut in size prior to the DCT transformation by a process called subsampling. Typically a 2×2 block of pixels is replaced with a single pixel either representing the maximum or the average of the pixels in the block.

Return to CONTENTS

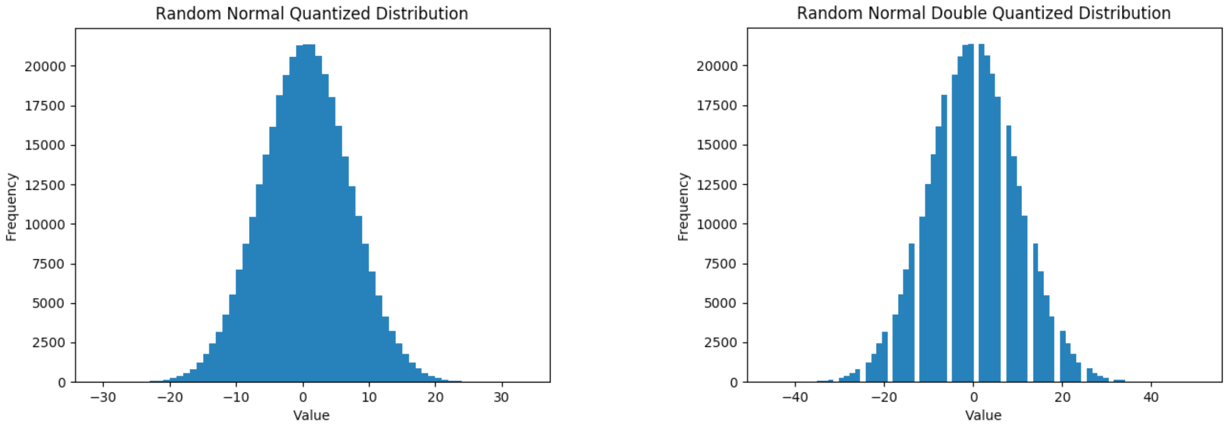


FIGURE 11. Double compression affects random normal quantized distribution as is illustrated in this simulation. As a result, the 192 histograms (64 values x 3 channels) of the DCT coefficients are more likely to have empty bins.

Compression affects the distribution of DCT coefficients since they are discretized via the rounding step. As a result, the 192 histograms (64 values x 3 channels) of the DCT coefficients are more likely to have empty bins (see figure 11). This characteristic has been exploited by several forensic algorithms [19, 20] specifically looking for double quantization effects. These algorithms have some drawbacks such as being sensitive to compression quality parameters, or that unnatural images (e.g., scanned documents) can cause false positives.

Another set of forensic techniques detects evidence of a copy-clone manipulation. These techniques have a similar underlying idea. At a high level, an image is first divided into overlapping blocks of a specified size and a feature vector is computed on each block. Next, a comparison is performed between pairs of feature vectors to measure similarities in order to detect potential regions of copy-clone. See [21] for a specific example and figure 12 for an example output of such an algorithm. One caveat of these algorithms is that the outputs are highly dependent on the parameters (e.g., block size used to compute each feature vector). Another caveat is that images containing large areas of similar content may generate false positives.

The previous examples have illustrated that although there are many papers focusing on robust authentication methods for specific manipulations, they have all fallen short of reliably detecting ALL manipulations. The DARPA MediFor program and the researchers at the Department of Defense have moved the field one step closer to the goal of universally detecting all media manipulations through the fusion of specialized techniques to identify manipulations [18]. However, it is imperative that those examiners using these new algorithms understand how to interpret them.

Deepfake detection techniques

Due to the hype surrounding deepfakes, there has been a lot of recent work on developing deepfake-specific detection techniques. A sample of those techniques is presented here. The first example is an

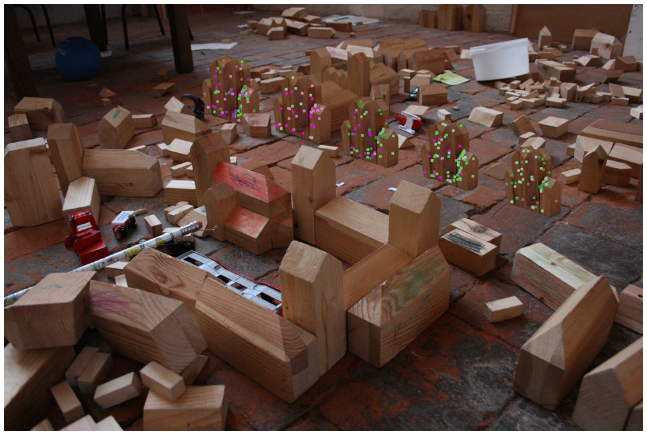


FIGURE 12. This image is an example output of a copy-move detection algorithm run on a manipulated image from [5].

[d] The quantization table determines how much information is lost in the compression process. The tables are designed to produce a visually appealing result at the human-specified level of compression.

algorithm that determines the authenticity of videos of well-known public political figures, such as President Obama and President Trump [22]. The group developed an algorithm that was trained on real videos of the person of interest, and tested on real videos, synthetically modified videos, and impersonators. Their trained model can tell the difference between a face swap and a real video, by picking up on particular cues from the person. An example of the "soft biometric" they are able to capture is that Obama tends to tilt his head when he says "Hello everyone" during his weekly addresses while President. Face swaps and impersonators cannot fully pick up all of these tiny movements that a person naturally does. The drawback to this modeling approach is that a lot of data is needed to create a robust detector that can learn all of the idiosyncrasies of a particular person. The Obama detector trained on about 19 hours of authentic video. This amount of data is prohibitive for most use cases as there must be a substantial existing pool of authentic videos of the person of interest.

Another authentication technique for synthetic media takes an idea from camera forensics—that of camera fingerprinting. A camera fingerprint is a low-level noise residual in the pixel values of the image; that noise residual pattern can be compared to an averaged noise residual from multiple pictures known to come from a specific camera. The two noise residuals are correlated to test whether the image was taken by the query camera. This idea is extended to GAN-generated images, in that media created by the same exact trained GAN should have a similar noise residual, and ones created by different GANs should have significantly different noise residuals. In [23], the authors compare the noise residual of a single image of interest to the averaged noise residual of images known to be from the same GAN. The classifier they developed determined the correct trained GAN for each image with about 90% accuracy. Some limitations of this method are that access to the trained GAN to produce a collection of images or access to already produced images from the trained GAN are required to create the average noise residual, and that the algorithm cannot attribute images from GANs not in the training set.

Finally, in [24], the authors trained a neural network to classify a data set of images generated from four different GANs and real data, and determine the source of a new, unseen image. The neural network takes as input the query image, and outputs one of the five classes: one of the four GAN architectures or the real data set. This method attains about 98% accuracy on the correct class. The algorithm is forced to make a prediction on any input image into one of the five pretrained classes. If, for example, an image generated by a new GAN was classified by this algorithm, it would still give a result (albeit meaningless) even though the correct class (the new GAN) is not an option.

Deepfakes as an alternative to media content creation

The COVID-19 quarantine has cultivated a need for alternative methods for media content creation and provided the time to explore new methods. Deepfake technology has already been widely applied to late night television shows postrecording (e.g., Bill Hader impressions [25]); however, during quarantine, late night shows (e.g., Jimmy Kimmel) have started to use deepfake videos as part of their show content [26]. On a larger scale, media companies are working toward commercial usage of synthetic media. For example, Disney is researching the use of high-resolution face swap technology for visual effects [27] that are almost good enough for commercial projects. Additionally, in the 2020 baseball season, Fox Sports added synthetic fans to empty baseball stadiums [28] that were adjusted based on the status of the game and weather.

As synthetic media becomes more prevalent in media creation, we also expect it to play a more extensive role in cybercrimes. In March of 2019, a high-profile case of a $243,000 fraudulent transfer resulted from the use of a chief executive's voice impersonation created by what was believed to be commercial voice-generating software [29].

Conclusion

With increased accessibility to media manipulation software, the problem of multimedia authenticity is a topic that warrants more general awareness. The rapidly growing areas of AI will only make the landscape of manipulations grow and become more complex. Consequently, we must be vigilant in identifying the truth in multimedia by verifying sources and being knowledgeable and cognizant of our senses. Concurrently, we must trust and support the forensic analyst community to continue developing mechanisms to aid our ability to identify the fakes.

Return to CONTENTS

References

[1] Farid H. *Digital Image Forensics*. 2013. Available at: https://farid.berkeley.edu/downloads/tutorials/digitalimageforensics.pdf.

[2] Kidd C. "How to know." *33rd Conference on Neural Information Processing Systems (NeurIPS);* 2019 Dec 9; Vancouver, Canada.

[3] Schiff A, Murphy S, Curbelo C. Letter to ODNI on Deep Fakes, 2018 September 13. Available at: https://schiff.house.gov/imo/media/doc/2018-09%20ODNI%20Deep%20Fakes%20letter.pdf.

[4] ASTM International. "ASTM E2916-13: Standard terminology for digital and multimedia evidence examination." 2013. Available at: http://www.astm.org/cgi-bin/resolver.cgi?E2916-13.

[5] Christlein V, Riess C, Jordan J, Riess C, Angelopoulou E. "An evaluation of popular copy-move forgery detection approaches." *IEEE Transactions on Information Forensics and Security*. 2012;7(6):1841–1854. doi: 10.1109/TIFS.2012.2218597. Data set available at: http://www5.cs.fau.de/research/data/image-manipulation/.

[6] Gloe T, Böhme R. "The 'Dresden Image Database' for benchmarking digital image forensics." *Journal of Digital Forensic Practice.* 2010;3(2–4):1584–1590. doi: 10.1080/15567281.2010.531500. Database available at: http://forensics.inf.tu-dresden.de/ddimgdb.

[7] Adobe Enterprise Content Team. "Amplifying human creativity with artificial intelligence." 2019 April 4. Available at: https://theblog.adobe.com/amplifying-human-creativity-with-artificial-intelligence.

[8] Kingma D, Welling M. "Auto-encoding variational bayes." In: *Proceedings of the 2nd International Conference on Learning Representation (ICLR)*; 2014 Dec 20.

[9] Van den Oord A, Dieleman S, Zen Heiga, Simonyan K, Vinyals O, Graves A, Kalchbrenner N, Senior A, Kavukcuoglu K. "WaveNet: A generative model for raw audio." 2016 September 12. Cornell University Library, arXiv: 1609.03499.

[10] Hochriter S, Schmidhuber J. "Long short-term memory." *Neural Computation.* 1997;9(8):1735–1780.

[11] I Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y. "Generative adversarial nets." In: *Proceedings of the 27th International Conference on Neural Information Processing Systems, volume 2;* 2014 Dec: pp. 2672–2680.

[12] Gulrajani I, Ahmed F, Arjosky M, Demoulin V, and Courville A. "Improved training of Wasserstein GANs." 2017 Dec 25. Cornell University Library, arXiv:1701.07875v3.

[13] Kaiming H, Xiangyu Z, Shaoqing R, and Jian S. "Deep residual learning for image recognition." 2015. Cornell University Library, arXiv:1512.03385.

[14] Karras T, Aila T, Laine S, Lehtinen J. "Progressive growing of GANs for improved quality, stability, and variation." In: *Proceedings of the 2018 International Conference on Learning Representations;* 2018. Available at: https://openreview.net/pdf?id=Hk99zCeAb.

[15] Karras T, Laine S, Aittala M, Hellsten J, Lehtinen J, Aila T. "Analyzing and improving the image quality of StyleGAN." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition;* 2020. Available at: https://openaccess.thecvf.com/content_CVPR_2020/papers/Karras_Analyzing_and_Improving_the_Image_Quality_of_StyleGAN_CVPR_2020_paper.pdf.

[16] Faceswap. 2019. Github repository. Available at: https://github.com/deepfakes/faceswap.

[17] Zucconi A. "An introduction to DeepFakes. Part 6: Understanding the technology behind DeepFakes." 2018 March 14. Available at: https://www.alanzucconi.com/2018/03/14/understanding-the-technology-behind-deepfakes.

[18] Turek M. "Media forensics (MediFor)." Defense Advanced Research Projects Agency. Available at: https://www.darpa.mil/program/media-forensics [accessed 2019].

[19] Popescu AC. "Statistical tools for digital image forensics," 2005. PhD thesis, PhD dissertation. Department of Computer Science, Dartmouth College, Hanover, NH.

[20] Lukas J, Fridrich J. "Estimation of primary quantization matrix in double compressed JPEG images." In: *Proceedings of Digital Forensic Research Workshop;* 2003.

[21] Fridrich J, D Soukal, Lukas J. "Detection of copy-move forgery in digital images." *International Journal of Computational Science.* 2003;3:55–61.

[22] Agarwal S, Farid H, Gu Y, He M, Nagano K, Li H. "Protecting world leaders against deep fakes." In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops;* 2019: pp. 38–45.

[23] Marra F, Gragnaniello D, Verdoliva L, Poggi G. "Do GANs leave artificial fingerprints?" In: *2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR);* 2019: pp. 506-511.

[24] Yu N, Davis L, Fritz M. "Attributing fake images to GANs: Learning and analyzing GAN fingerprints." In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV);* 2019: pp. 7556–7566.

[25] YouTube. Username: Ctrl Shift Face. "Bill Hader impersonates Arnold Schwarzenegger [DeepFake]." Uploaded 10 May 2019; accessed 7 Aug 2020. Available at: https://www.youtube.com/watch?v=bPhUhypV27w.

[26] Niemietz B. “See it: Jimmy Kimmel’s deep-fake video of Trump’s face on a fibbing baby is priceless.” *Daily News.* 2020 May 13. Available at: https://www.nydailynews.com/snyde/ny-jimmy-kimmel-deep-fake-trump-baby-coronavirus-20200513-uvw3ybqdsnbshctta4hxbyradm-story.html.

[27] Neruniec J, Helminger L, Schroers C, Weber RM. “High-resolution neural face swapping for visual effects.” *Eurographics Symposium on Rendering*; 2020 Jun 29. Available at: http://studios.disneyresearch.com/2020/06/29/high-resolution-neural-face-swapping-for-visual-effects/.

[28] Concha J. “Fox Sports to add virtual crowds to MLB ballparks.” *The Hill.* 2020 Jul 23. Available at: https://thehill.com/homenews/media/508661-fox-sports-to-add-virtual-crowds-to-mlb-ballparks.

[29] Stupp C. “Fraudsters used AI to mimic CEO’s voice in unusual cybercrime case.” *The Wall Street Journal.* 2019 Aug 30. Available at: https://www.wsj.com/articles/fraudsters-use-ai-to-mimic-ceos-voice-in-unusual-cybercrime-case-11567157402#:~:text=Catherine%20Stupp,-Biography&text=Criminals%20used%20artificial%20intelligence%2Dbased,intelligence%20being%20used%20in%20hacking.

The Attribution Solution: Who Said What, When?

Ryan Kaliszewski

In 1995, the Unabomber demanded that prominent newspapers publish his 35,000-word manifesto. Phrases that Ted Kaczynski used in personal communications with his brother David Kaczynski and sister-in-law Linda Patrick, such as "Technology has already made it impossible for us to live as physically independent beings," sounded eerily similar to the Unabomber's writings. After turning his brother in, David Kaczynski said, "[The manifesto] just sounded like my brother's voice." Can we similarly teach computers to distinguish authors and identify coauthorship just through what is written?

One of the greatest weapons in propaganda and information warfare is an adversary's ability to blend in and pretend to be a trusted source. In a world of quickly developing deepfakes, throw-away social media accounts, botnets, computer-generated text, and coordinated influence operations, it is becoming increasingly difficult to trust anything that we see or read. Authorship attribution—or being able to identify the true author of a body of text—can be an important way to pierce deception and provide confidence in modern communications.

Throughout the COVID-19 pandemic, medical and vaccine researchers were publishing the results of clinical trials and their lab research, the Centers for Disease Control and World Health Organization were issuing guidance, and local and national leaders were having to make difficult decisions balancing economic welfare with personal safety. In this shifting and dangerous environment, it is easy for malicious actors to produce disinformation and spread it under a legitimate name or for ambitious influencers to put a famous name on their work for instant attention, and it is often hard to differentiate between the two. Consider the example of a viral post centered around spiritual beliefs associated with the COVID-19 pandemic and allegedly written by Bill Gates [1]. The author of the article points out the unorthodox writing style and capitalization to suggest that the article was not written by Gates, with his sources speculating that the name was attached to promote visibility. Having tools to distinguish between peer-reviewed research and valid government direction versus political manipulation and sensationalism helps to fight the uncertainty and potential loss of life caused by these actors.

Problem background

Stylometry, or the study of linguistic style, and its application to authorship attribution arose as a serious research topic in the nineteenth and early twentieth centuries with pioneering work by Mendenhall [2] on the plays of Shakespeare and later statistical analyses by Yule [3] and Zipf [4]. Perhaps the most famous authorship attribution problem and solution was contributed by Mosteller and Wallace [5] on the authorship of *The Federalist Papers* (see figure 1), where experts disputed whether Alexander Hamilton or James Madison had authored several unsigned papers. Mosteller and Wallace were able to discriminate between Hamilton and Madison based on a statistical analysis of common words such as "and" and "but."

The successful use of statistics legitimized stylometry and authorship attribution as a branch of mathematical study. Soon all manner of stylistic features entered the collective research domain: sentence length, word length, word frequencies, character frequencies, and vocabulary richness are just some examples. Near the end of the late twentieth century, authorship attribution was accepted to the point of being admissible in court as expert evidence.

The true attribution solution—being able to identify any author through any sample of writing—is far too difficult to approach in general. Consider the example "Yes." This single word tells you nothing about the author, and the list of candidate authors includes everybody who ever lived! In fact, many people have authored this exact text, and from that simple word, it is impossible to distinguish which author provided any particular sample of the word "Yes."

This is why traditional authorship attribution has been done under specific assumptions. For example, in determining the author of unsigned content within *The Federalist Papers,* the true author was known to be among a small collection of candidates—this is an example of a *closed-set* attribution problem. On the other hand, in an *open-set* attribution problem, no restrictions are placed upon the authors; however, because attributing a single sentence is very difficult unless there is some truly unique phrasing or underlying meaning, such as in the Kaczynski example, we must assume that there are many writing samples available for each author. Modern scenarios of author attribution better fit the open-set problem. They often involve emails, social media posts, or text messages by an author with an uninformative or deceptive screen name, but we typically have access to more writing samples to aggregate into one corpus. The more samples present, the more likely we can successfully attribute authorship.

Since authors are represented by their writing samples in authorship attribution, we can restate the open-set problem as comparing two text corpora in order to determine whether they were written by the same author. For example, due to similar writing

Return to CONTENTS

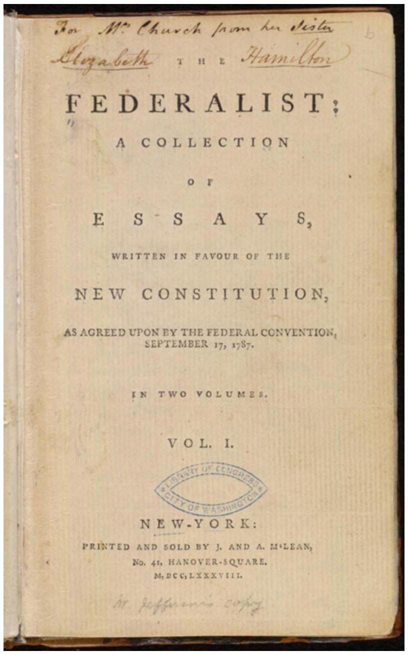
styles, some people believe Anthony Bourdain posted on a Brazilian jiu-jitsu blog under a pseudonym. There are several blog posts that could be concatenated into one text corpus under the pseudonym and another corpus consisting of Bourdain's articles. This variation of the open-set authorship attribution asks how likely that the two corpora were written by the same person given just the text in the corpora. This is the research question that my team in the NSA Research Directorate focused on, and we will discuss our methods and results in the following paragraphs.

Generalizing the closed-set to open-set problem

With the rise of advanced computing resources, closed-set authorship attribution methods have included sophisticated algorithms such as deep neural nets [6], genetic algorithms [7], support vector machines [8], and ensembles of classifiers [9]; however, the open-set problem has remained relatively untouched. Contributions by Stolerman, Overdorf, and Greenstdadt [10] toward a mixed open- and closed-set authorship attribution and recent work by Badirli, et al. [11] on open-set attribution of Victorian authors have demonstrated stylometry to be a promising approach to the open-set problem.

At the PAN-CLEF 2018 authorship attribution shared task, Custódio and Paraboni [12] introduced a closed-set authorship attribution method based on an ensemble of logistic regressions trained on stylometry. On the PAN-CLEF data set, Custódio and Paraboni's method outperformed other methods based on recurrent and convolutional neural nets, and their method maintained its high-performance ratings across five languages: English, French, Italian, Polish, and Spanish.

Since Custódio and Paraboni's method uses logistic regression at its base, it tests whether stylometric features appearing in its training data are common to a corpus. The method classifies the candidate text corpus into one of several bins based on which features are present. Therefore, if we remove the classification step, we have an algorithm that tests whether or not certain stylometric features appear within a corpus. The weights of the stylometric features that Custódio and Paraboni's method compares are learned through the training data.



For Mrs Church from her Sister
Elizabeth THE Hamilton
FEDERALIST:
A COLLECTION
OF
ESSAYS,
WRITTEN IN FAVOUR OF THE
NEW CONSTITUTION,
AS AGREED UPON BY THE FEDERAL CONVENTION,
SEPTEMBER 17, 1787.
IN TWO VOLUMES.
VOL. I.
LIBRARY OF CONGRESS
CITY OF WASHINGTON
NEW-YORK:
PRINTED AND SOLD BY J. AND A. M'LEAN,
No. 41, HANOVER-SQUARE.
M,DCC,LXXXVIII.
Mr. Jefferson's copy

FIGURE 1. Perhaps the most famous authorship attribution problem and solution was contributed by Mosteller and Wallace [2] on the authorship of *The Federalist Papers*. [Photo credit: Library of Congress, Rare Books and Special Collections Division]

To address the open-set problem, we propose a variation of Custódio and Paraboni's method where we determine stylometric features from our two test corpora and then generate a similarity metric instead of a classification. This similarity metric represents a comparison of writing styles, where a high score suggests that the corpora were coauthored. The stylometric features that Custódio and Paraboni used were three types of n-grams: character-grams, punctuation-grams, and word-grams.

An n-gram is a sequence of *n* consecutive objects that appear within the text. A character-gram is a sequence of upper and lowercase characters that will help to keep track of unusual spellings, capitalization, and typographical errors. A punctuation-gram is a sequence of characters in which all letters that lack diacritics have been replaced by asterisks. For example, in a punctuation-gram the keystrokes *a'* would be replaced with *', but the diacritical *á* would be retained. This helps to track punctuation styles. A word-gram is a sequence of words that will help track unusual phrases or irregular grammar.

Cleaning text and determining phrases

At the next PAN-CLEF (2019) Muttenthaler, Lucas, and Amann [13] refined Custódio and Paraboni's methodology and suggested the following text preprocessing to improve performance:

- Replace all digits with 0 placeholders,
- Replace all URLs, email addresses, and other hyperlinks with @,
- Do not lowercase text,
- Do not remove stop words,
- Restrict to character-grams of length 2 to 5, punctuation-grams of length 1 to 3, and word-grams of length 1 to 3.

This is because digits and the structure of hyperlinks do not reveal information about writing style, but their presence and frequency are certainly indicative. Similarly, stop-word usage has been successfully used to attribute authors in closed-set attribution; for example, in *The Federalist Papers* attribution many of the common words that Mosteller and Wallace used were stop words. Capitalization can be indicative, but specific character distributions indicate statistical biases of the underlying language rather than the author, so character 1-grams should be excluded. Upper bounds on n-grams need to be included to limit vocabulary size and Muttenthaler, Lucas, and Amann tested to find the optimum upper bounds. Since peculiar vocabulary and symbology may indicate authorship, they suggest retaining word and punctuation 1-grams.

We followed all of the advice of Muttenthaler, Lucas, and Amann with the exception that we lowercased words for word-grams. We did this to simplify the vocabulary, and we felt that we were already taking capitalization into consideration when we evaluated character-grams.

Rather than use term frequency-inverse document frequency (TF-IDF) or singular value decomposition to reduce vocabulary complexity, as Muttenthaler, Lucas, and Amann suggest, we reduced our data set to only include n-grams of statistical importance, which we will call *phrases*. For example, the word-gram "Great Wall" means more than the individual words "great" and "wall" so we consider it a phrase. Conversely, the word-gram "it is" means exactly what the sequence ("it," "is") tells us, so we don't consider it a phrase. Phrases could include unusual punctuation patterns, misspellings, or slang; each of which is highly indicative of authorship.

To detect phrases, we will rely on a Bayesian model introduced by Gunel and Dickey [14]. The work of Hannah and Wallach [15] assigns prior beliefs and a threshold to determine when a 2-gram appears more frequently than predicted by the underlying corpus. When the odds ratio of a 2-gram appearing in concert against the 1-grams appearing independently exceeds 10, then they consider the 2-gram is a phrase.

Even though Hannah and Wallach restrict their work to 2-grams, we can extend this result to any n-gram $(w_1\ w_2 \cdots w_n)$, $n \geq 3$ by decomposing it into a 1-gram and an (n-1)-gram in two different ways:

$$(w_1)(w_2 \cdots w_n), \qquad (w_1 \cdots w_{n\text{-}1})(w_n).$$

If the odds ratio of either of these decompositions exceeds 10, then we will conclude that $(w_1\ w_2 \cdots w_n)$ is a phrase.

Similarity metric

After extracting the character-, punctuation-, and word-gram phrases from each corpus, we need a way to decide if the corpora are similar. Since the collection of phrases is a set, a natural similarity measure is *Jaccard (set) similarity*: J(A,B)=(|A∩B|)/(|A∪B|).

For corpus i, let T_i denote the set of character-phrases, P_i denote the set of punctuation-phrases, and W_i denote the set of word-phrases. Define the similarity to be

$$Sim(T_1,T_2) = \alpha \cdot J(C_1,C_2) + \beta \cdot J(P_1,P_2) + \gamma \cdot J(W_1,W_2),$$

with α, β, γ being weights that we set to $\alpha = \beta = \gamma = 1/3$. While these parameters could be tuned, we chose to leave them equal since we had no prior knowledge about which n-grams would be more indicative of authorship. If the similarity of the two texts is high, this is evidence that the corpora were written by the same author. If the similarity is low then it is evidence that the corpora were written by different authors.

Data set and results

We tested our results using the Reuter_50_50 Data Set [16], a subset of the Reuters Corpus Volume 1 (RCV1) data set. The Reuter_50_50 Data Set contains 5,000 articles from 50 authors who each contributed 100 articles. This data set is used as a benchmark for authorship attribution, and so the articles are chosen to contain at least one subtopic of the corporate/industrial category (CCAT) in order to minimize topic factor in attribution. The articles have been split into two equal sets, a training set consisting of 50 articles per author and a non-overlapping testing set consisting of the other 50 articles per author.

For each author, we concatenated all of their articles from the training set into a *training* corpus and, similarly, concatenated all of their articles from the testing set into a *testing* corpus. We then ran our algorithm to compare every training corpus to every testing corpus. On a desktop computer, it took approximately 24 minutes to read all of the text, prepare it, and identify phrases. It then took approximately two minutes to generate the similarity scores.

Open-set results

Recall that for the open-set problem, we must make each decision about coauthorship independently. That is, when we compare a testing corpus with a training corpus we cannot look at how the testing text corpus compared against other corpora. We can only look at the current score and use that to decide coauthorship. Thus, we need to find some threshold, $0 < \tau < 1$, so that if the similarity equals or exceeds τ we can conclude the corpora were coauthored with some confidence.

We chose values for τ from the interval 0.30, 0.45 in 0.01-step intervals. If, for training corpus T_1 and testing corpus T_2, $Sim(T_1,T_2) \geq \tau$ and T_1,T_2 were coauthored, then we considered this a true positive, and if they were not coauthored, we considered this a false positive. If $Sim(T_1,T_2) < \tau$ and T_1,T_2 were coauthored, we considered this a false negative, and if they were not coauthored, then we considered it a true negative.

We computed the recall, precision, and Matthews correlation coefficient (MCC) for each value of τ and plotted the results in figure 2.

In figure 2, as the threshold increases, recall begins to drop and precision begins to increase. There is a crossing point just above , which is also near the maximum value of the MCC. Since this is approximately the maximum performance of our algorithm, we will conclude empirically that $Sim(T_1,T_2) > 0.40$ gives strong evidence that T_1,T_2 are coauthored.

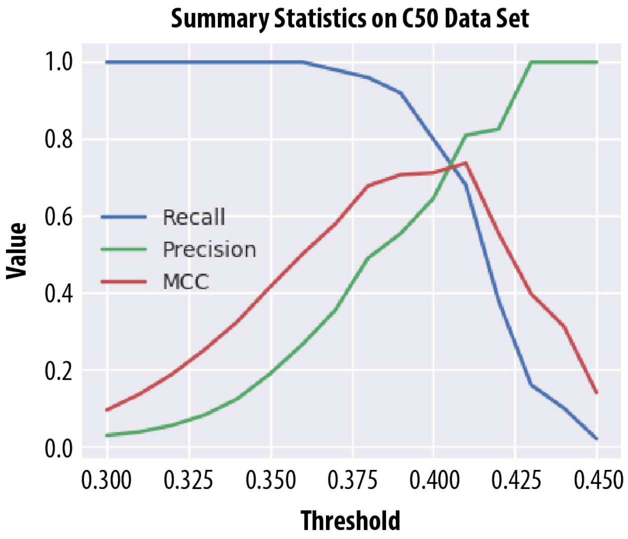


FIGURE 2. The plot of the recall, precision, and Matthews correlation coefficient (MCC) for values of the similarity threshold (x-axis). The confluence of the curves between 0.40 and 0.41 suggests that a similarity score over 0.40 indicates strong evidence of coauthorship.

Under this assumption, our method has an estimated recall of 0.8, an estimated precision of 0.645, and an estimated MCC of 0.712.

Mixed-set results

If we add the additional assumption that we know that the author is present in our training set, we can simply associate the testing corpus to the training corpus with the highest similarity score. This is not quite the same as a true closed-set authorship attribution task, because we did not attempt to attribute each testing article individually, but rather we lumped the testing articles into one testing corpus.

Of the 50 authors, this method correctly associated 47 of them. We considered a correct association as a true positive and a correct nonassociation as a true negative. Each incorrect association was both a false positive and a false negative since the method did not identify the correct author and identified an incorrect author. The recall and precision of the algorithm was 0.94, and the MCC was 0.939.

We observed that for the three incorrectly associated testing corpora, the correct training corpora had the second highest similarity score. Therefore, in top-2 mixed-set attribution, our recall, precision, and MCC was 1.

Closed-set results

If we attribute each article in the testing set individually against the training sets, then the performance drops dramatically. Longer texts increase the likelihood of correct attribution since the author is more likely to use identifying writing. In this instance, the precision and recall were 0.1356 and the MCC was 0.118.

Because the articles are relatively short, our statistical method of phrase detection had trouble sifting out what is important. If we allow all n-grams to be phrases, we lose a powerful tool in attribution, but in the small document case, it improves performance with a recall and precision of 0.186 and an MCC of 0.169. Even if we consider a top-5 closed-set authorship attribution, the estimated precision and recalls only become 0.418.

Future work and conclusion

There are many unanswered questions about open-set authorship attribution. The most important observation is that this threshold value of 0.40 was found for *this data set*. There is no evidence that a threshold of 0.40 should work well for other authors. Can a universal "best" threshold be found?

The algorithm performed poorly against short articles, but very successfully against long corpora. At what point does the reliability degrade too much to be useful?

Also, there are many alternative choices that could have been made at each step. Phrase detection or similarity metrics could be revisited to ensure that there are not alternative options that would provide stronger results. We count n-grams for detecting phrases, but we could use those counts in the similarity score. Perhaps alternative similarities such as

$$Sim(A,B) = \frac{\sum_{w\in A\cup B} \min(f_1(w), f_2(w))}{\sum_{w\in A\cup B} \max(f_1(w), f_2(w))},$$

where $f_i(w)$ is the count of n-gram w in document i, or $f_i(w)$ is the odds ratio used for phrase detection, or some other function. Perhaps n-grams' frequencies should be treated as distributions and a distribution metric should be used to determine similarity. Also, changing the weights between character-grams, punctuation-grams, and word-grams could improve performance.

We have a candidate method for open-set authorship attribution that is based on a generalization of a very successful closed-set authorship attribution. This method performs very well on the open-set problem and a mixed-set problem when applied to the Reuter_50_50 Data Set. While our algorithm is a useful first step in open-set authorship attribution, there are many avenues of research open for further improvement.

References

[1] "A fake Bill Gates quote about the coronavirus has jumped from WhatsApp to major news outlets." *Buzzfeed.News*. 2020 Mar 25. Available at: https://www.buzzfeed.com/joeydurso/coronavirus-fake-bill-gates-quote [Accessed: 2020 June 24].

[2] Mendenhall TC. "The characteristic curves of composition." *Science.* 1887;9(214):237–249. doi: 10.1126/science.ns-9.214S.237.

[3] Yule GU. "On sentence length as a statistical characteristic of style in prose with application to two cases of disputed authorship." *Biometrika.* 1938; 30:363-390.

[4] Zippf GK. *Selected Studies of the Principle of Relative Frequency in Language.* Cambridge (MA): Harvard University Press; 1932.

[5] Mosteller F, Wallace DL. *Inference and Disputed Authorship: The Federalist.* Reading (MA): Addison-Wesley; 1964.

[6] Khosmood F, Levinson R. "Toward unification of source attribution processes and techniques." In: *Proceedings of the Fifth International Conference on Machine Learning and Cybernetics;* 2006 Aug 13–16; Dalian, China: pp. 4551–4556. doi: 10.1109/ICMLC.2006.258376.

[7] Holmes DI, Forsyth R. "The Federalist revisited: New directions in authorship attribution." *Literary and Linguistic Computing.* 1995; 10(2):111–127. doi: 10.1093/llc/10.2.111.

[8] Sanderson C, Guenter S. "Short text authorship attribution via sequence kernels, Markov chains and author unmasking: An investigation." In: *Proceedings of the International Conference on Empirical Methods in Natural Language Engineering*; 2006 Jul; Sydney, Australia: pp. 482–491. doi: 10.3115/1610075.1610142.

[9] Stamatatos E. "Authorship attribution based on feature set subspacing ensembles." *International Journal on Artificial Intelligence tools.* 2006;15(5):823–838. doi: 10.1142/S0218213006002965.

[10] Stolerman A, Overdorf R, Afroz S, Greenstadt R. (2014). "Breaking the closed-world assumption in stylometric authorship attribution." In: Peterson G, Shenoi S, editors. *Advances in Digital Forensics X. DigitalForensics 2014. IFIP Advances in Information and Communication Technology, Vol 433.* Berlin, Heidelberg: Springer, 2014. pp. 185–205. doi: 10.1007/978-3-662-44952-3_13.

[11] Badirli S, Ton MB, Gungon A, Dundar M. 2019. "Open set authorship attribution toward demystifying Victorian periodicals." Cornell University Library, arXiv: abs/1912.08259.

[12] Custódio J, Paraboni I. "An ensemble approach to cross-domain authorship attribution." In: Crestani F et al, editors. *Experimental IR Meets Multilinguality, Multimodality, and Interaction. CLEF 2019. Lecture Notes in Computer Science, Vol 11696.* Springer, Cham, 2019. pp. 201–212. doi: 10.1007/978-3-030-28577-7_17.

[13] Muttenthaler L, Lucas G, Amann J. "Authorship attribution in fan-fictional texts given variable length character and word n-grams." Notebook for PAN at CLEF 2019. 2019. Available at: http://ceur-ws.org/Vol-2380/paper_49.pdf.

[14] Gunel E, Dickey J. (1974). "Bayes factors for independence in contingency tables." *Biometrika.* 1974;61(3):545–557. doi: 10.2307/2334738.

[15] Hannah L, Wallach H. (2014). "Summarizing topics: From word lists to phrases." In: *NIPS 2014 Workshop on Modern Machine learning and Natural Language Processing*; 2014; Montreal, Quebec, Canada: pp. 1–5.

[16] Liu Z. Reuter_50_50 Data Set. (Donated by Zhi Liu of National Engineering Research Center for E-Learning Technology, Hubei Wuhan, China.) Hosted on the University of California Irvine Center for Machine Learning and Intelligence's Machine Learning Repository. Available at: http://archive.ics.uci.edu/ml/datasets/Reuter_50_50.

RaNdOm Is RoBuSt: Using Randomness to Make Classifiers Resistant to Attack

James Holt, Edward Raff

Adversarial machine learning has been a research area for over a decade [1], but it has recently received increased focus and attention. This is largely due to the success of modern deep learning techniques within the realm of computer vision tasks, and to the surprising ease with which such systems are fooled into producing incorrect decisions [2]. For example, there are concerns about the safety of self-driving cars, as machine learning attacks have demonstrated that they can be tricked into misreading stop signs as speed limit signs [3]. The decision to deploy a machine learning model must now factor in risk analysis and performance when under attack, in addition to classic concerns such as whether the model has sufficient accuracy, size, and throughput to "solve" the problem at hand.

[Photo credit: WMC, lmk3nnyma/CC, filtrefil]

Consider an adversary who wishes to attack a machine learning system, such as the vision system of the self-driving car mentioned above, and has easy access to that system to test how it behaves given different inputs. The attacker wants to develop a function that will make small alterations to a true input so the system will classify it incorrectly. The objective could be to make it give *any* wrong answer, or to make it give a *specific* wrong answer. Making it give a *specific* wrong answer is called a *targeted attack*. For example, starting with an image of a stop sign, this function will make small changes to individual pixels, causing the resulting image to be classified as a speed limit sign by the model. The changes are often so subtle that a human cannot tell the image has been altered. Attacks like this are

Return to CONTENTS

possible, easy to create, have a high success rate, and work in the real physical world. In the case of the stop sign, researchers were able to demonstrate that they could cause misclassification by putting several small stickers on the sign.

Now imagine that the owner of the self-driving car's visual system could build a function or transformation, to be applied to all images it receives, that somehow flushes out or disrupts the attacker's changes. That would be useful! If it were possible to find such a defensive transformation it would provide a simple and convenient way to circumvent the adversarial problem, effectively defeating a whole class of attacks. Many researchers have attempted to find such a transformation. Though several candidates have been proposed, to date all have been quickly defeated by attackers.

In the work described here, we introduce the idea of stochastically combining a large number of individually weak defenses into a single barrage of randomized transformations to build a stronger defense against adversarial attacks. We will outline how this algorithm works, both at training time and for evaluation of new images, and we will discuss the challenges of testing, our approach, and our test results. Finally, we will discuss future research directions that could follow this work.

Three types of adversarial attacks

There are currently three known types of adversarial attacks on machine learning systems. If we call the machine learning system targeted by an attack the *victim,* an attacker may wish to make the victim: 1) *do* the wrong thing, 2) *learn* the wrong thing, or 3) *reveal* the wrong thing [4]. In this article, we focus on 1), which is called an *evasion attack* in the academic literature.

There are also many different ways to use machine learning models. The most common is as a classifier, which will be the focus in this article. A classifier receives some input, for example an image, observes its features, and from a limited set of classes, chooses a specific class to which it thinks the image belongs. For example, a simple classifier may receive images and classify each image as "cat," "dog," or "other." Sophisticated models may have thousands of classes. If it is a good model, it will be right most of the time, but very few models are correct 100% of the time. We can think of "prediction" as an image paired with the class label that a model assigned to it, and "truth" as an image paired with its true class label. When these are the same, the model has made a correct classification.

The anatomy of an attack

Consider a single input image, which provides the features that the model will use to make a decision, and its associated true class. We expect a trained classification model, given this input, to produce the true class. In an evasion attack, the adversary will try to perturb this single input to produce a new input that tricks the victim model into making an erroneous decision. If the adversary has access to the victim model, these attacks are fairly easy to create. Even without full access to the internal data of the model, if the attacker is able to query the model or system without restriction, they can evade it [5] and even learn enough to steal a copy of it [6] which can also be used to create successful attacks. This type of attack can be done against machine learning systems that classify any type of data, not just those that classify images.

The concern with such evasion attacks is not just how easy they are to create, but also the surprising degree to which they are effective and the small amount of change necessary for these attacks to succeed. This is easiest to observe for image classification, where small changes to the pixels, imperceptible to humans, cause the victim model to make drastically different classification decisions. In the attack shown in figure 1 (on the following page), the model is tricked by imperceptibly small changes. The adversary intentionally makes these changes as small as possible so that they will not be noticed by a human observer and can thus avoid detection by manual inspection. It also makes it obvious that if the result is identical to a human, the answer should be identical too!

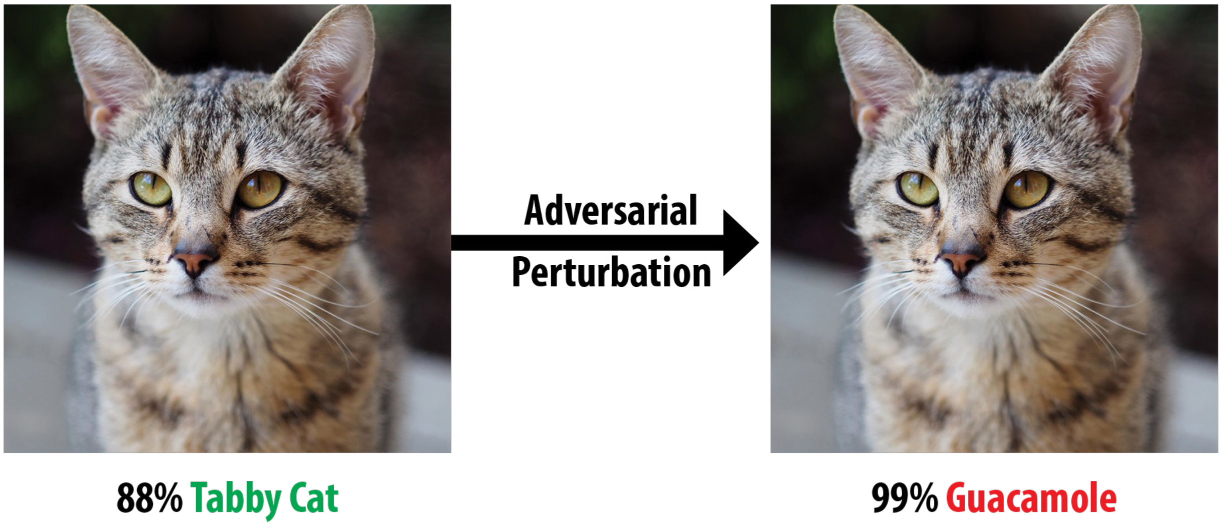


FIGURE 1. This is an example of an evasion attack. The left image is the original image of a cat, which the model correctly classifies as "tabby cat." The image on the right is an adversarial example in which a small perturbation has been applied that does not visibly change the image but convinces the model that the cat is actually guacamole, with higher confidence than the original correct prediction! [Photo credit: CC/filtrefil/public domain]

There have been many attempts to build models that are robust to such adversarial perturbations, but most have not proven effective. Some approaches work but cannot scale up without using massive resources [7]. Many more have simply been broken by attackers [2, 8]. Often this is because in evaluating a new defense, the developers of the approach forget to give the adversary knowledge that the new defense is in place. The effect, referred to as *obfuscated gradient* [8], is apparent success due to hiding information about defenses from the attacker during evaluation. Effectively, the defensive transformation being applied introduces an additional gradient that the attacker must account for and optimize over. Hiding this is not a good defense and may lead to a false sense of security. Research has shown that attackers can detect and defeat hidden gradients, overcoming new defenses whether they are made public or not. Taking this into account, we conduct our testing in a *white-box* fashion, which means we assume the adversary has access to all the same information as the defender/victim. A defense that is successful in the white-box scenario should only perform better in *gray-box* (i.e., where the attacker has limited information) or *black-box* (i.e., where the attacker has no knowledge of the model or defense) scenarios.

Adversarial training

One of the few, reliable, and most effective defenses to evasion attacks is *adversarial training*. This is when adversarial examples are created and included during model training. In theory this will teach the model to recognize the object in an image under many conditions, including when under attack. There is a significant amount of work along these lines showing that this defense reduces the success of attacks [9, 10, 11]. It has been very successful on smaller data sets, but computational costs present significant challenges when scaling the technique up to larger data sets like ImageNet [12].

Adversarial training is a massive undertaking at an ImageNet scale. Adversarial training requires training on attacked inputs, meaning each adversarial input goes through an extensive iterative optimization process. Testing by Xie, Wu, Maaten, et al. [13] required hundreds of graphics processing units (GPUs) just to build a single model.

The results on ImageNet, after all that processing, have been positive but modest. Furthermore, the scaling challenges indicate that there may be trouble ahead when we consider data sets that are orders of magnitude larger than ImageNet. In the results section below, we will show how our work compares with adversarial training approaches.

Adversarial training is a complex and rapidly evolving field. This section included a brief sketch intended to give those new to the field a glimpse; it is based on the state of the field when our work was done, but does not fully represent work in this field. See cited references throughout for more depth.

Defense by transformation

A major question is, can we circumvent the high cost of adversarial training while simultaneously retaining its benefits? An intuitive approach, that has since been shown as insufficient to defend a model, is that we should be able to defeat the adversary by introducing our own perturbations to the image before classifying it. Because the adversary's perturbations are so small, if we introduce bigger perturbations they will "wash away" the adversary's alterations and render the attack muted. Take for example a blurring transformation, this could be any Gaussian or box filter that makes an image look more fuzzy or blurry. If we apply this before sending an image to our model for classification at test time, we would hopefully obtain the same answer as before. The intuition is that blurring will alter the image so much that an adversary's tiny and innocuous perturbations will be overwhelmed and washed away.

Why did this strategy not work? Blurring the image clearly causes far more distortion and visible change than the adversary's perturbation in the guaca-cat in figure 1! This approach fell victim to the obfuscated gradient problem. Once the adversary became aware that the blur transformation was being been applied to the image, they were able to trivially defeat the defense. The adversary could make small random changes, put them through both the defensive transformation and classification, observe the change in classification, and use this signal to hone in on and optimize changes effective for their goals. This approach has been tried with many different transformations, all of which have been defeated.

In particular, the use of such transformations is defeated largely because the *same process is repeated for all inputs*. If your model is a neural network, you can imagine the transformation as being just another layer or two in the stack of layers, and thus can be defeated with the exact same approaches!

BaRT: The power of randomness

But what if we, the defenders, did not know what transformation was going to be used? Suppose we have a large collection of possible transformations, we choose a random subset of these transformations, and apply them in a random order (see figure 2). At training time, for each image in our training set, we train on the original plus many versions of it after random sets of transformations are applied. Each time we classify an image, we select and apply a new and different set of random transformations, hopefully leading our classifier to achieve the same answer regardless

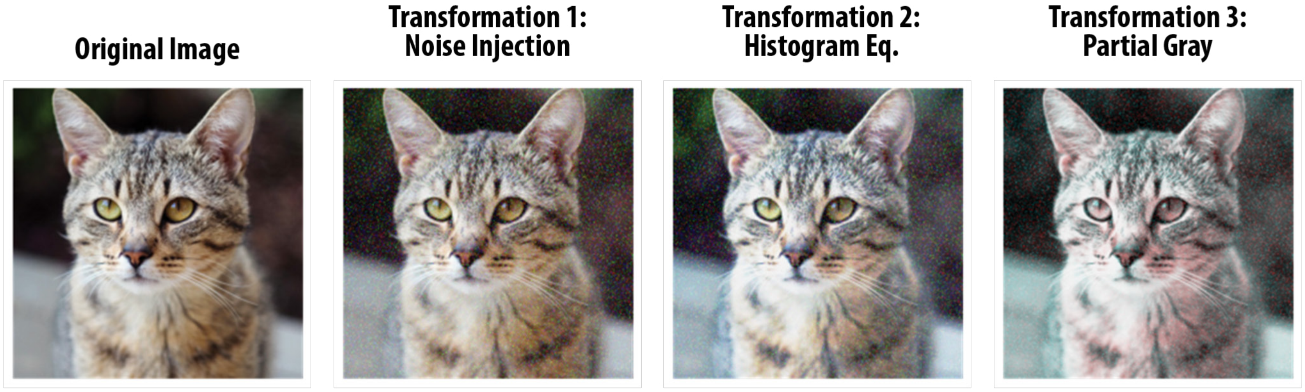

Original Image
Transformation 1: Noise Injection
Transformation 2: Histogram Eq.
Transformation 3: Partial Gray

FIGURE 2. These photos show a series of random transformations. By the final transformation, the original input has been significantly perturbed, but we and the adversary do not know how we will transform each image, making it difficult to "plan ahead."

of whether an adversary is attacking us. (But, realistically, we know there will be a cost in accuracy for doing this.)

For a representation of these ideas in mathematical notation, let's call the input data x, its true label y, the classification model $f(.)$, and the adversary's attack $A(.)$. A correct classification means $y = f(x)$. A successful attack would be $y \neq f(A(x))$. Our potential transformations are $t_1(\cdot), t_2(\cdot), \ldots, t_n(\cdot)$, and we select k of them, where $k \leq n$. The order of the transformations is represented by $\pi(1), \pi(2), \ldots, \pi(k)$. Accordingly, equation (1) below represents our desired outcome: when we apply our subset of transformations in random order, the classification result is the same regardless of whether the adversary has applied their attack function.

$$f(t_{\pi(1)}(t_{\pi(2)}(t_{\pi(\ldots)}(t_{\pi(\mathrm{k})}(x)))) = f(t_{\pi(1)}(t_{\pi(2)}(t_{\pi(\ldots)}(t_{\pi(\mathrm{k})}(A(x))))) \quad (1)$$

This is the essence behind our new approach to defending against adversarial attacks—to apply a ***Barrage of Random Transformations (BaRT)*** to the input [14]. The intuition is that when we have a fixed pipeline of transformations, the adversary need only learn how to defeat one system, and that it is only incrementally more challenging than before. But with BaRT, the adversary needs to find a single perturbation that works effectively against a combinatorial explosion of n choose k, or $\binom{n}{k} = \frac{n!}{k!(n-k)!}$, possible transformations! This, we hope, fundamentally changes the amount of work an adversary has to do to defeat a classifier. It also raises the possibility that no single perturbation exists that can defeat all $\binom{n}{k}$ possible transformations (see figure 3).

Building BaRT's adversary

To truly understand whether this defense is effective, we need to build an adversary for BaRT[a] that is as strong as possible and fully aware of how BaRT works. We want a fully white-box evaluation, which assumes that the adversary knows everything about the implementation of BaRT, and we want to use the most effective attacks available.

Our model is a neural network, and neural networks learn by means of a loss function. There are many different loss functions, but all of them measure in some way the prediction errors that a model makes. The model itself is a collection of weights, and these weights are adjusted during learning. A *gradient* is a mathematical technique that allows us to measure the direction of change in error as changes are made to one of the weights. In a single training run, a small set of training data is run through the model and the gradient of the error with respect to each weight is calculated. Using that gradient, the weights are adjusted, inching closer to the desired performance. Over many training runs, if the model, training, and data are constructed properly, the overall performance of the model improves.

A *gradient attack* works similarly, but instead of minimizing the error, we try to maximize it, and instead of updating the model weights, we update the input (i.e., make changes to the image to be classified). After many cycles of updates, the result is an input that is optimized to achieve the goal of the attacker. It is an adversarial example.

BaRT is a combination of two techniques: applying multiple transformations and randomizing transformations. To construct the best possible attacker to evaluate BaRT, we will use a combination of two attack techniques which we think will be most effective against BaRT's defenses.

The first attack requires us to be able to compute the gradient through every transformation applied. However, some of the transformations cannot be represented as a differentiable function; in other words, we do not know how to compute all of the gradients, so some can only be estimated. We do this for each transformation $t_i(\cdot)$ by learning a corresponding neural network $f_i(\cdot)$, which has the goal of approximating the transformation such that $t_i(x) \approx f_i(x)$. This is effective whether the function is differentiable or not, and the resulting neural network is always differentiable. This is called Backward Pass Differentiable Approximation (BPDA), and it is the standard attack for defenses that transform the input [8].

The second attack strategy we add deals with the randomness that BaRT introduces. Say the space of all possible combinations of transformations is T, and we sample a specific combination of transformations t from T. Using BPDA, we can compute the gradient through this specific combination. We take a new sample and repeat that process, then combine

[a] No, not Principal Skinner. ☺

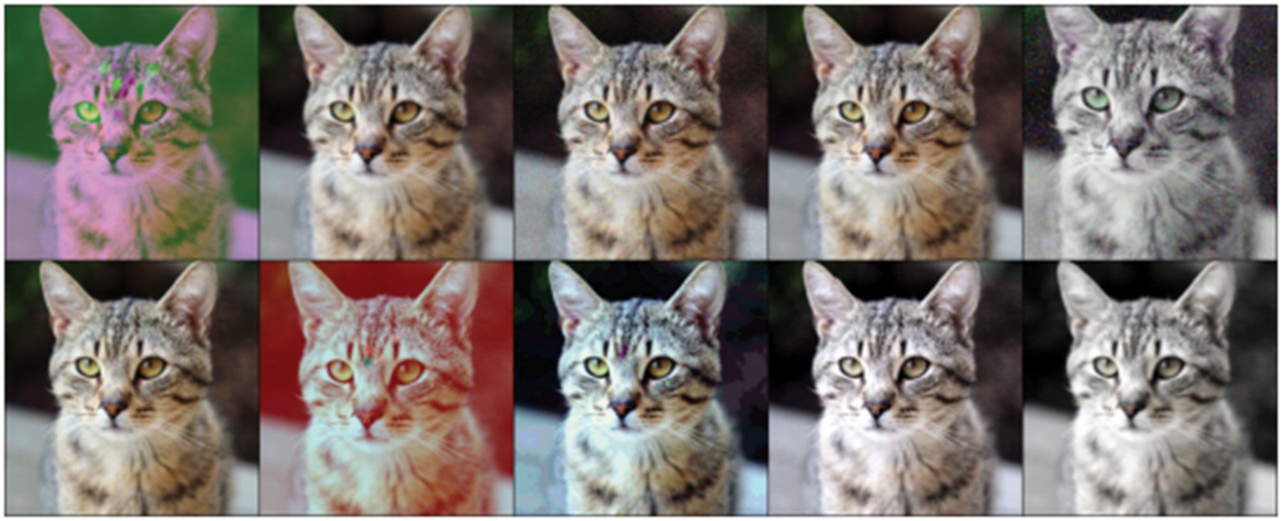


FIGURE 3. Starting with the same initial image, these photos show results from 10 different transformation chains; each one has five random transformations applied in random order. While we can tell that all of these come from the same image, and that it was a cat, the resulting images are very different.

the results across all samples, creating an estimate of what we expect to happen during a randomly selected set of transformations. This approach is called the Expectation over Transformations (EoT) [15] and has been used to train models to recognize objects from multiple viewing angles. The formula below represents estimation of the EoT by summing the gradients of samples. It says that the gradient of our expectation for the outcome of a given function over all possible transformations can be approximated by the sum of the gradients from a sample of outputs.

$$\nabla E_{t\sim T} f(t(x)) = E_{t\sim T} \nabla f(t(x)) \approx \sum_{i=1}^{z} \nabla f(t(x))$$

We have armed our adversary with two techniques, BPDA and EoT, which together allow sampling and estimation of gradients across all possible random sets of transformations. This provides a signal which indicates the direction to make changes. Another piece that is necessary is a way to use this signal to explore and optimize changes, with the goal of making the attack work. For this optimization, we use the Projected Gradient Descent (PGD) method, which is the strongest currently available technique for the type of perturbations we are considering [11].

Together, we believe these techniques constitute the best known approaches to defeat BaRT's defenses. This is an adversarial attack informed by full knowledge of how BaRT works and employing the best known attack techniques. We believe that this enables a strong evaluation of BaRT. All of that being said, we are only claiming that this is the best *known* way to attack BaRT *today,* as we write this. More on that after we take a look at the results.

Round 1: Evaluating BaRT performance

There are two more details to cover before we get into the results. Firstly, the data sets used for this work: ImageNet [12] is an image data set containing over 14 million images labeled with over 20,000 concepts or classes. The ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2012 [16] is a subset of the larger ImageNet which is very popular for machine learning research. It is a training set of 1.2 million images in 1,000 classes and a test set of 150,000 images with the same 1,000 classes. We used the ILSVRC 2012 data set for all of the work described here.

Secondly, let us describe the model architectures used. There are several neural network architectures that are in common use and have been shown to be effective for learning on ImageNet. One of those is ResNet-50, a 50-layer-deep residual learning convolutional neural network [17]. For all of our work, we start with a ResNet-50 network structure and learn from there. Inception [18] is a similarly complex neural network architecture that is popular in current research. Kurakin, Goodfellow, and Bengio did work on adversarial training [10] using the InceptionV3 network architecture. The adversarial training result we compare to below is from their work.

Now that we have described both BaRT and how we attack it, we can talk about results! Since the subset of ImageNet that we are testing has 1,000 different classes, we will consider the accuracy of the model when it gets the exact correct answer (i.e., top-1 accuracy), as well as when it gets the correct answer in its top five guesses (i.e., top-5 accuracy).

Table 1 shows a summary of our results, compared with previous results. A normal ResNet-50 model gets top-1 and top-5 accuracies of 76% and 93%. But, note that a PGD attack is always successful, giving top-1 and top-5 accuracies of 0%. Kurakin, Goodfellow, and Bengio's adversarial training improved upon this situation, achieving 5.5% success while under PGD attack. An improvement, but a modest one. Condition 3 shows what happens if we train the ResNet-50 model and apply the BaRT transformations at training time, but *not* at evaluation time. The accuracy while not under attack drops to 65% and 85%. But then we have to pay another price—because we do not know when we are under attack, we must use these transformations all of the time. That gets us to the condition 4: when we apply BaRT transformations at training time *and* evaluation time, our accuracy drops again to 54% and 76%. We show conditions 3 and 4 to separate out the costs of BaRT at training time and evaluation time. Condition 3 shows only the training time cost, while condition 4 shows both training time and evaluation time costs together. Condition 4 is where we start seeing some good news. In the PGD Attack columns, there is a significant improvement when under attack.

Round 2: Evaluating BaRT as an ensemble

Round 1 already shows a major improvement. But wait—we have one more trick up our sleeve. Because BaRT creates a new transformed version of the input for every evaluation, and it does so very efficiently, BaRT can be used *as its own ensemble*. In machine learning, *ensemble learning* refers to running an input through multiple classifiers and allowing those classifiers to *vote* to determine the output. Rather than classify an image once and use that as the answer, we take an image, run random transformations on it multiple times, classify each output, and use the most common answer as the final output. This significantly improves our results. We show in figure 4 that as the ensemble size increases, our accuracy on clean images (no attack) increases back to the level of BaRT trained only (condition 3 in table 1), causing us to recover the penalty we paid when applying BaRT transformations to inputs, for the small price of doing random transformations and classifying a few extra times.

Applying the ensemble technique to the tests we did in round 1, our top-5 accuracy under PGD attack also increases significantly, reaching 70%. Table 2 shows our previous table with a fifth condition added showing the results of using BaRT as an ensemble.

Round 3: Evaluating BaRT on targeted attacks

As we mentioned way back in the introduction, when the adversary seeks not just to make the classifier give a wrong answer, but to make it give a specific wrong answer, that is called a *targeted attack*. Targeted attacks present a much more significant danger. Calling a cat guacamole doesn't seem like such a big deal on the surface, but many machine learning systems in use today make classification decisions that feed into and

TABLE 1. Accuracies for various models, with and without PGD attack

		No Attack		PGD Attack	
Condition	**Model**	**Top-1**	**Top-5**	**Top-1**	**Top-5**
1	ResNet-50	76%	93%	0%	0%
2	Inception w/adversarial training	78%	94%	1.5%	5.5%
3	ResNet-50, BaRT trained	65%	85%	0%	0%
4	ResNet-50, BaRT trained & applied	54%	76%	16%	51%

inform real decisions and actions with consequences. Beyond just forcing models to make mistakes, which essentially is injecting noise into some decision process, if attackers can drive a classifier to a specific wrong classification, they may be able to make a larger system act in a certain way. Going back to the stop sign example: an image was not only classified wrong, but classified as a specific wrong class that the attacker desired, and this led to an autonomous vehicle driving through a stop sign without slowing down. The downstream decision informed by the classification can have significant consequences, and successful targeted attacks give attackers control of that classification.

Here we evaluate how well the BaRT defense works when an adversary tries to perform a targeted attack. Instead of any error being acceptable, i.e., $y \neq f(A(x))$, the adversary needs to fool the victim into making a specific error, i.e., $y \neq y_{\text{target}} = f(A(x))$. On normal ResNet-50, the adversary can achieve this targeted attack 100% of the time. But with BaRT, as we can see in figure 5 (on the following page), their ability to perform targeted attacks is reduced to 6% running single BaRT, and goes down to less than 1% as we combine more BaRT predictions in an ensemble.

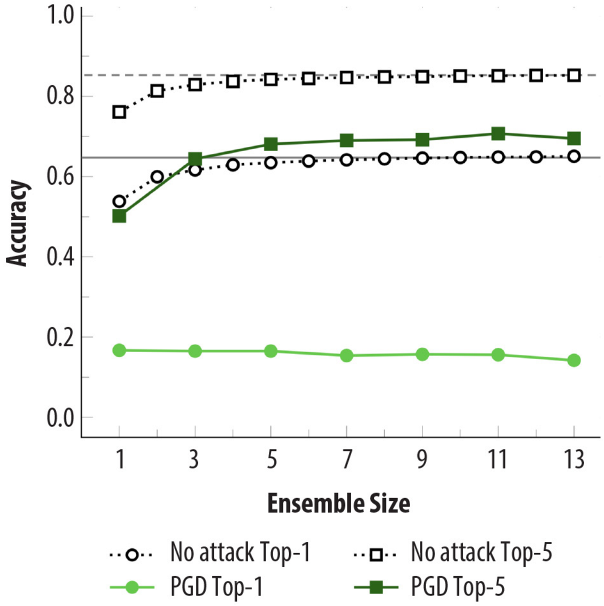


FIGURE 4. This graph shows the accuracy of BaRT when under no attack (black and white) and PGD attack (green) with different sizes of ensembles. Circles indicate top-1 accuracy, squares top-5 accuracy. Accuracies for all cases improve significantly from ensemble size 1 to 5 and flatten out after 5.

Computational efficiency and scaling

One final benefit of BaRT worth mentioning is computational. We believe, based on how the BaRT technique works, that it can easily scale up to larger data sets and larger numbers of classes. Running an ensemble of BaRT predictions takes only a few hundred milliseconds on decent hardware (not even using GPUs for image transformations, which could further increase speed). But, when we discuss the computational cost for the adversary, the situation becomes much worse. In our paper [14], where we have more evaluation detail, running inference for all of our experiments takes only a few hours. But running the attacks required over *seven GPU years* to perform! While this amount of computational resources is not insurmountable, it is significant. These are the kind of defensive system modifications we are interested in—those that have a reasonable and bounded computational cost for defenders but create exponentially increasing costs for attackers. In many security situations, asymmetry works in favor of the attacker. Here, through clever design, it works in favor of the defender.

TABLE 2. Accuracies for various models, with and without PGD attack

		No Attack		PGD Attack	
Condition	**Model**	**Top-1**	**Top-5**	**Top-1**	**Top-5**
1	ResNet-50	76%	93%	0%	0%
2	Inception w/adverasarial training	78%	94%	1.5%	5.5%
3	ResNet-50, BaRT trained	65%	85%	0%	0%
4	ResNet-50, BaRT trained & applied	54%	76%	16%	51%
5	ResNet-50, BaRT ensemble	65%	76%	16%	70%

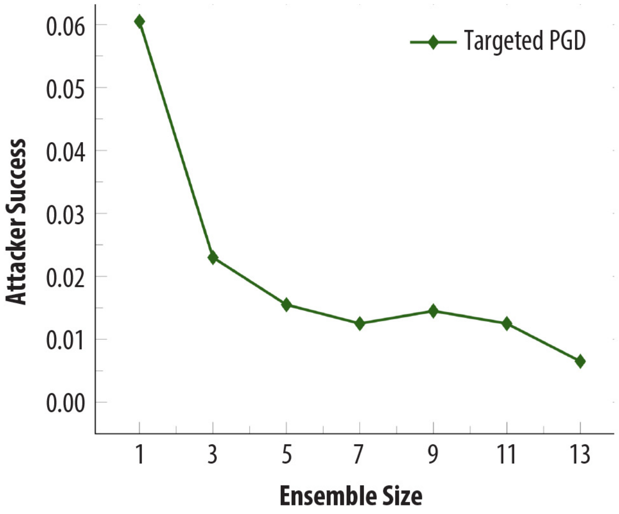


FIGURE 5. This graph shows the success rate of a targeted PGD attack for various BaRT ensemble sizes. For ensemble size 1 (i.e., a single BaRT), the success rate is near 6%, falling to below 1% with an ensemble size of 13.

Conclusion

In this article we have described the threats of adversarial attacks to machine learning models. With machine learning models driving more and more decisions in our economy and our devices, manipulation of those models by attackers is a real threat.

We have given a high-level description of our new approach, called BaRT, which demonstrates the use of randomness to build more effective defenses against adversarial attack. It is a significant improvement over what came before. Hopefully the strategy of using randomness will be useful to others who are working to make classifiers resistant to attack.

For nontargeted attacks, our top result achieves the correct class in the top-5 70% of the time while under attack. For this benefit, we lose approximately 7% in top-5 accuracy while not under attack. This is a major improvement over prior work. Perhaps more significantly, in the targeted attack case, our top result reduces success of attacks to below 1%.

However, we do not claim that this approach is perfect. There is more work to be done. We do not yet have mathematical proofs supporting this approach, only intuition and empirical testing. Our nontargeted result still leaves much room for improvement in accuracy, in both no-attack and attack situations.

Also, it is not a defense that can be applied immediately to other problem domains such as audio data or malware detection. In each domain we will need to carefully consider what transformations might be available that preserve the salient structure of the information necessary for classification, and what can be randomized in ways that interfere with attacks but have a manageable cost in accuracy.

Research in adversarial machine learning is very active. It is virtually guaranteed that new attack techniques will emerge in the near future. If BaRT is considered the current best defense, many researchers may focus on figuring out ways to defeat it. We look forward to this and hope this work will stimulate advances in both attacking and defending models, as both increase our understanding of the underlying fundamentals. Ultimately, we hope that advances will enable us to build machine learning systems that are safer, more accurate, more reliable, and more trustworthy.

Return to CONTENTS

References

[1] Biggio B, Roli F. "Wild patterns: Ten years after the rise of adversarial machine learning." *Pattern Recognition.* 2018;84:317–331. Available at: https://doi.org/10.1016/j.patcog.2018.07.023.

[2] Carlini N, Wagner D. "Adversarial examples are not easily detected: Bypassing ten detection methods." In: *AlSec '17: Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security.* New York (NY): ACM; 2017. pp. 3–14. Available at: https://doi.org/10.1145/3128572.3140444.

[3] Eykholt K, Evtimov I, Fernandes E, Li B, Rahmati A, Xiao C, Prakash A, Kohno T, Song D. "Robust physical-world attacks on deep learning models." 2018. Cornell University Library, arXiv: 1707.08945v5.

[4] National Science & Technology Council, Select Committee on Artificial Intelligence. "The national artificial intelligence research and development strategic plan: 2019 update," 2019 Jun. Available at: https://www.nitrd.gov/pubs/National-AI-RD-Strategy-2019.pdf.

[5] Nelson B, Rubinstein BIP, Huang L, Joseph AD, Lee SJ, Rao S, Tygar JD. "Query strategies for evading convex-inducing classifiers." *Journal of Machine Learning Research.* 2012;13(1):1293–1332.

[6] Hong S, Davinroy M, Kaya Y, Dachman-Soled D, Dumitras T. "How to own the NAS in your spare time." 2020. Cornell University Library, arXiv: 2002.06776. Accepted to International Conference on Learning Representations (ICLR), 2020.

[7] Wong E, Schmidt F, Metzen JH, Kolter JZ, "Scaling provable adversarial defenses." In: Bengio S, Wallach H, Larochelle H, Grauman K, Cesa-Bianchi N, Garnett R, editors. *Advances in Neural Information Processing Systems 31.* 2018. pp. 8400–8409.

[8] Athalye A, Carlini N, Wagner D. "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples." 2018. Cornell University Library, arXiv:1802.00420.

[9] Goodfellow IJ, Shlens J, Szegedy C. "Explaining and harnessing adversarial examples." 2015. Cornell University Library, arXiv:1412.6572v3.

[10] Kurakin A, Goodfellow I, Bengio S. "Adversarial machine learning at scale." 2017. Cornell University Library, arXiv: 1611.01236v2.

[11] Madry A, Makelov A, Schmidt L, Tsipras D, Vladu A. "Towards deep learning models resistant to adversarial attacks." In: *International Conference on Learning Representations (ICLR);* 2018. Available at: https://openreview.net/forum?id=rJzIBfZAb.

[12] ImageNet.org. About ImageNet. Web page, accessed 2020 Jan 23. Available at: http://www.image-net.org/about-overview.

[13] Xie C, Wu Y, van der Maaten L, Yuille AL, He K. "Feature denoising for improving adversarial robustness." In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR);* 2019; Long Beach, CA: pp. 501–509. Available at: https://doi.org/10.1109/CVPR.2019.00059.

[14] Raff E, Sylvester J, Forsyth S, McLean M. "Barrage of Random Transforms for adversarially robust defense." In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR);* 2019; Long Beach, CA: pp. 6521-6530. Available at: https://doi.org/10.1109/CVPR.2019.00669.

[15] Athalye A, Engstrom L, Ilyas A, Kwok K. "Synthesizing robust adversarial examples." 2017. Cornell University Library, arXiv: 1707.07397.

[16] ImageNet. Large Scale Visual Recognition Challenge 2012 (ILSVRC2012). Web page, accessed 2020 Jan 24. Available at: http://image-net.org/challenges/LSVRC/2012/.

[17] He K, Zhang X, Ren S, Sun J. "Deep residual learning for image recognition." In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR);* 2016 Jun; Las Vegas, NV: pp. 770–778. Available at: https://doi.org/10.1109/CVPR.2016.90.

[18] Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z. "Rethinking the inception architecture for computer vision." In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR);* 2016 Jun; Las Vegas, NV. Available at: https://doi.org/10.1109/CVPR.2016.308.

National Security Agency Central Security Service

Defending Our Nation. Securing The Future.