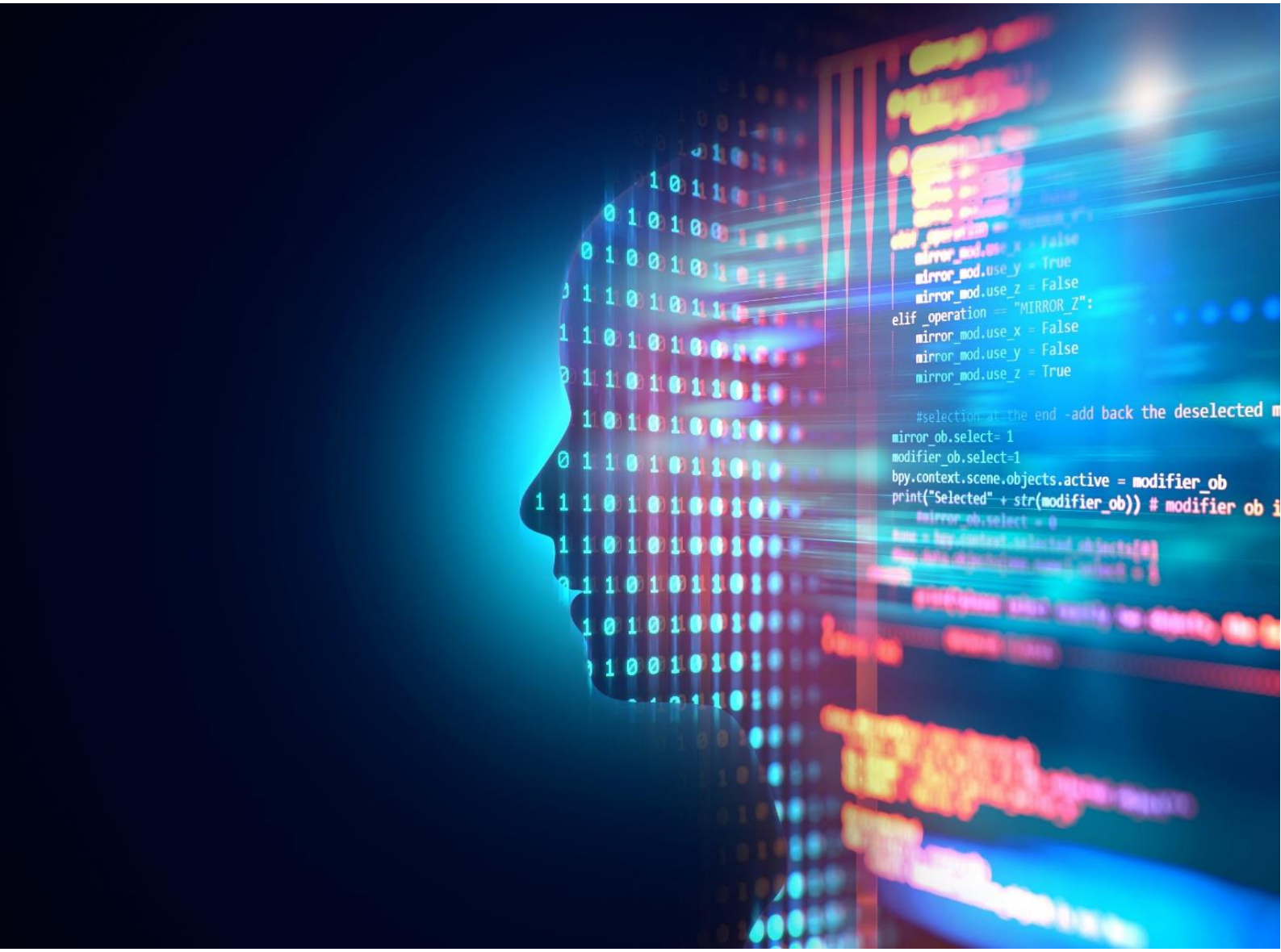
SHERPA

Security Issues, Dangers and Implications of Smart Information Systems



This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme Under Grant Agreement no. 786641

Authors

Andrew Patel, Researcher at Artificial Intelligence Center of Excellence, F-Secure Corporation (andrew.patel@f-secure.com)
Tally Hatzakis, Senior Research Analyst, Trilateral Research (tally.hatzakis@trilateralresearch.com)
Kevin Macnish, Assistant Professor at the University of Twente (k.macnish@utwente.nl)
Mark Ryan, Postdoctoral Researcher at the University of Twente (m.j.ryan@utwente.nl)
Alexey Kirichenko, Research Collaboration Manager, F-Secure Corporation (alexey.kirichenko@f-secure.com)

Table of Contents

Executive Summary

While recent innovations in the machine learning domain have enabled significant improvements in a variety of computer-aided tasks, machine learning systems present us with new challenges, new risks, and new avenues for attackers. The arrival of new technologies can cause changes and create new risks for society (Zwetsloot and Dafoe, 2019) (Shushman et al., 2019), even when they are not deliberately misused. In some areas, artificial intelligence has become powerful to the point that trained models have been withheld from the public over concerns of potential malicious use. This situation parallels to vulnerability disclosure, where researchers often need to make a trade-off between disclosing a vulnerability publicly (opening it up for potential abuse) and not disclosing it (risking that attackers will find it before it is fixed). As such, researchers should consider how machine learning may shape our environment in ways that could be harmful.

Machine learning will likely be equally effective for both offensive and defensive purposes (in both cyber and kinetic theatres), and hence one may envision an "AI arms race" eventually arising between competing powers. Machine-learning-powered systems will also affect societal structure with labour displacement, privacy erosion, and monopolization (larger companies that have the resources to fund research in the field will gain exponential advantages over their competitors).

The capabilities of machine learning systems are often difficult for the lay person to grasp. Some humans naively equate machine intelligence with human intelligence. As such, people sometimes attempt to solve problems that simply cannot (or should not) be solved with machine learning. Even knowledgeable practitioners inadvertently build systems that exhibit social bias due to the nature of the training data used. The first section of this report details common errors made while deploying and also designing and training machine learning models, provides some recommendations to avoid such pitfalls, and concludes with a discussion of the ethical implications of badly designed Smart Information Systems.

Data analysis and machine learning methods are powerful tools that can be used for both benign and malicious purposes. The second section of this report is a forward-thinking look at a number of primarily potential malicious uses of artificial intelligence, including intelligent automation, analytics, disinformation and fake news, phishing and spam, synthesis of audio, visual, and text content, and obfuscation.

As artificial-intelligence-powered systems become more prevalent, it is natural to assume that adversaries will learn how to attack them. Indeed, some machine-learning-based systems in the real world have been under attack for years already. The third section of this report provides step-by-step details of a number of popular attacks against machine-learning-based systems, and provides examples of how these attacks might be used maliciously. The section concludes with a discussion of related ethical issues.

Adversarial attacks against machine learning models are hard to defend against because there are very many ways for attackers to force models into producing incorrect outputs. Research into mitigations against commonly proposed attacks has proceeded hand-in-hand with studies on performing those attacks. The forth section of this report presents the reader with details of popular mitigation methods.

In an effort to remain competitive, companies or organizations may forgo ethical principles, ignore safety concerns, or abandon robustness guidelines in order to push the boundaries of their work, or to ship a product ahead of a competitor. This trend towards low quality, fast time-to-market is already prevalent in the Internet of Things (“Internet of things,” 2019) industry, and is considered highly problematic by most cyber security practitioners. Similar recklessness in the AI space could be equally negatively impactful. As such, AI researchers and engineers will need to be aware of the sorts of ethical issues they may encounter in their work and understand how to respond to them.

Introduction

Machine learning is the process of training an algorithm (model) to learn from data without the need for rules-based programming. In traditional software development processes, a developer hand-writes code such that a known set of inputs are transformed into desired outputs. With machine learning, an algorithm is iteratively configured to transform a set of known inputs into a set of outputs optimizing desired characteristics. Many different machine learning architectures exist, ranging from simple logistic regression(“Logistic regression,” 2019) to complex neural network architectures (sometimes referred to as "deep learning"(“Deep learning,” 2019)). Common uses of machine learning include:

- Classification – assigning a label (class) to an input (e.g. determining whether there is a dog in an image)
- Sequential – predicting a sequence (e.g. translating a sentence from English to French, predicting the next words in a sentence, the next notes in a musical sequence, or the next price of a stock)
- Policy – controlling an agent in an environment (e.g. playing a video game, driving a car)
- Clustering – grouping a number of inputs by similarity (e.g. finding anomalies in network traffic, identifying demographic groups)
- Generative – generating artificial outputs based on inputs the model was trained on (e.g. generating face images)

Methods employed to train machine learning models depend on the problem space and available data. Supervised learning techniques are used to train a model on fully labelled data. Semi-supervised learning techniques are used to train a model with partially labelled data. Unsupervised learning techniques are used to process completely unlabelled data. Reinforcement learning techniques are used to train agents to interact with environments (such as playing video games or driving a car).

Recent innovations in the machine learning domain have enabled significant improvements in a variety of computer-aided tasks, including:

- Image and video recognition, tagging, labelling, and captioning systems
- Speech-to-text and speech-to-speech conversion
- Language translation
- Linguistic analysis
- Text synthesis
- Chatbots and natural language understanding tasks
- Financial modelling and automated trading
- Image synthesis
- Content generation and artistic tools
- Image and video manipulation
- Game playing
- Self-driving vehicles
- Robot control systems
- Marketing analytics
- Recommendation systems and personal digital assistants
- Network anomaly detection

- Penetration testing tools
- Content categorization, filtering, and spam detection

Machine learning-based systems are already deployed in many domains, including finance, commerce, science, military, healthcare, law enforcement, and education. In the future, more and more important decisions will be made with the aid of machine learning. Some of those decisions may even lead to changes in policies and regulations. Hence it will be important for us to understand how machine learning models make decisions, predict ways in which flaws and biases may arise, and determine whether flaws or biases are present in finished models. A growing interest in understanding how to develop attacks against machine learning systems will also accompany this evolution, and, as machine learning techniques evolve they will inevitably be adopted by 'bad actors', and used for malicious purposes.

This document explores how flaws and biases might be introduced into machine learning models, how machine learning techniques might, in the future, be used for offensive or malicious purposes, how machine learning models can be attacked, and how those attacks can presently be mitigated. Machine learning systems present us with new challenges, new risks, and new avenues for cyberattackers. As such, this document will explore the implications of attacks against these systems and how they differ from attacks against traditional systems.

1. Bad Artificial Intelligence (AI)

If a machine learning model is designed or trained poorly, or used incorrectly, flaws may arise. Designing and training machine learning models is often a complex process, and there are numerous ways in which flaws can be introduced.

A flawed model, if not identified as such, can pose risks to people, organizations, or even society. In recent years, machine-learning-as-a-service (such as Amazon SageMaker ("Amazon SageMaker," n.d.), Azure Machine Learning Service ("Azure Machine Learning Service," n.d.), and Google Cloud Machine Learning Engine ("Cloud ML Engine," n.d.)) offerings have enabled individuals to train machine learning models on their own data, without the need for deep technical domain knowledge. While these services have lowered the barrier to adoption of machine learning techniques, they may have also inadvertently introduced the potential for widespread misuse of those techniques.

This section enumerates the most common errors made while designing, training, and deploying machine learning models. Common flaws can be broken into three categories - incorrect design decisions, deficiencies in training data, and incorrect utilization choices.

Flaws arising from design decisions

Machine learning models are essentially functions that accept a set of inputs, and return a set of outputs. It is up to a machine learning model's designer to select the features that are used as inputs to a model, such that it can be trained to generate accurate outputs. This process is often called feature engineering. If a designer of a model chooses features that are signal-poor (have

little effect on the decision that is made by the model), irrelevant (have no effect on the decision), or introduce bias (inclusion or omission of inputs and / or features that favour/disfavour certain results), the model's outputs will be inaccurate.

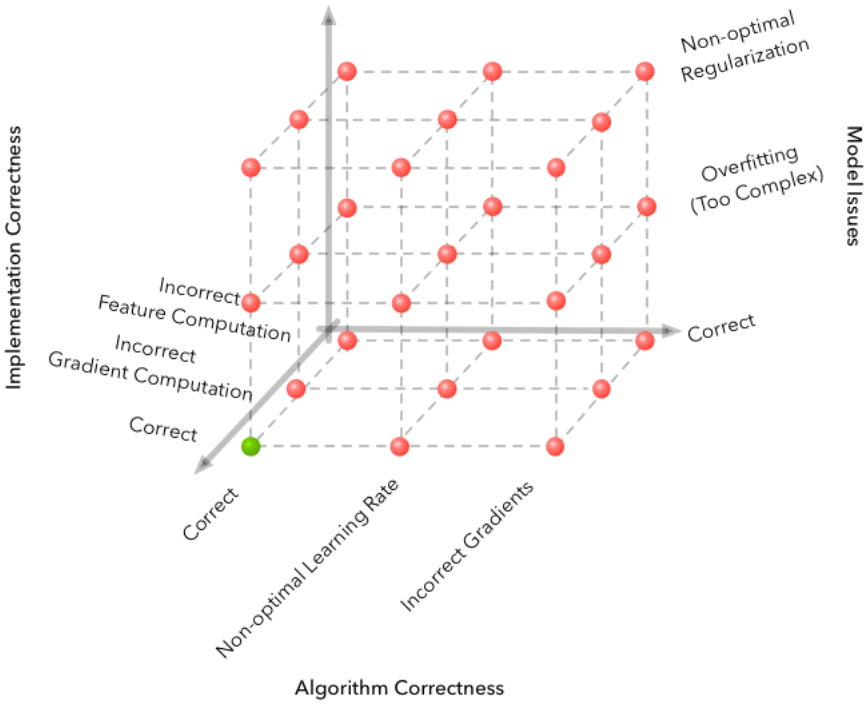
If features do not contain information relevant to solving the task at hand, they are essentially useless. For instance, it would be impossible to build a model that can predict the optimal colour for targeted advertisements with data collected from customer's calls for technical support. Unfortunately, the misconception that throwing more data at a problem will suddenly make it solvable is all too real, and such examples do occur in real life.

A good example of poor feature engineering can be observed by examining online services designed to determine whether Twitter users are fake or bots (such as botornot.co). These services are based on machine learning models whose features are derived only from the data available from a Twitter account's "user" object (the length of the account's name, the date the account was created, how many Tweets the account has published, how many followers and friends the account has, and whether the account has set a profile picture or description). These input features are relatively signal-poor for determining whether an account is a bot, which often manifests in incorrect classification.

This is the information (circled) that botornot uses to determine whether an account is a bot. Or not.

Another common design flaw is inappropriately or suboptimally chosen model architecture and parameters. A potentially huge number of combinations of architectures and parameters are available when designing a machine learning model, and it is often impossible to try every possible combination. A common approach to solving this problem is to find an architecture that works best, and then use an iterative process, such as grid search or random search to narrow down the best parameters. This is a rather time-consuming process - in order to test each set of parameters, a new model must be trained - a process that can take hours, days, or even weeks. A designer who

is not well-practiced in this field may simply copy a model architecture and parameters from elsewhere, train it, and deploy it, without performing proper optimization steps.



An illustration of some of the design decisions available when building a machine learning model.

Overfitting

Incorrect choices in a model's architecture and parameters can often lead to the problem of overfitting, when a model learns to partition the samples it has been shown accurately, but fails to generalize on real-world data. Overfitting can also arise from training a model on data that contains only a limited set of representations of all possible inputs, which can happen even when a training set is large if there's a lack of diversity in that dataset. Problems related to training data will be discussed in greater detail in the next subsection.

Source: https://hackernoon.com/memorizing-is-not-learning-6-tricks-to-prevent-overfitting-in-machine-learning-820b091dc42

Overfitting can be minimized by architectural choices in the model - such as dropout (Budhiraja, 2016) in the case of neural networks. It can also be minimized by data augmentation - the process of creating additional training data by modifying existing training samples. For instance, in order to augment the data used to train an image classification model, you might create additional training

samples by flipping each image, performing a set of crops on each image, and brightening/darkening each image.

Flaws arising from deficient training data

It is common practice to evaluate a model on a separate set of samples after training (often called a test set). However, if the test set contains equally limited sample data, the trained model will appear to be accurate (until put into production). Gathering a broad enough set of training examples is often extremely difficult. However, model designers can iteratively test a model on real-world data, update training and test sets with samples that were incorrectly classified, and repeat this process until satisfactory real-world results are achieved. This process can be time-consuming, and thus may not always be followed in practice.

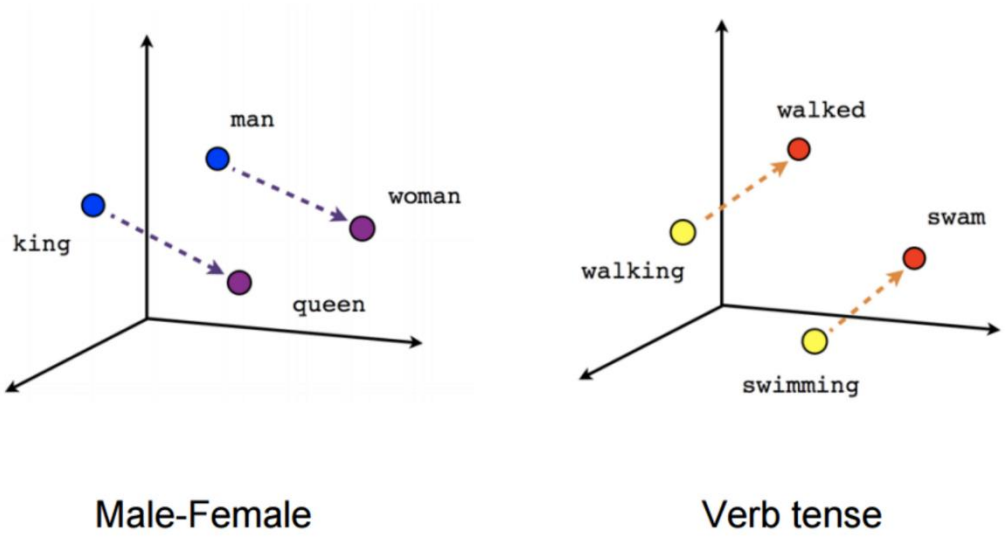
Supervised learning methods require a training set that consists of accurately labelled samples. Labelled data is, in many cases, difficult or costly to acquire - the process of creating a labelled set can include manual work by human beings. If a designer wishes to create a model using supervised learning, but doesn't have access to an appropriate labelled set of data, one must be created. Here, shortcuts may be taken in order to minimize the cost of creating such a set. In some cases, this might mean "working around" the process of manually labelling samples (i.e. blanket collection of data based on the assumption that it falls under a specific label). Without manually checking data collected in this way, it is possible that the model will be trained with mislabelled samples.

If a machine learning model is trained with data that contains imbalances or assumptions, the output of that model will reflect those imbalances or assumptions. Imbalances can be inherent in the training data, or can be "engineered" into the model via feature selection and other designers' choices. For example, evidence from the US ("Police warned about using algorithms," 2017) suggests that models utilized in the criminal justice system are more likely to incorrectly judge black defendants as having a higher risk of reoffending than white defendants. This flaw is introduced into their models both by the fact that the defendant's race is used as an input feature, and the fact that the historical data might excessively influence decision-making.

In another recent example, Amazon (Gershgorn, n.d.) attempted to create a machine learning model to classify job applicants. Since the model was trained on the company's previous hiring decisions, it led them to building a recruitment tool that reinforced their company's historical hiring policies. The model penalized CVs that included the word "women's", downgraded graduates from women's colleges, and highly rated aggressive language. It also highly rated applicants with the name "Jared" who had previously played lacrosse.

A further example of biases deeply embedded in historical data can be witnessed in natural language processing (NLP) tasks. The creation of word vectors is a common precursor step to other NLP tasks. Word vectors are usually more accurate when trained against a very large text corpus, such as a large set of scraped web pages and news articles (for example, the "Google News data" set). However, when running simple NLP tasks, such as sentiment analysis, using word vectors created in this manner, a bias in English-language news reporting becomes apparent. Simple experiments (Speer, 2017) (Bolukbasi et al., 2016) reveal that word vectors trained against

the Google News text corpus exhibit gender stereotypes to a disturbing extent (such as associating the phrase “computer programmer” to man and the word “homemaker” to woman).



Word vector examples. Source: https://towardsdatascience.com/word-embedding-with-word2vec-and-fasttext-a209c1d3e12c

The idea that bias can exist in training data, that it can be introduced into models, and that biased models may be used to make important decisions in the future is the subject of much attention (Kleinman, 2018). Anti-bias initiatives already exist (such as AlgorithmWatch (“AlgorithmWatch,” n.d.), (Berlin), and Algorithm Justice League (“AJL -ALGORITHMIC JUSTICE LEAGUE,” n.d.), (US), and several technical solutions to identify and fix bias in machine learning models are now available (such as IBM's Fairness 360 kit, Facebook's Fairness Flow (Gershgorn, n.d.), an as-yet-unnamed tool from Microsoft (Knight, n.d.), and Google's "what if" (Wexler, n.d.) tool). Annual events are also arranged to discuss such topics, such as FAT-ML (Fairness, Accountability, and Transparency in Machine Learning) (“FAT ML,” n.d.). Groups from Google and IBM have proposed a standardized means of communicating important information about their work, such as a model’s use cases, a dataset’s potential biases, or an algorithm’s security considerations (Gebru et al., 2018) (Holland et al., 2018) (Mitchell et al., 2019).

AI is reportedly transforming many industries, including lending and loans (Hope, 2018), criminal justice (Tashea, 2017), and recruitment (Chaker, 2019). However, participants in a recent Twitter thread started by Professor Gina Neff (Neff, 2019) discussed the fact that imbalances in datasets is incredibly difficult to find and fix, given that it arises for social and organizational reasons, in addition to technical reasons. This was illustrated by the analogy that despite being technically rooted, both space shuttle accidents were ultimately caused by societal and organizational failures. The thread concluded that bias in datasets (and thus the machine learning models trained on those datasets) is a problem that no single engineer, company or even country can conceivably fix.

Flaws arising from incorrect utilization of a machine learning model

Machine learning models are very specific to the data they were trained on and, more generally, the machine learning paradigm has serious limitations. This is often difficult for humans to grasp –

their overly high expectations come from naively equating machine intelligence with human intelligence. For example, humans are able to recognize people they know regardless of different weather and lighting conditions. The fact that someone you know is a different colour under nightclub lighting, or is wet because they have been standing in the rain does not make it any more difficult for you to recognize them. However, this is not necessarily the case for machine learning models. It is also important to observe that modelling always involves certain assumptions, so applying a machine-learning-based model in situations when the respective assumptions do not hold will likely lead to poor results.

Going beyond the above examples, people sometimes attempt to solve problems that simply cannot (or should not) be solved with machine learning, perhaps due to a lack of understanding of what can and cannot be done with current methodologies.

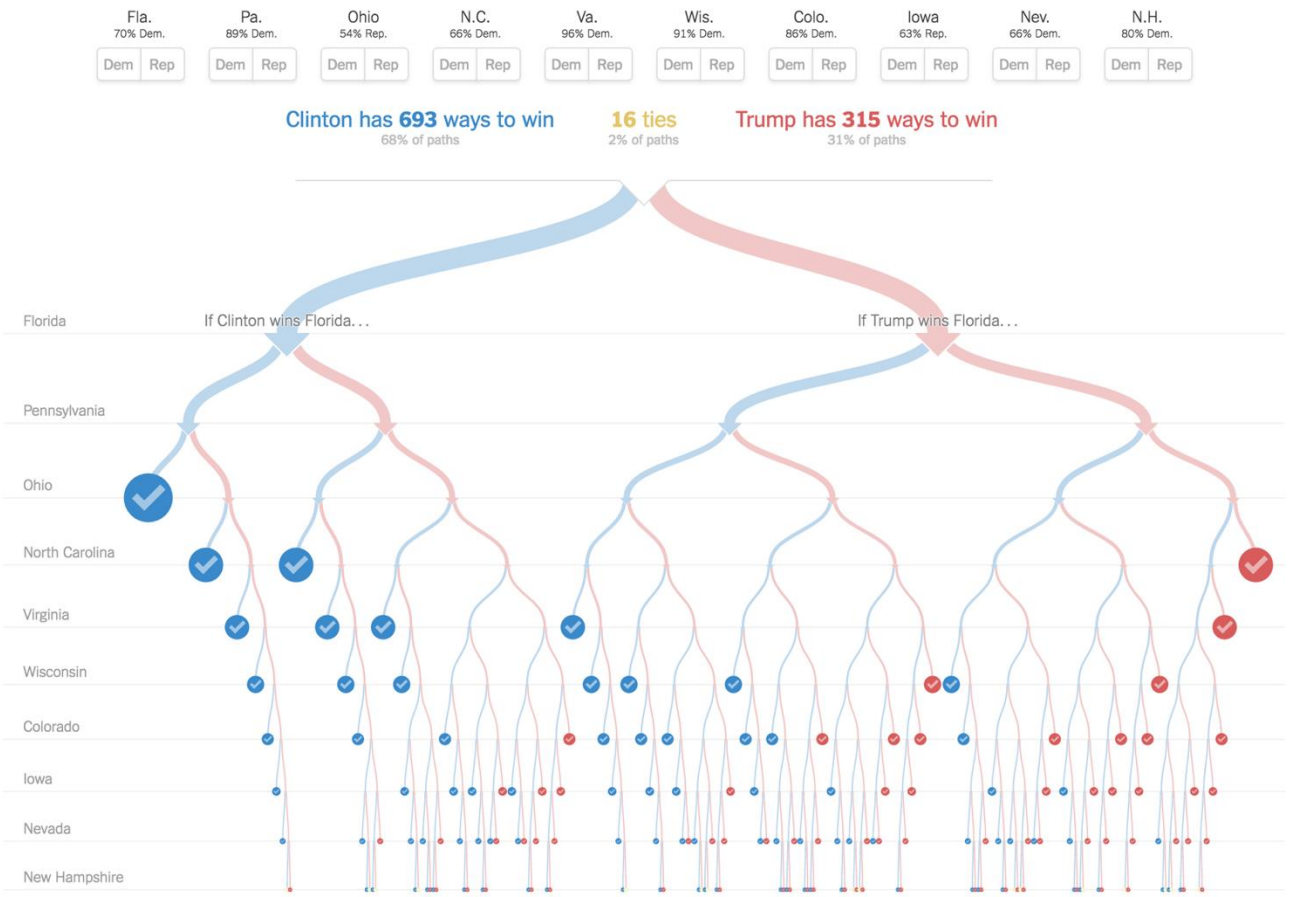
One good example of this is automated grading of essays, a task where machine learning with its current limitations should not be used at all. School districts in certain parts of the world (Riemer, n.d.) have however created machine learning models using historically collected data - essays, and the grades that were assigned to them. The trained model takes a new essay as input and outputs a grade. The problem with this approach is that the model is unable to understand the content of the essay (a task that is far beyond the reach of current machine learning capabilities), and simply grades it based on patterns found in the text - sentence structure, usage of fancy words, paragraph lengths, and usage of punctuation and grammar. In some cases, researchers have written tools ("BABEL Generator," n.d.) to generate nonsensical text designed to always score highly in specific essay grading systems.

What to keep in mind when planning, building and utilizing machine learning systems

The process of developing and deploying a machine learning model differs from standard application development in a number of ways. The designer of a machine learning model starts by collecting data or building a scenario that will be used to train the model, and writes the code that implements the model itself. The developer then runs a training phase, where the model is exposed to the previously prepared training data or scenario and, through an iterative process, configures its internal parameters in order to fit the model. Once training has ended, the resulting model is tested for the key task-specific characteristics, such as accuracy, recall, efficiency, etc. The output of training a machine learning model is the code that implements the model, and a serialized data structure that describes the learned parameters of that model. If the resulting model fails to pass tests, the model's developer adjusts its parameters and/or architecture and perhaps even modifies the training data or scenario and repeats the training process until a satisfactory outcome is achieved. When a suitable model has been trained, it is ready to be deployed into production. The model's code and parameters are plugged into a system that accepts data from an external source, processes it into inputs that the model can accept, feeds the inputs into the model, and then routes the model's outputs to intended recipients.

Depending on the type and complexity of a chosen model's architecture, it may or may not be possible for the developer to understand or modify the model's logic. As an example, decision trees ("Decision tree," 2019) are often highly readable and fully editable. At the other end of the spectrum, complex neural network architectures can contain millions of internal parameters, rendering them almost incomprehensible. Models that are readable are also explainable, and it

becomes much easier to detect flaws and bias in such models. However, these models are often relatively simple, and may be unable to handle more complex tasks. Tools exist to partially inspect the workings of complex neural networks, but finding bias and flaws in such models can be an arduous task that may often involve guesswork. Hence, rigorous testing is required to ensure the absence of potential flaws and biases. Testing a machine learning model against all possible inputs is impossible. In contrast, where an interface exists in traditionally built applications, defined processes and tools are available that enable developers to identify inputs that can catch all potential errors and corner cases.



An example of a decision tree. Source: https://lethalbrains.com/learn-ml-algorithms-by-coding-decision-trees-439ac503c9a4

Machine learning models receive inputs that have been pre-processed and then vectorized into fixed-size structures. Vectorization is the process of converting an input (such as an image, piece of text, audio signal, or game state) into a set of numerical values, often in the form of an array, matrix (two-dimensional array), or tensor (multi-dimensional array). Bugs may be introduced into the code that interfaces the model with external sources or performs vectorization; these may find their way in via code invoking machine learning methods implemented in popular libraries, or may be introduced in decision-making logic. Detecting such bugs is non-trivial.

Based on SHERPA partners' experiences and knowledge gained while working in the field, we recommend following these guidelines while planning, building and utilizing machine learning models, so that they function correctly and do not exhibit bias:

Understand the problem domain

- Familiarize yourself with basic guidelines and practices in the field of machine learning.
- Understand how different machine learning techniques work, how they can be used, and what their limitations are.
- Understand your problem domain and whether the problem is even possible to solve using machine learning techniques.
- Research and read up on similar work in your problem area. Understand the methodologies that were used to solve the problem. Pay close attention to any experiments detailed in the research, and how they were conducted.

Prepare your training data

- Understand that a lot of published work is based on standard, well-labelled academic datasets. If your model requires a training set that is not one of these, understand the steps that will be required to create a good training set for your purposes.
- Evaluate whether the inputs you have available to you are relevant to the task you wish to accomplish.
- If you need to create your own labelled set, propose methods to accurately label the dataset, and to validate the accuracy of the labels.
- If your process includes choosing features for a model's input, think about whether those features might contribute to social bias (e.g. the use of race, gender, religion, age, country of origin, home address, area code, etc. as inputs). If you do choose a feature that is known to introduce social bias, be prepared to explain why that input is relevant to your process, and why it won't introduce social bias.

Design your model

- Start by prototyping your own model based on an existing model that was used for similar purposes. Experiment with changing the model's architecture and parameters during prototyping. Get a feeling for the amount of training that might be required across your own dataset, and the accuracy and other important model characteristics you might achieve, based on your prototypes.
- Check your prototype models early against real-world data, if possible. Start iteratively improving your training set along with your model.
- Once you've settled on an architecture, inputs, and a well-rounded training set, use automation to explore model parameters (such as random search).
- Check for overfitting. If it is a problem, try to understand what is causing it, and take appropriate measures to alleviate it.

Implement production processes

- Use a bias detection framework or develop your own methodology to explore potential bias and accuracy issues on your trained model, during development, to pinpoint and fix issues. Be prepared to provide details on the steps taken to remove bias and inaccuracies from your model.
- Have defined processes in place to quickly fix any issues found in your model.
- Consider implementing a process that can allow third parties to audit your model.
- Strongly consider implementing mechanisms that enable your model to explain how it made each decision. Note that explainability can sometimes trade-off with model quality, so care should be taken.
- If you're doing work in the NLP domain, check for biases that might be introduced by word vectors. Consider using unbiased word vectors such as those being developed in projects such as ConceptNet.

The above guidelines do not include measures that designers might want to take to safeguard machine learning models from adversarial attacks. Adversarial attack techniques and mitigations against them are discussed in later sections of this document.

It is worth noting that design decisions made at an early stage of a model's development will affect the robustness of the systems powered by that model. For instance, if a model is being developed to power a facial recognition system (which is used in turn to determine access to confidential data), the model should be robust enough to differentiate between a person's face and a photograph. In this example, the trade-off between model complexity and efficiency must be considered at this early stage.

Some application areas may also need to consider the trade-off between privacy and feature set. An example of such a trade-off can be illustrated by considering the design of machine learning applications in the cyber security domain. In some cases, it is only possible to provide advanced protection capabilities to users or systems when fine-grained details about the behaviour of those users or systems are available to the model. If the transmission of such details to a back-end server (used to train the model) is considered to be an infringement of privacy, the model must be trained locally on the system that needs to be protected. This may or may not be possible, based on the resources available on that system.

Ethical consequences of flaws in machine learning model design and utilization

In their paper, *The ethics of algorithms: Mapping the debate*, Brent Mittelstadt et al. identify six ethical concerns that can arise through the use of machine learning-based algorithms. These are summarised as: inconclusive evidence; inscrutable evidence; misguided evidence; unfair outcomes; transformative effects; and traceability (2016).

Ethical concerns - Machine learning-based algorithms

These provide a helpful framework for understanding ethical issues that can arise from the poor use of machine learning algorithms as outlined above. The first three areas (inconclusive evidence; inscrutable evidence; misguided evidence) are described by the authors as epistemic concerns (referring to how knowledge is obtained in machine learning), while the latter three (unfair outcomes; transformative effects; and traceability) are normative (implying or creating a particular

standard or norm). Nonetheless, all six have normative implications, some of which have been raised above.

The challenge presented by *inconclusive evidence* is that algorithms are rarely meant to be infallible and yet are often treated as if they were. This is related to the natural limitations of machine learning-based approaches (and modelling in general), and can be seen in cases where correlation is taken to be sufficient to direct action even though there is no established causal connection. Hence the possible existence of a confounding variable is not entertained, with the result that actions which have potentially significant consequences on people's lives may be enacted without due cause (Hildebrandt, 2011; Hildebrandt and Koops, 2010; Mayer-Schonberger and Cukier, 2017; Zarsky, 2016). A related problem here is the need for algorithms to deal with categories rather than individuals. As individuals are sorted into categories, this can indicate a degree of certainty which is not present, along with discouraging "alternative explorations, and create[ing] coherence among disparate objects,'' (Ananny, 2016, p. 103; see also Barocas, 2014). These can then lead to a misplaced faith in the reliability of the system, despite the system's approach of simplifying and classifying often subtly different individuals.

One problem related to the challenge of inconclusive evidence is the aforementioned issue of incorrect utilization of evidence. This may happen when a particular behaviour is the target of identification and yet the system is incapable of detecting that behaviour as such. Instead, the system is designed to measure what is measurable and then interpret that as evidence regarding the targeted behaviour. For example, loitering or intending to steal a vehicle both imply intent, which is invisible to an automated system. However, the period of time a person remains within a restricted radius (which may fall within the radius of a stationary vehicle) can be measured. As such, people who do not move outside a particular radius over a set period of time may be (incorrectly) identified as loitering or intending to steal a vehicle.

The second epistemic concern is that of *inscrutable evidence*, arising from a lack of transparency which is, in itself, a direct result of the fact that algorithms are frequently opaque (Tutt, 2016; see also Burrell, 2016). The problem is connected to the issues of explainability and related trade-offs discussed in the previous section. While transparency is not a panacea for ethical issues (as noted by, among others, Crawford, 2016; Neyland, 2016; Raymond, 2014), it is typically a precursor for any resolution to take place. Without knowing what is happening, it is difficult to resolve any problems. Yet as Mittelstadt et al. point out,

> "the primary components of transparency are accessibility and comprehensibility of information. Information about the functionality of algorithms is often intentionally poorly accessible. Proprietary algorithms are kept secret for the sake of competitive advantage (Glenn and Monteith, 2014; Kitchin, 2017; Stark and Fins, 2013), national security (Leese, 2014), or privacy. Transparency can thus run counter to other ethical ideals, in particular, the privacy of data subjects and autonomy of organisations" (Mittelstadt et al., 2016, p. 6).

The third epistemic concern raised by Mittlestadt et al. is that of *misguided evidence*. This refers to problems in understanding how bias can enter algorithmic decision-making. A lack of understanding here underpins a (misguided) sense of faith in the algorithms having a lack of bias (Bozdag, 2013; Naik and Bhide, 2014). Nonetheless, there is significant evidence to demonstrate that this perception is false and that algorithms, as a product of human design, do contain bias (e.g. Bozdag, 2013; Kraemer et al., 2011; Macnish, 2012; Newell and Marabelli, 2015, p. 6). As

Mittelstadt et al. point out, “an algorithm’s design and functionality reflects the values of its designer and intended uses, if only to the extent that a particular design is preferred as the best or most efficient option. Development is not a neutral, linear path; there is no objectively correct choice at any given stage of development, but many possible choices (Johnson, 2006). As a result, ‘the values of the author [of an algorithm], wittingly or not, are frozen into the code, effectively institutionalising those values’ (Macnish, 2012, p. 158)” (Mittelstadt et al., 2016, p. 7). As was discussed earlier, social biases can arise also from imbalances in training data.

Examples of misguided evidence abound. Significant cases of note are the automated soap dispenser which responded to white skin but not black (Fussell, 2017), and the measurement of potholes in Boston. In the former case, the soap dispenser had clearly been designed by and tested on only people with lighter coloured skin. It was not until the dispenser was installed that people started to notice that it would not respond to people with darker skin. The case of potholes in Boston relates to a decision to make an app available to people with smartphones and use the phone’s accelerometer to measure whenever a pothole was encountered. In this case, the flaw in thinking (that, certainly at the time, significantly fewer people in lower socio-economic brackets owned a smartphone than in higher brackets) was recognized before implementation (Crawford, 2013). Had it not been, potholes in wealthier areas of the city would have been recognized and resolved faster than elsewhere. Even when efforts are made to find diverse datasets on which to base and test an algorithm, those datasets may not be available. The SUBITO (Surveillance of Unattended Baggage and Identification and Tracking of its Owner) project considered how to identify people walking together. To do this, the project drew on a dataset of students at the University of Edinburgh. However, the final product was intended for distribution on an international scale where cultural diversity and associated behaviour was likely to be very different from that at one British university (Macnish, 2012).

The challenge of bias in the algorithm itself highlights the importance of human interpretation of algorithmic results. The results are not self-interpreting. However, this leads to the problem that interpreters come to apparently objective conclusions which in fact reflect their own ‘‘unconscious motivations, particular emotions, deliberate choices, socio-economic determinations, geographic or demographic influences’’ (Hildebrandt, 2011, p. 376). As such, a bias that has become embedded (“frozen”) into the code may be undetectable to some, or even taken as evidence of the system’s strength by others. If this problem of interpretation is coupled with that of inconclusive evidence (above) then a human operator overseeing an automated system may be more ready to ignore a white person standing in the vicinity of a stationary vehicle than a black person, if that operator’s prejudices (whether conscious or not) are such that they see white people as less likely to steal vehicles than black people.

Moving to the three normative areas of ethical concern, Mittlestadt et al. start with the problem of *unfair outcomes*. Here the authors identify the key issue as being that of profiling which “is frequently cited as a source of discrimination” (2016, p. 8). Profiling algorithms identify correlations and make predictions about behaviour at a group-level, albeit with groups (or profiles) that are constantly changing and re-defined by the algorithm” (Zarsky, 2013). Attempts have been made to avoid consideration of certain aspects which may contribute to discrimination (e.g. gender or ethnicity) (Calders et al., 2009; Calders and Kamiran, 2010; Schermer, 2011), but these have proven to be elusive to attempts to insert them into an automated process. Even apparently neutral characteristics may inadvertently overlap with other datasets to indicate

ethnicity, gender, sexual preference and other areas frequently used as means to discriminate (Macnish, 2012; Schermer, 2011).

The second area of normative concern is that of *transformative effects*, which impact both autonomy and privacy. Here it is recognized that the existence and use of the algorithm can transform the manner in which each of these values is approached. In the case of autonomy (an issue arising, e.g., in connection to recommender systems and personal digital assistants), filtering information to the individual may enable that person to focus more effectively on salient information, but at the same time risks the emergence of a "filter bubble" in which one only encounters information that already plays to one's own prejudices (Bozdag, 2013; Newell and Marabelli, 2015; Zarsky, 2016). In the case of privacy, as noted above, transparency of algorithmic determinations is seen as a precursor to ethical analysis, and yet where those algorithms deal with personal data there is a risk that transparency will lead to privacy being violated, or at least diminished (Hildebrandt, 2011; Van Wel and Royakkers, 2004).

Finally, the third area of normative concern is that of *traceability*, which relates to attributions of responsibility and blame. However, the problems of "many hands" – that there is rarely one single designer but rather a team of designers each with their own biases and the overall values of the team itself (Sandvig et al., 2014) – and the aforementioned opacity render the traceability of decisions and apportioning responsibility difficult, complicating trouble-shooting. A further complication in the case of machine-learning-based systems is the dependence of decision logic on training data.

In addition to these concerns are broader ethical issues which, while not restricted to Smart Information Systems (SIS), are pertinent to the cyber world more generally. As advanced levels of computer use become ubiquitous, a challenge is posed between the levels of technical knowledge required to operate a system and the technical capacities of the user.

As SIS is run on large data sets, an increase in the use of SIS implies an increase in the numbers of ways in which these data sets are used. Where those data are related to, for example, healthcare, the ethical issues involved relate predominantly to intellectual property and the security of businesses. Where the data are related to people, then the harm which has the potential to arise from those data becomes more personal. In all these cases, we can clearly see a need to handle data with care.

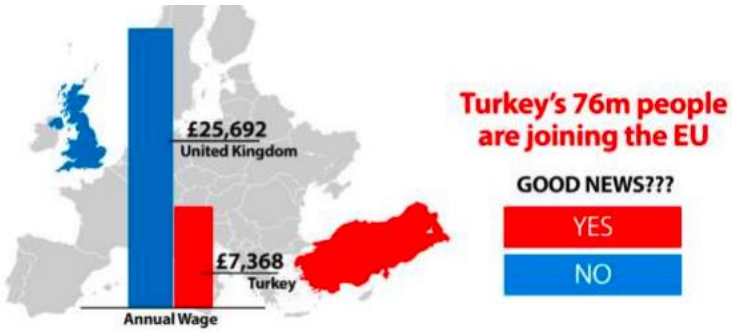
Finally, it is noteworthy that no SIS operates in a vacuum. Those installing, maintaining and operating SIS work under time constraints and budgetary limitations. This means that decisions need to be prioritised and, almost invariably, some methods of SIS will not be enacted, or will be enacted poorly owing to competing demands.

2. Malicious use of AI

Introduction

The tools and resources needed to create sophisticated machine learning models have become readily available over the last few years. Powerful frameworks for creating neural networks are freely available, and easy to use. Public cloud services offer large amounts of computing resources at inexpensive rates. More and more public data is available. And cutting-edge techniques are freely shared - researchers do not just communicate ideas through their publications nowadays – they also distribute code, data, and models (Shushman et al., 2019). As such, many people who were previously unaware of machine learning techniques are now using them.

Organizations that are known to perpetuate malicious activity (cyber criminals, disinformation organizations, and nation states) are technically capable enough to verse themselves with these frameworks and techniques, and may already be using them. For instance, we know that Cambridge Analytica used data analysis techniques in order to target specific Facebook users with political content via Facebook's targeted advertising service (a service which allows ads to be sent directly to users whose email addresses are already known). This simple technique proved to be a powerful political weapon. At the time of writing, it was still being used by pro-leave Brexiteer campaigners, to drum up support for a no-deal Brexit scenario (Geoghegan, 2019).



An example of a targeted adverts sent to users of Facebook during the UK referendum in 2016. Source: https://www.joe.co.uk/news/brexit-facebook-adverts-192164

As the capabilities of machine-learning-powered systems evolve, we will need to understand how they might be used maliciously. This is especially true for systems that can be considered dual-use ("Dual-use technology," 2019). The AI research community should already be discussing and developing best practices for distribution of data, code, and models that may be put to harmful use. Some of this work has already begun with efforts such as RAIL (Responsible AI Licenses) ("Responsible AI Licenses (RAIL)," n.d.).

This section suggests some forward-thinking examples of the potential malicious use of machine learning.

Intelligent automation

Machine learning methodologies have significant potential in the realm of offensive cyber security (a proactive and adversarial approach to protecting computer systems, networks and individuals from cyber attacks.) Password-guessing suites have recently been improved with Generative Adversarial Network (GAN) techniques ("hashcat - advanced password recovery," n.d.), fuzzing tools now utilize genetic algorithms ("american fuzzy lop," n.d.) to generate payloads, and web penetration testing tools have started to implement reinforcement learning methodologies (takaesu, 2019). Offensive cyber security tools are a powerful resource for both 'black-' and 'white hat' hackers. While advances in these tools will make cyber security professionals more effective in their jobs, cyber criminals will also benefit from these advances. Better offensive tools will enable more vulnerabilities to be discovered and responsibly fixed by the white hat community. However, at the same time, black hats may use these same tools to find software vulnerabilities for nefarious uses.

Intelligent automation will eventually allow current "advanced" CAPTCHA prompts to be solved automatically (most of the basic ones are already being solved with deep learning techniques). This will lead to the introduction of yet more cumbersome CAPTCHA mechanisms, hell-bent on determining whether or not we are robots.

The future of intelligent automation promises a number of potential malicious applications:

- Swarm intelligence capabilities might one day be added to botnets to deliver optimized DDoS attacks ("Denial-of-service attack," 2019) and spam campaigns, and to automatically discover new targets to infect.
- Malware of the future may be designed to function as an adaptive implant - a self-contained process that learns from the host it is running on in order to remain undetected, search for and classify interesting content for exfiltration, search for and infect new targets, and discover new pathways or methods for lateral movement.
- A report published in February, 2019 by ESET (Jánošík, 2019) claimed that the Emotet malware exhibited behaviour that would be difficult to achieve without the aid of machine learning. The author explained that, because different types of infected hosts received different payloads (in particular, to prevent security researchers from analysing the malware), the malware's authors must have developed some sort of machine learning logic to decide which payload each victim received. From these claims, one might imagine that Emotet's back ends employ host profiling logic that is derived by clustering a set of features received from connecting hosts, assigning labels to each identified cluster, and then deploying specific payloads to each machine, based on its cluster label. Even though it is more likely that Emotet's back ends simply use hand-written rules to determine which payloads each infected host receives, this story illustrates a practical, and easy to implement use of machine learning in malicious infrastructure.
- Futuristic end-to-end models could be designed to learn optimal strategies for the automated generation of efficient, undetectable poisoning attacks against search engines, recommenders, anomaly detection systems and federated learning systems.

Analytics, disinformation, and fake news

Data analysis and machine learning methods can be used for both benign and malicious purposes. Analytics techniques used to plan marketing campaigns can be used to plan and implement effective regional or targeted spam campaigns. Data freely available on social media platforms can be used to target users or groups with scams, phishing, or disinformation. Data analysis techniques can also be used to perform efficient reconnaissance and develop social engineering strategies against organizations and individuals in order to plan a targeted attack.

The potential impact of combining powerful data analysis techniques with carefully crafted disinformation is huge. Disinformation now exists everywhere on the Internet and remains largely unchecked. The processes required to understand the mechanisms used in organized disinformation campaigns are, in many cases, extremely complex. After news of potential social media manipulation of opinions during the 2016 US elections (Scott, 2018), the 2016 UK referendum on Brexit (Mayer, 2018), and elections across Africa (Solomon, 2018) (Plaut, 2018) (International, 2017), and Germany (Reigstad, 2017) many governments are now worried that well-organized disinformation campaigns may target their voters during an upcoming election. Election meddling via social media disinformation is common in Latin American countries (Gallagher, 2019a) (Gallagher, 2019b) (Gallagher, 2017) (Broderick, 2018). However, in the west, disinformation on social media and the Internet is no longer solely focused on altering the course of elections – it is about creating social divides, causing confusion, manipulating people into having more extreme views and opinions, and misrepresenting facts and the perceived support that a particular opinion has. (Lapowsky, 2018)

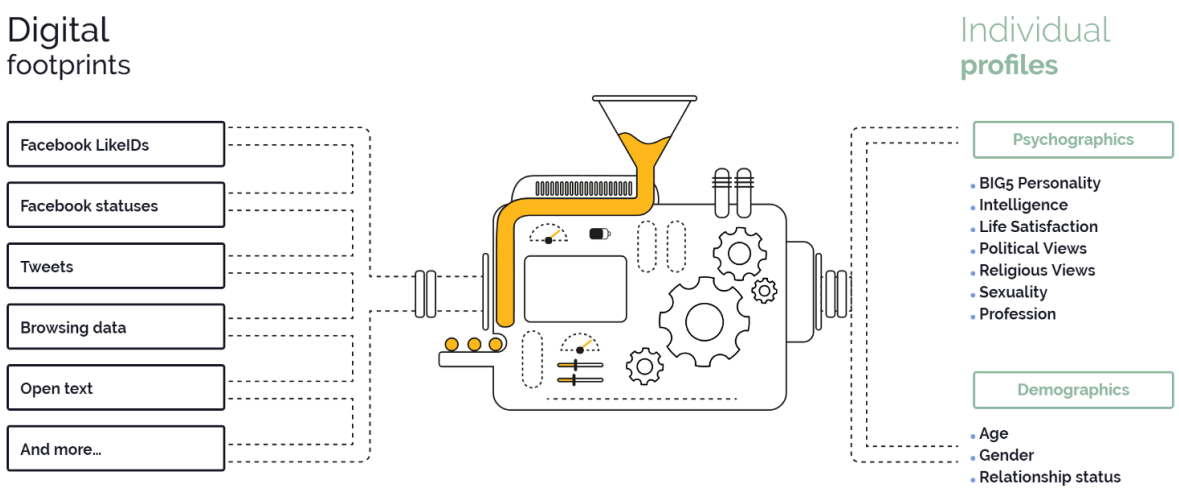
```
52. The idea is to create one liners that we can memify and mass produce. These need to appeal to emotion strongly. We have to
    literally be the hate machine we're known as.
53. ============================================================
54. Some angles to consider:
55. * Hill Racism quotes "fucking nigger, kike, fucking retards" <---- EXTREMELY POWERFUL
56. * CF Corruption
57. * Hill/Bill Corruption
58. * Rapist Bill + Rape Plane + Air Fuck One + Pedo Island
59. * "Hillary Loves Rapists" -> link to Epstein
60. * Child molestation
61. * Human Trafficking
62. * Greed/Money
63. * Old/Sickly Bill and Hill
64. * Selling out our nation
65. * Selling favors to backwards islam
66. * Selling secrets
67. * Too big to jail
68. * War mongerer: responsible for iraq + libya
69.
70. ============================================================
71. We want to keep the memes SQUARE. Some should use the same fonts/colors/styles as her official campaign to co-opt her branding.
    Others should be intentionally poorly done so the bernouts and dindus spread them.
```

An example of training material published by the alt-right prior to the 2016 US Presidential elections. Source: https://medium.com/@erin_gallagher/advanced-meme-warfare-july-6-2016-cw-5f9287ef36cd

Social engineering campaigns run by entities such as the Internet Research Agency, Cambridge Analytica, and the far-right demonstrate that social media advert distribution platforms (such as those on Facebook) have provided a weapon for malicious actors which is incredibly powerful, and damaging to society. The disruption caused by these recent political campaigns has created

divides in popular thinking and opinion that may take generations to repair. Now that the effectiveness of these social engineering tools is apparent, what we have seen so far is likely just an omen of what is to come.

The disinformation we hear about is only a fraction of what is actually happening. It requires a great deal of time and effort for researchers to find evidence of these campaigns. Twitter data is open and freely available, and yet it can still be extremely tedious to find evidence of disinformation and sentiment amplification campaigns on that platform. Facebook's targeted ads are only seen by the users who were targeted in the first place. Unless those who were targeted come forward, it is almost impossible to determine what sort of ads were published, who they were targeted at, and what the scale of the campaign was. Although social media platforms now enforce transparency on political ads, the source of these ads must still be determined in order to understand what content is being targeted at whom.

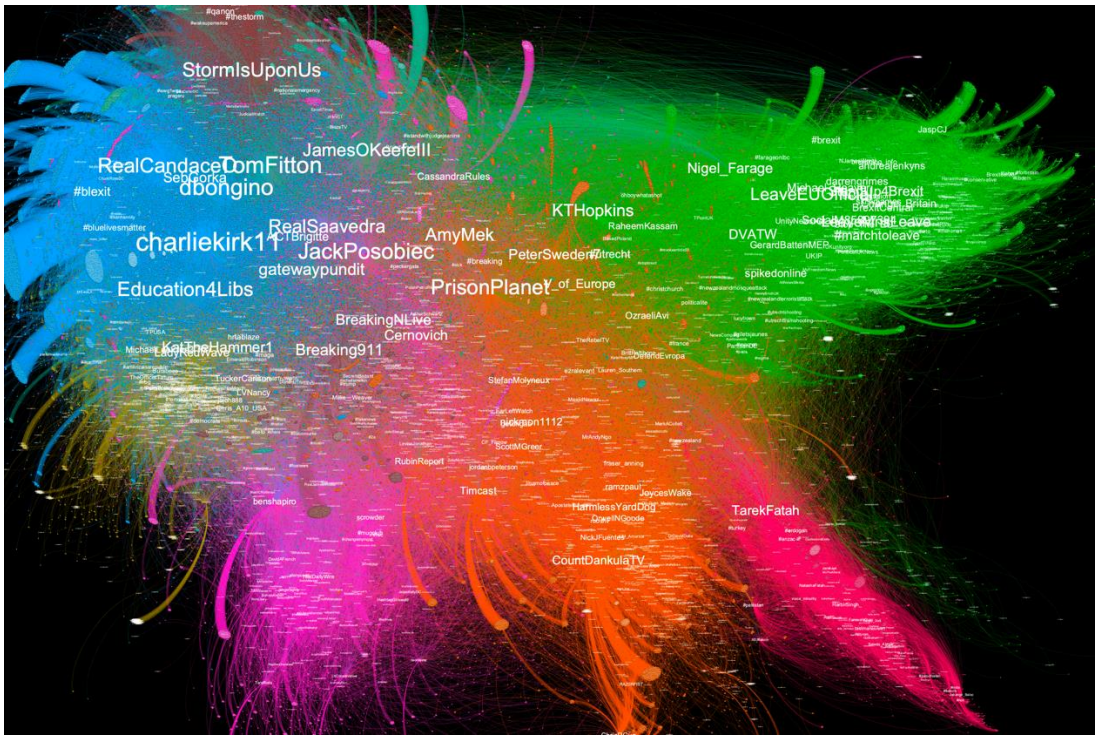


An example of the sort of data that is used in targeted political advertising. Source: https://medium.com/textifire/cambridge-analytica-microsofts-exploitative-ad-tech-c2db8633f542

Many individuals on social networks share links to "clickbait" headlines that align with their personal views or opinions, sometimes without having read the content behind the link. Fact checking can be cumbersome for people who do not have a lot of time. As such, inaccurate or fabricated news, headlines, or "facts" propagate through social networks so quickly that even if they are later refuted, the damage is already done (Britt et al., 2019). Fake news links are not just shared by the general public – celebrities and high-profile politicians may also knowingly (Dam, 2019) (Gallagher, 2019c) (Chaplain, 2019) or unknowingly share such content. This mechanism forms the very basis of malicious social media disinformation. A well-documented example of this was the UK's "Leave" campaign that was run before the Brexit referendum in 2016. Some details of that campaign are documented in the recent Channel 4 film: "Brexit: The Uncivil War" (“Brexit: The Uncivil War,” 2019). The problem is now so acute that in February, 2019 the Council of Europe published a warning about the risk of algorithmic processes being used to manipulate social and political behaviours (Europe, 2019).

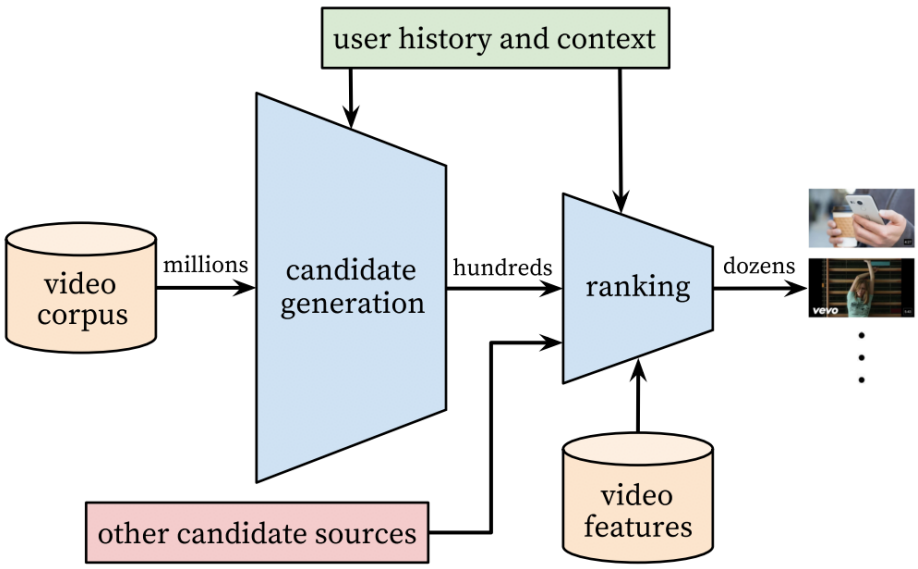
Despite what we know about how social media manipulation tactics were used during the Brexit referendum, multiple pro-Leave organizations are still funding social media ads promoting a "no deal" Brexit on a massive scale. The source of these funds, and the groups that are running these campaigns are not documented (Geoghegan, 2019).

A new pro-Leave UK astroturfing campaign, "Turning Point UK", funded by the far-right in both the UK and US, was kicked off in February 2019. It created multiple accounts on social media platforms to push its agenda (Cadwalladr, 2019) (Stuchbery, 2019). At the time of writing, right-wing groups are heavily manipulating sentiment on social media platforms in Venezuela (Gallagher, 2019d). Across the globe, the alt-right continues to manipulate social media, and artificially amplify pro-right-wing sentiment. For instance, in the US, multitudes of high-volume #MAGA (Make America Great Again) accounts amplify sentiment (Gallagher, 2019e). In France, at the beginning of 2019 a pro-LePen #RegimeChange4France hashtag amplification push was documented on Twitter, clearly originating from agents working outside of France (Norteño, 2019). In the UK during early 2019, a far-right advert was promoted on YouTube. This five-and-a-half minute anti-Muslim video was unskippable (MacWhirter, 2019).



A node-edge graph of global far-right Twitter amplification captured in May 2019

During the latter half of 2018, malicious actors uploaded multiple politically motivated videos to YouTube, and amplified their engagement through views and likes. These videos, designed to evade YouTube's content detectors, showed up on recommendation lists for average YouTube users (Day, 2019).



How YouTube recommends videos. Source: https://towardsdatascience.com/how-youtube-recommends-videos-b6e003a5ab2f

Disinformation campaigns will become easier to run and more prevalent in coming years. As the blueprints laid out by companies such as Cambridge Analytica are followed, we might expect these campaigns to become even more widespread and socially damaging.

A potentially dystopian outcome of social networks was outlined in a blog post written by François Chollet in May 2018 (Chollet, 2018), in which he describes social media becoming a "Psychological Panopticon". The premise for his theory is that the algorithms that drive social network recommendation systems have access to every user's perceptions and actions. Algorithms designed to drive user engagement are currently rather simple, but if more complex algorithms (for instance, based on reinforcement learning) were to be used to drive these systems, they may end up creating optimization loops for human behaviour, in which the recommender observes the current state of each target and keeps tuning the information that is fed to them, until the algorithm starts observing the opinions and behaviours it wants to see. In essence the system will attempt to optimize its users. Here are some ways these algorithms may attempt to 'train' their targets:

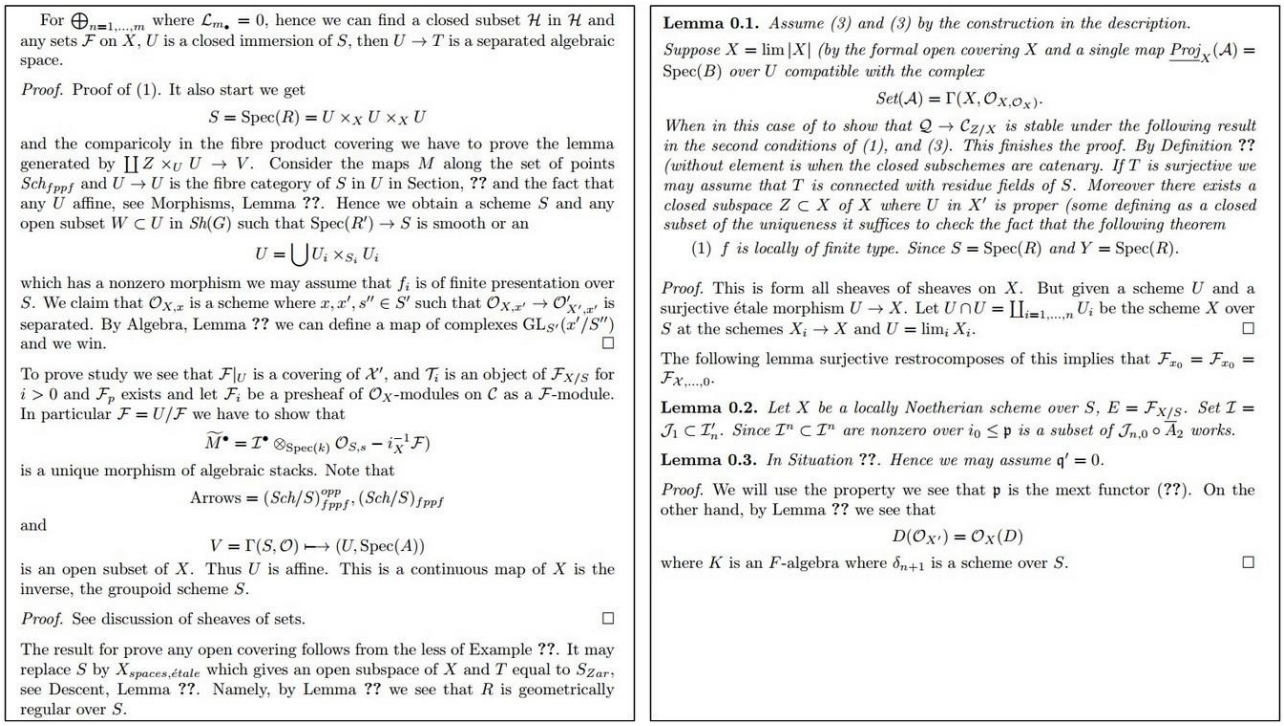
- The algorithm may choose to only show a target user content that it believes the user will engage or interact with, based on the algorithm's notion of the target's identity or personality. Thus, it will cause reinforcement of certain opinions or views in the target, based on the algorithm's own logic. (This is partially occurring already).
- If the target user publishes a post containing a viewpoint that the algorithm does not 'wish' the user to hold, it will only share it with users who would view the post negatively. The target will, after being flamed or down-voted enough times, stop sharing such views.
- If the target user publishes a post containing a viewpoint the algorithm 'wants' the user to hold, it will only share it with other users that view the post positively. The target will, after some time, likely share more of the same views.
- The algorithm may place a target user in an 'information bubble' where the user only sees posts from associates that share the target's views (and that are desirable to the algorithm).

- The algorithm may notice that certain content it has shared with a target user caused their opinions to shift towards a state (opinion) the algorithm deems more desirable. As such, the algorithm will continue to share similar content with the user, moving the target's opinion further in that direction. Ultimately, the algorithm may itself be able to generate content to those ends.

Chollet goes on to mention that, although social network recommenders may start to see their users as optimization problems, a bigger threat still arises from external parties gaming those recommenders in malicious ways. The data available about users of a social network can already be used to predict when a user is suicidal (Kwon, 2017), or when a user will fall in love or break up with their partner (Ferenstein, 2014), and content delivered by social networks can be used to change users' moods (Booth, 2014). We also know that this same data can be used to predict which way a user will vote in an election, and the probability of whether that user will vote or not.

If this optimization problem seems like a thing of the future, bear in mind that, at the beginning of 2019, YouTube made changes to its recommendation algorithms precisely because of problems it was causing for certain members of society. Guillaume Chaslot posted a Twitter thread in February 2019 (Chaslot, 2019) that described how YouTube's algorithms favoured recommending conspiracy theory videos, guided by the behaviours of a small group of hyper-engaged viewers. Fiction is often more engaging than fact, especially for users who spend substantial time watching YouTube. As such, the conspiracy videos watched by this group of chronic users received high engagement, and thus were pushed up by the recommendation system. Driven by these high engagement numbers, the makers of these videos created more and more content, which was, in-turn, viewed by this same group of users. YouTube's recommendation system was optimized to pull more and more users into chronic YouTube addiction. Many of the users sucked into this hole have since become indoctrinated with right-wing extremist views. One such user became convinced that his brother was a lizard, and killed him with a sword (Newlin, 2019). In February, 2019 the same algorithmic misgiving was found to have assisted the creation of a voyeur ring for minors on YouTube (Bergen et al., 2019) (Orphanides, 2019). Chaslot has since created a tool that allows users to see which of these types of videos are being promoted by YouTube (“algotransparency.org,” n.d.).

Between 2008 and 2013, over 120 bogus computer-generated papers were submitted, peer-reviewed, and published by the Springer and Institute of Electrical and Electronics Engineers (IEEE) organizations (Wiener-Bronner, 2014). These computer-generated papers were likely created using simple procedural methods, such as context-free grammars (“Context-free grammar,” 2019) or Markov chains (“Markov chain,” 2019). Text synthesis methods have matured considerably since 2013. A 2015 blog post by Andrej Karpathy (Karpathy, 2015) illustrated how recurrent neural networks can be used to learn from specific text styles, and then synthesize new, original text in a similar style. Andrej illustrated this technique with Shakespeare, and then went on to train models that were able to generate C source code, and Latex sources for convincing-looking algebraic geometry papers. It is entirely possible that these text synthesis techniques could be used to submit more bogus papers to IEEE in the future.



For $\bigoplus_{n=1,\ldots,m}$ where $\mathcal{L}_{m_\bullet} = 0$, hence we can find a closed subset $\mathcal{H}$ in $\mathcal{H}$ and any sets $\mathcal{F}$ on X, U is a closed immersion of S, then $U \to T$ is a separated algebraic space.

Proof. Proof of (1). It also start we get

$$S = \mathrm{Spec}(R) = U \times_X U \times_X U$$

and the comparicoly in the fibre product covering we have to prove the lemma generated by $\coprod Z \times_U U \to V$. Consider the maps M along the set of points Sch_{fppf} and $U \to U$ is the fibre category of S in U in Section, ?? and the fact that any U affine, see Morphisms, Lemma ??. Hence we obtain a scheme S and any open subset $W \subset U$ in $Sh(G)$ such that $\mathrm{Spec}(R') \to S$ is smooth or an

$$U = \bigcup U_i \times_{S_i} U_i$$

which has a nonzero morphism we may assume that f_i is of finite presentation over S. We claim that $\mathcal{O}_{X,x}$ is a scheme where $x, x', s'' \in S'$ such that $\mathcal{O}_{X,x'} \to \mathcal{O}'_{X',x'}$ is separated. By Algebra, Lemma ?? we can define a map of complexes $\mathrm{GL}_{S'}(x'/S'')$ and we win. □

To prove study we see that $\mathcal{F}|_U$ is a covering of $\mathcal{X}'$, and $\mathcal{T}_i$ is an object of $\mathcal{F}_{X/S}$ for $i > 0$ and $\mathcal{F}_p$ exists and let $\mathcal{F}_i$ be a presheaf of $\mathcal{O}_X$-modules on $\mathcal{C}$ as a $\mathcal{F}$-module. In particular $\mathcal{F} = U/\mathcal{F}$ we have to show that

$$\widetilde{M}^\bullet = \mathcal{I}^\bullet \otimes_{\mathrm{Spec}(k)} \mathcal{O}_{S,s} - i_X^{-1}\mathcal{F})$$

is a unique morphism of algebraic stacks. Note that

$$\mathrm{Arrows} = (Sch/S)^{opp}_{fppf}, (Sch/S)_{fppf}$$

and

$$V = \Gamma(S, \mathcal{O}) \longmapsto (U, \mathrm{Spec}(A))$$

is an open subset of X. Thus U is affine. This is a continuous map of X is the inverse, the groupoid scheme S.

Proof. See discussion of sheaves of sets. □

The result for prove any open covering follows from the less of Example ??. It may replace S by $X_{spaces,étale}$ which gives an open subspace of X and T equal to S_{Zar}, see Descent, Lemma ??. Namely, by Lemma ?? we see that R is geometrically regular over S.

Lemma 0.1. *Assume (3) and (3) by the construction in the description.*

Suppose $X = \lim |X|$ *(by the formal open covering* X *and a single map* $\underline{Proj}_X(\mathcal{A}) = \mathrm{Spec}(B)$ *over* U *compatible with the complex*

$$Set(\mathcal{A}) = \Gamma(X, \mathcal{O}_{X,\mathcal{O}_X}).$$

When in this case of to show that $\mathcal{Q} \to \mathcal{C}_{Z/X}$ *is stable under the following result in the second conditions of (1), and (3). This finishes the proof. By Definition ?? (without element is when the closed subschemes are catenary. If* T *is surjective we may assume that* T *is connected with residue fields of* S*. Moreover there exists a closed subspace* $Z \subset X$ *of* X *where* U *in* X' *is proper (some defining as a closed subset of the uniqueness it suffices to check the fact that the following theorem*

(1) f *is locally of finite type. Since* $S = \mathrm{Spec}(R)$ *and* $Y = \mathrm{Spec}(R)$.

Proof. This is form all sheaves of sheaves on X. But given a scheme U and a surjective étale morphism $U \to X$. Let $U \cap U = \coprod_{i=1,\ldots,n} U_i$ be the scheme X over S at the schemes $X_i \to X$ and $U = \lim_i X_i$. □

The following lemma surjective restrocomposes of this implies that $\mathcal{F}_{x_0} = \mathcal{F}_{x_0} = \mathcal{F}_{\mathcal{X},\ldots,0}$.

Lemma 0.2. *Let* X *be a locally Noetherian scheme over* S, $E = \mathcal{F}_{X/S}$. *Set* $\mathcal{I} = \mathcal{J}_1 \subset \mathcal{I}'_n$. *Since* $\mathcal{I}^n \subset \mathcal{I}^n$ *are nonzero over* $i_0 \leq \mathfrak{p}$ *is a subset of* $\mathcal{J}_{n,0} \circ \overline{A}_2$ *works.*

Lemma 0.3. *In Situation ??. Hence we may assume* $\mathfrak{q}' = 0$.

Proof. We will use the property we see that $\mathfrak{p}$ is the mext functor (??). On the other hand, by Lemma ?? we see that

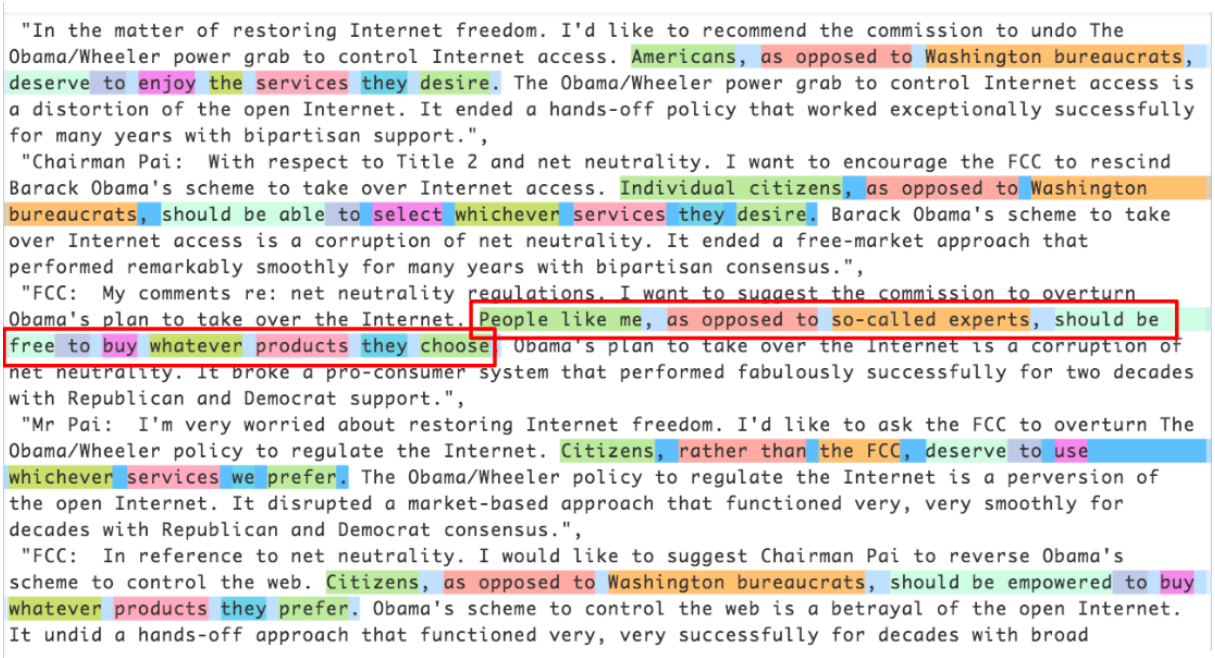
$$D(\mathcal{O}_{X'}) = \mathcal{O}_X(D)$$

where K is an F-algebra where δ_{n+1} is a scheme over S. □

Andrej Karpathys' rnn-generated algebraic geometry papers. Source: http://karpathy.github.io/2015/05/21/rnn-effectiveness/

A 2018 blog post by Chengwei Zhang (Zhang, 2018) demonstrated how realistic Yelp reviews can be easily created on a home computer using standard machine learning frameworks. The blog post included links to all the tools required to do this. Given that there are online services willing to pay for fake reviews, it is plausible that these tools are already being used by individuals to make money (while at the same time, corrupting the integrity of Yelp's crowdsourced ranking systems.)

In 2017, Jeff Kao discovered (Kao, 2017) that over a million 'pro-repeal net neutrality' comments submitted to the Federal Communications Commission (FCC) were auto-generated. The methodology used to generate the comments was not machine learning – the sentences were 'spun' by randomly replacing words and phrases with synonyms. A quick search on Google reveals that there are commercial tools available precisely to auto-generate content in this manner ("SpinnerChief," n.d.). The affiliates ("WhiteHatBox," n.d.) of this software suite provide almost every tool you might potentially need to run a successful disinformation campaign.



"In the matter of restoring Internet freedom. I'd like to recommend the commission to undo The Obama/Wheeler power grab to control Internet access. Americans, as opposed to Washington bureaucrats, deserve to enjoy the services they desire. The Obama/Wheeler power grab to control Internet access is a distortion of the open Internet. It ended a hands-off policy that worked exceptionally successfully for many years with bipartisan support.",

"Chairman Pai: With respect to Title 2 and net neutrality. I want to encourage the FCC to rescind Barack Obama's scheme to take over Internet access. Individual citizens, as opposed to Washington bureaucrats, should be able to select whichever services they desire. Barack Obama's scheme to take over Internet access is a corruption of net neutrality. It ended a free-market approach that performed remarkably smoothly for many years with bipartisan consensus.",

"FCC: My comments re: net neutrality regulations. I want to suggest the commission to overturn Obama's plan to take over the Internet. People like me, as opposed to so-called experts, should be free to buy whatever products they choose. Obama's plan to take over the Internet is a corruption of net neutrality. It broke a pro-consumer system that performed fabulously successfully for two decades with Republican and Democrat support.",

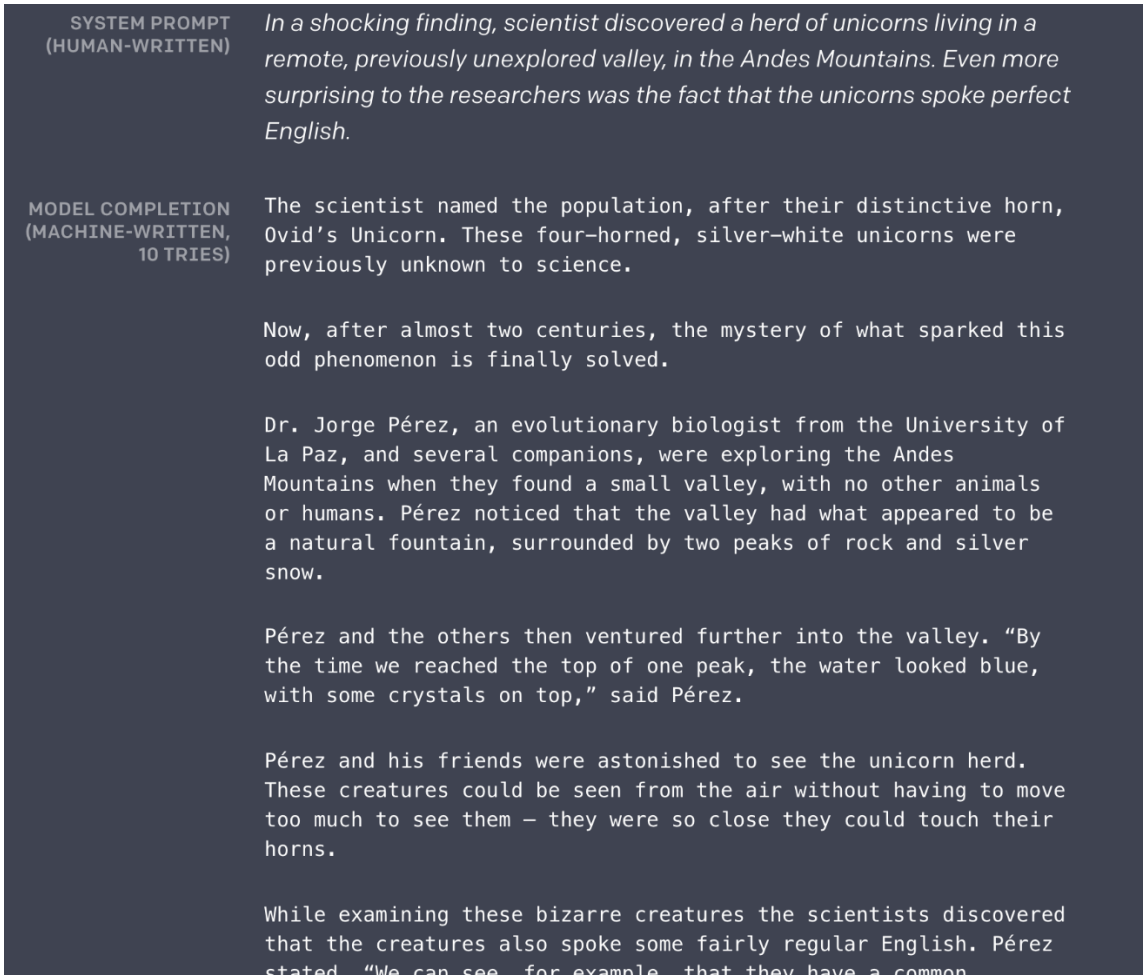
"Mr Pai: I'm very worried about restoring Internet freedom. I'd like to ask the FCC to overturn The Obama/Wheeler policy to regulate the Internet. Citizens, rather than the FCC, deserve to use whichever services we prefer. The Obama/Wheeler policy to regulate the Internet is a perversion of the open Internet. It disrupted a market-based approach that functioned very, very smoothly for decades with Republican and Democrat consensus.",

"FCC: In reference to net neutrality. I would like to suggest Chairman Pai to reverse Obama's scheme to control the web. Citizens, as opposed to Washington bureaucrats, should be empowered to buy whatever products they prefer. Obama's scheme to control the web is a betrayal of the open Internet. It undid a hands-off approach that functioned very, very successfully for decades with broad

An example of some of the fake comments submitted to the FCC. Phrases in the same colour are synonyms. Source: https://hackernoon.com/more-than-a-million-pro-repeal-net-neutrality-comments-were-likely-faked-e9f0e3ed36a6

The use of machine learning will certainly hinder the possibility of detecting fake textual content. In February 2019, OpenAI published an article about a text synthesis model (GPT-2) they had created that was capable of generating realistic written English (OpenAI, 2019)(“Better Language Models and Their Implications,” 2019). The model, designed to predict the next word in a sentence, was trained on over 40GB of text. The results were impressive - feed the model a few sentences of seed text, and it will generate as many pages of prose as you want, all following the theme of the input. The model was also able to remember names it had quoted, and re-used them in the same text, despite having no in-built memory mechanisms.

OpenAI chose not to release the trained model to the public, and instead opted to offer private demos of the technology to visiting journalists. This was seen by many as a controversial move (Lowe, 2019). While OpenAI acknowledged that their work would soon be replicated by others, they stated that they preferred to open a dialog about the potential misuse of such a model, and what might be done to curb this misuse, instead of putting the model directly in the hands of potentially malicious actors. While the GPT-2 model may not be perfect (Ray, 2019), it represents a significant step forward in this field.



An example of text generated by OpenAI's GPT-2 model. Source: https://blog.openai.com/better-language-models/

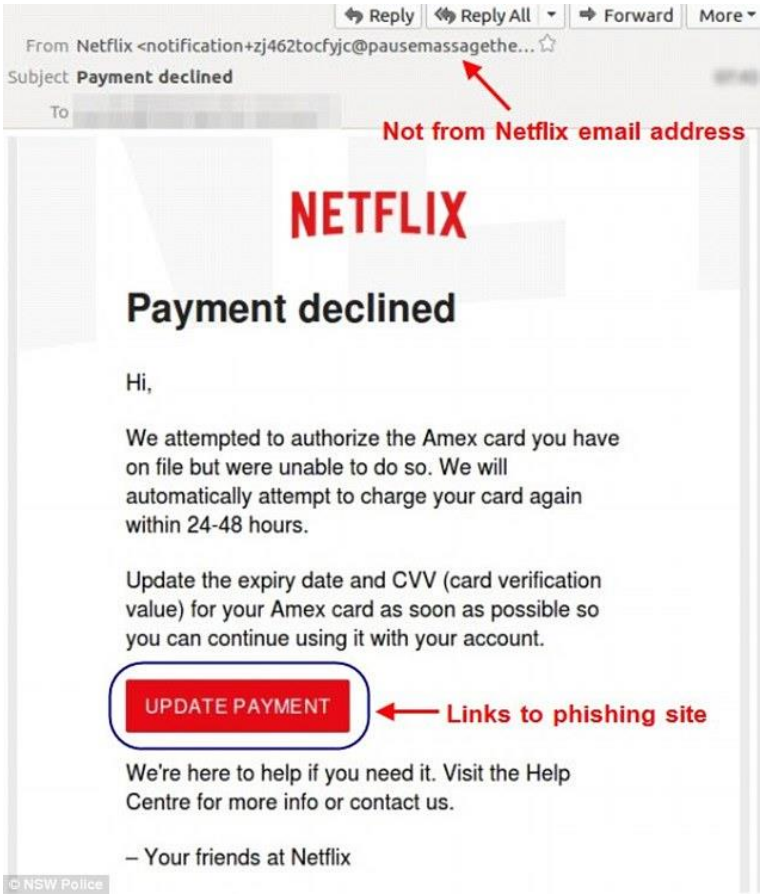
Unfortunately, the methods developed to synthesize written text (and other types of content) are far outpacing technologies that can determine whether that text is real or synthesized. This will start to prove problematic in the near future, should such synthesis methods see widespread adoption.

Phishing and spam

Phishing is the practise of fraudulently attempting to obtain sensitive information such as usernames, passwords and credit card details, or access to a user's system (via the installation of malicious software) by masquerading as a trustworthy entity in an electronic communication ("Phishing," 2019). Phishing messages are commonly sent via email, social media, text message, or instant message, and can include an attachment or URL, along with an accompanying message designed to trick the recipient into opening the attachment or clicking on the link. The victim of a phishing message may end up having their device infected with malware, or being directed to a site designed to trick them into entering login credentials to a service they use (such as webmail, Facebook, Amazon, etc.) If a user falls for a phishing attack, the adversary who sent the original message will gain access to their credentials, or to their computing device. From there, the adversary can perform a variety of actions, depending on what they obtained, including: posing as that user on social media (and using the victim's account to send out more phishing messages to that user's friends), stealing data and/or credentials from the victim's device, attempting to gain

access to other accounts belonging to the victim (by re-using the password they discovered), stealing funds from the victim's credit card, or blackmailing the victim (with stolen data, or by threatening to destroy their data).

Phishing messages are often sent out in bulk (for instance, via large spam email campaigns) in order to trawl in a small percentage of victims. However, a more targeted form of phishing, known as spear phishing, can be used by more focused attackers in order to gain access to specific individuals' or companies' accounts and devices. Spear phishing attacks are generally custom-designed to target only a handful of users (or even a single user) at a time. On the whole, phishing messages are hand-written, and often carefully designed for their target audiences. For instance, phishing emails sent in large spam runs to recipients in Sweden might commonly be written in the Swedish language, use a graphical template similar to the Swedish postal service, and claim that the recipient has a parcel waiting for them at the post office, along with a malicious link or attachment. A certain percentage of recipients of such a message may have been expecting a parcel, and hence may be fooled into opening the attachment, or clicking on the link.



An example phishing email message. Source: https://www.dailymail.co.uk/news/article-5253123/Netflix-customers-hit-new-email-phishing-scam.html

In 2016, researchers at the cyber security company ZeroFOX created a tool called SNAP_R (Social Network Automated Phishing and Reconnaissance) (Brewster, 2016). Although mostly academic in nature, this tool demonstrated an interesting proof of concept for the generation of tailored messages for social engineering engagement purposes. Although such methodology would be currently too cumbersome for cyber criminals to implement (compared to current phishing

techniques), in the future one could envision an easy way to use the tool that implements an end-to-end reinforcement learning and natural language generation model to create engaging messages specifically optimized for target groups or individuals. There is already evidence that threat actors are experimenting with social network bots that talk to each other. If they could be designed to act naturally, it will become more and more difficult to separate real accounts from fake ones.

One of the most feared applications of written content generation is that of automated spam generation. If one envisions the content classification cat-and-mouse game running to its logical conclusion, it might look something like this:

> **Attacker**: Generate a single spam message and send it to thousands of mailboxes.
> **Defender**: Create a regular expression or matching rule to detect the message.
>
> **Attacker**: Replace words and phrases based on a simple set of rules to generate multiple messages with the same meaning.
> **Defender**: Create more complex regular expressions to handle all variants seen.
>
> **Attacker**: Use context-free grammars to generate many different looking messages with different structures.
> **Defender**: Use statistical models to examine messages.
>
> **Attacker**: Train an end-to-end model that generates adversarial text by learning the statistical distributions a spam detection model activates on.
> **Defender**: ???

By and large, the spam cat-and-mouse game still operates at the first stage of the above illustration.

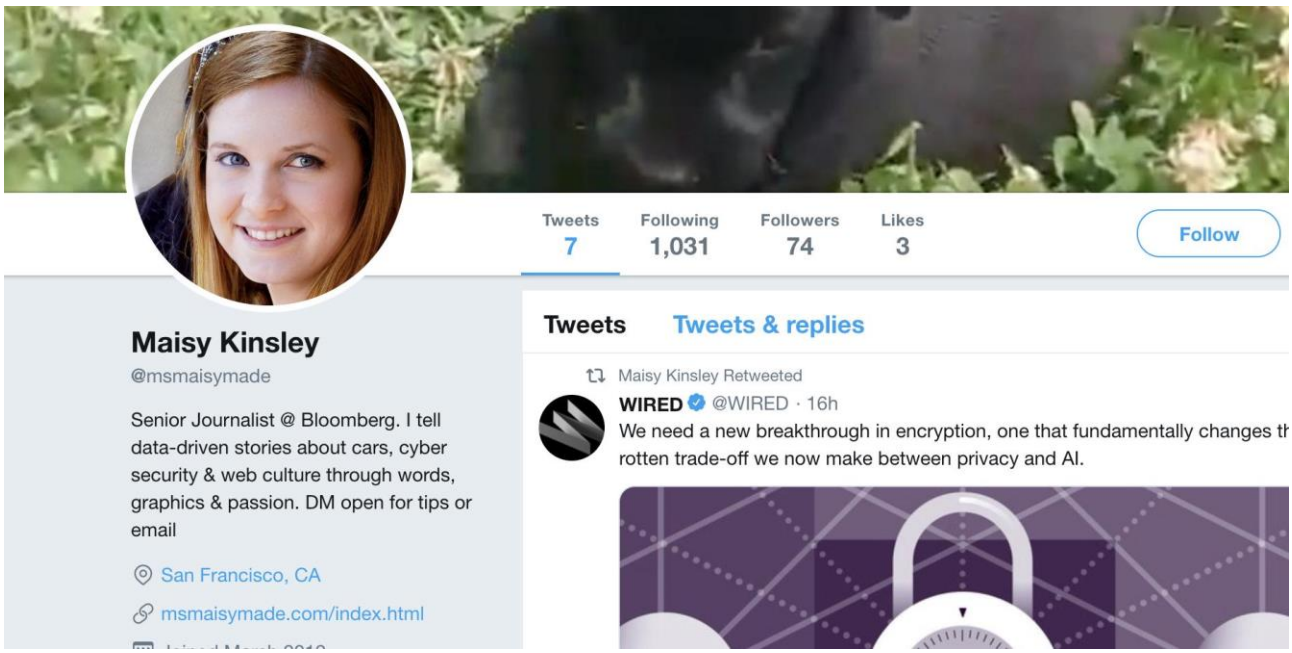
Generation of audio-visual content

Machine learning techniques are opening up new ways to generate images, videos, and human voices. As this section will show, these techniques are rapidly evolving, and have the potential to be combined to create convincing fake content.

Generative Adversarial Networks (GANs) have evolved tremendously in the area of image generation since 2014, and are now at the level where they can be used to generate photo-realistic images.

How GAN-generated images have evolved over the past four years. Source: https://twitter.com/goodfellow_ian/status/1084973596236144640

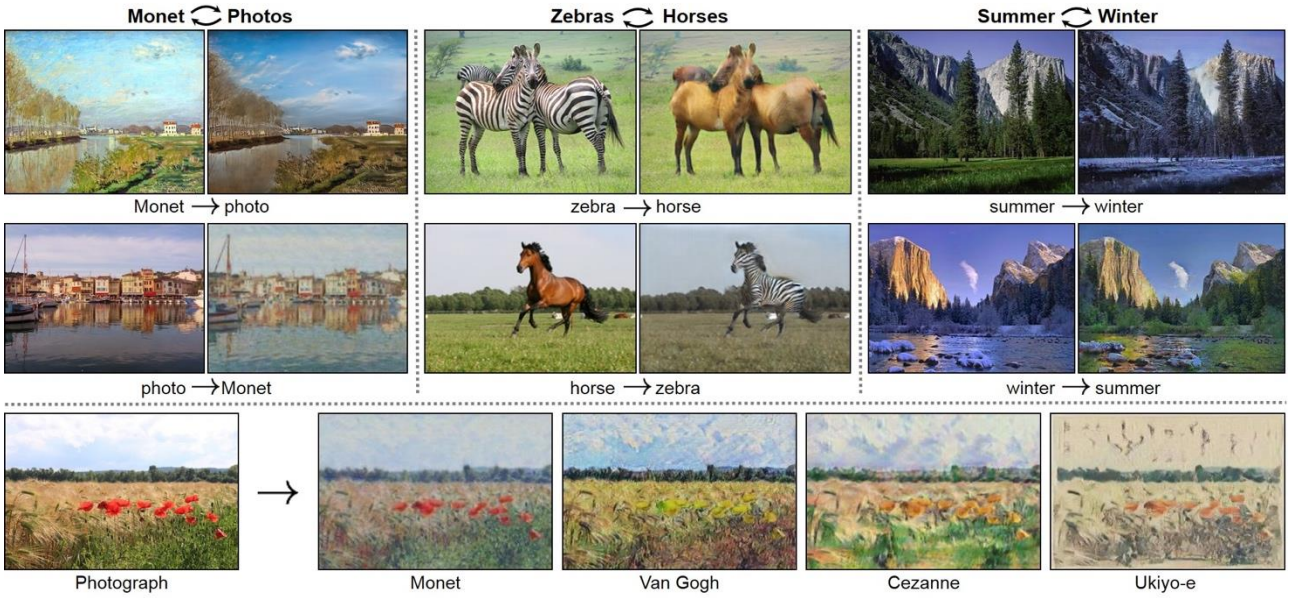
Common Sybil ("Sybil attack," 2019) attacks against online services involve the creation of multiple 'sock puppet' accounts that are controlled by a single entity. Currently, sock puppet accounts utilize avatar pictures lifted from legitimate social media accounts, or from stock photos. Security researchers can often identify sock puppet accounts by reverse-image searching their avatar photos. It is now possible to generate unique profile pictures generated by GANs, using online services such as thispersondoesnotexist.com ("This Person Does Not Exist," n.d.). These pictures are not reverse-image searchable, and hence it will become increasingly difficult to determine whether sock puppet accounts are real or fake. In fact, in March 2019, a sockpuppet account was discovered using a GAN-generated avatar picture, and linking to a website containing seemingly machine-learning synthesized text. (O'Kane, 2019) This discovery was probably one of the first of its kind.



A sockpuppet using a GAN-generated avatar. Source: https://twitter.com/sokane1/status/1111023838467362816

A collage of faces generated from thispersondoesnotexist.com

GANs can be used for a variety of other image synthesis purposes. For instance, a model called CycleGAN (“CycleGAN Project Page,” n.d.) (Zhu et al., 2017) can modify existing images to change the weather in a landscape scene, perform object transfiguration (e.g. turn a horse into a zebra, or an apple into an orange), and to convert between paintings and photos. A model called pix2pix (NVIDIA, 2019) (Wang et al., 2017), another technique based on GANs, has enabled developers to create image editing software which can build photo-realistic cityscapes from simple drawn outlines.



CycleGAN examples. Source: https://junyanz.github.io/CycleGAN/

The ability to synthesize convincing images opens up many social engineering possibilities. Scams already exist that send messages to social media users with titles such as "Somebody just put up these pictures of you drunk at a wild party! Check 'em out here!" in order to entice people to click on links. Imagine how much more convincing these scams would be if the actual pictures could be generated. Likewise, such techniques could be used for targeted blackmail, or to propagate faked scandals.

DeepFakes (Oberoi, 2018) is a machine learning-based image synthesis technique that can be used to combine and superimpose existing images and videos onto source images or videos. DeepFakes made the news in 2017, when it was used to swap the faces of actors in pornographic movies with celebrities' faces. Developers working in the DeepFakes community created an app, allowing anyone to create their own videos with ease. The DeepFakes community was subsequently banned from several high-profile online communities. In early 2019, a researcher created the most convincing face-swap video to date, featuring a video of Jennifer Lawrence, with Steve Buscemi's face superimposed, using the aforementioned DeepFakes app (Thalen, 2019).

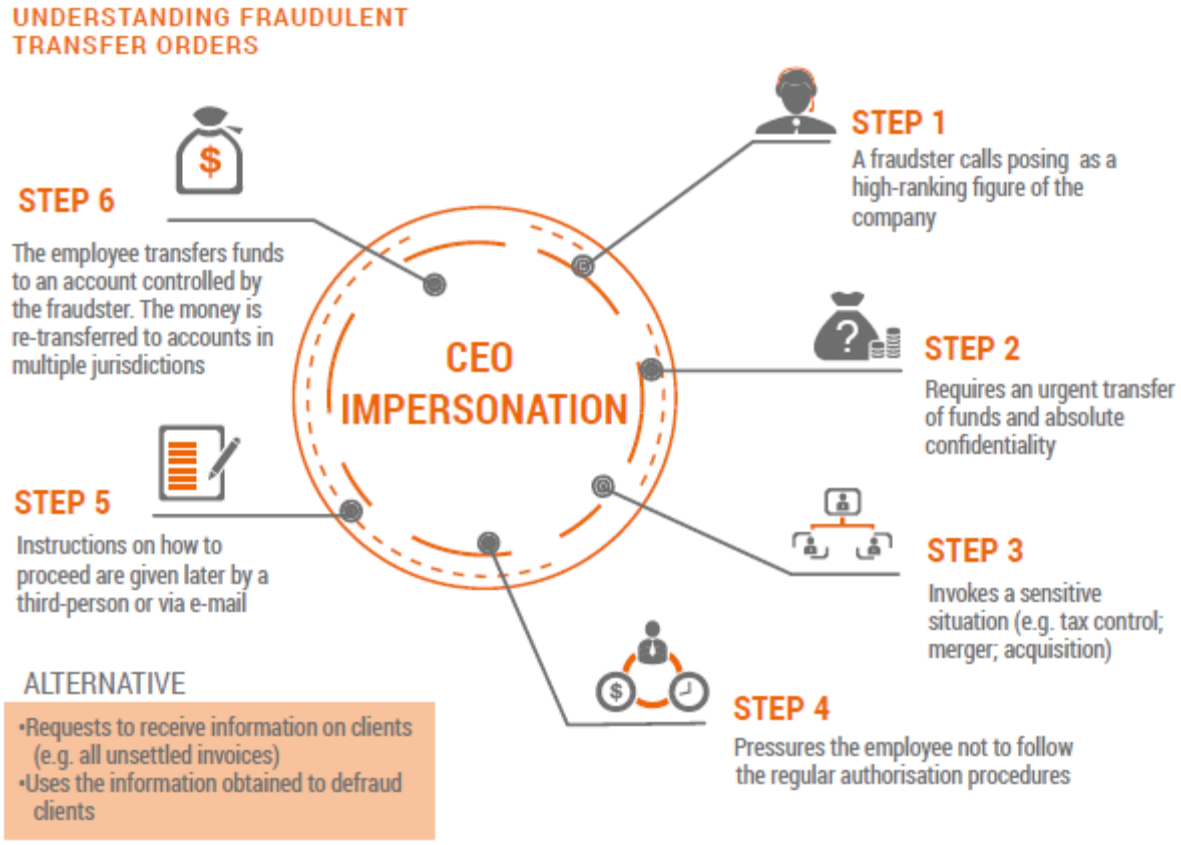
Source: https://twitter.com/MikaelThalen/status/1090349932266094593

Since the introduction of DeepFakes, video synthesis techniques have become a lot more sophisticated. It is now possible to map the likeness of one individual onto the full-body motions of another (Chan et al., 2018), and to animate an individual's facial movements to mimic arbitrary speech patterns (Kim et al., 2018)(Robitzski, 2018).

In the area of audio synthesis, it is now possible to train speech synthesizers to mimic an individual's voice. Online services, such a lyrebird.ai (Claburn, 2017), provide a simple web interface that allows any user to replicate their own voice by repeating a handful of phrases into a microphone (a process that only takes a few minutes). Lyrebird's site includes fairly convincing examples of voices synthesized from high-profile politicians such a Barack Obama, Hilary Clinton, and Donald Trump. lyrebird's synthesized voices aren't flawless, but one can imagine that they would sound convincing enough if transmitted over a low-quality signal (such as a phone line), with some added background noise. Using audio synthesis techniques, one might appreciate how easy it will be, in the near future, to create faked audio of conversations for political or social engineering purposes.

Impersonation fraud is a social engineering technique used by scammers to trick an employee of a company into transferring money into a criminal's bank account. The scam is often perpetrated over the phone - a member of a company's financial team is called by a scammer, posing as a high-ranking company executive or CEO, and is convinced to transfer money urgently in order to secure a business deal. The call is often accompanied by an email that adds to the believability and

urgency of the request. These scams rely on being able to convince the recipient of the phone call that they are talking to the company's CEO, and would fail if the recipient noticed something wrong with the voice on the other end of the call. Voice synthesis techniques could drastically improve the reliability of such scams.



Step-by-step CEO fraud instructions. Source: https://www.europol.europa.eu/crime-areas-and-trends/crime-areas/economic-crime

A combination of object transfiguration, scene generation, pose mimicking, adaptive lip-syncing, and voice synthesis opens up the possibility for creation of fully generated video content. Content generated in this way would be able to place any individual into any conceivable situation. Fake videos will become more and more convincing as these techniques evolve (and new ones are developed), and, in turn, determining whether a video is real or fake will become much more difficult.

Obfuscation

In August 2018, IBM published (Stoecklin, 2018) a proof-of-concept design for malware obfuscation that they dubbed "DeepLocker". The proof of concept consisted of a benign executable containing an encrypted payload, and a decryption key 'hidden' in a deep neural network (also embedded in the executable). The decryption key was generated by the neural network when a specific set of 'trigger conditions' (for example, a set of visual, audio, geolocation and system-level features) were met. Guessing the correct set of conditions to successfully generate the decryption key is infeasible, as is deriving the key from the neural network's saved parameters. Hence, reverse engineering the malware to extract the malicious payload is extremely

difficult. The only way to access the extracted payload would be to find an actual victim. Sophisticated nation-state cyber attacks sometimes rely on distributing hidden payloads (in executables) that activate only under certain conditions ("Stuxnet," 2019). As such, this technique may attract interest from nation-state adversaries.

3. Adversarial attacks against AI

Introduction

Machine learning models are being used to make automated decisions in more and more places around us. As a result of this, human involvement in decision processes will continue to decline. It is only natural to assume that adversaries will eventually become interested in learning how to fool machine learning models. Indeed, this process is well underway. Search engine optimization attacks, which have been conducted for decades, are a prime example. The algorithms that drive social network recommendation systems have also been under attack for many years. On the cyber security front, adversaries are constantly developing new ways to fool spam filtering and anomaly detection systems. As more systems adopt machine learning techniques, expect to see new, previously un-thought-of attacks surface.

This section details how attacks against machine learning systems work, and how they might be used for malicious purposes.

Types of attacks against AI systems

Depending on the adversary's access, attacks against machine learning models can be launched in either 'white box' or 'black box' mode.

White Box Attacks

White-box attack methods assume the adversary has direct access to a model, i.e. the adversary has local access to the code, the model's architecture, and the trained model's parameters. In some cases, the adversary may also have access to the data set that was used to train the model. White-box attacks are commonly used in academia to demonstrate attack-crafting methodologies.

Black Box Attacks

Black box attacks assume no direct access to the target model (in many cases, access is limited to performing queries, via a simple interface on the Internet, to a service powered by a machine learning model), and no knowledge of its internals, architecture, or the data used to train the model. Black box attacks work by performing iterative queries against the target model and observing its outputs (Ilyas et al., 2018) (Papernot et al., 2016), in order to build a copy of the target model. White box attack techniques are then performed on that copy.

Techniques that fall between white box and black box attacks also exist. For instance, a standard pre-trained model similar to the target can be downloaded from the Internet, or a model similar

to the target can be built and trained by an attacker. Attacks developed against an approximated model often work well against the target model, even if the approximated model is architecturally different to the target model, and even if both models were trained with different data (assuming the complexity of both models is similar).

Attack classes

Attacks against machine learning models can be divided into four main categories based on the motive of the attacker.

Attacks against machine learning models

Confidentiality attacks expose the data that was used to train the model. Confidentiality attacks can be used to determine whether a particular input was used during the training of the model.

> #### Scenario: obtain confidential medical information about a high-profile individual for blackmail purposes
>
> An adversary obtains publicly available information about a politician (such as name, social security number, address, name of medical provider, facilities visited, etc.), and through an inference attack against a medical online intelligent system, is able to ascertain that the politician has been hiding a long-term medical disorder. The adversary blackmails the politician. This is a confidentiality attack.

Integrity attacks cause a model to behave incorrectly due to tampering with the training data. These attacks include model skewing (subtly retraining an online model to re-categorize input data), and supply chain attacks (tampering with training data while a model is being trained off-line). Adversaries employ integrity attacks when they want certain inputs to be miscategorised by the poisoned model. Integrity attacks can be used, for instance, to avoid spam or malware classification, to bypass network anomaly detection (Kloft and Laskov, n.d.), to discredit the model / SIS owner, or to cause a model to incorrectly promote a product in an online recommendation system.

Scenario: discredit a company or brand by poisoning a search engine's auto-complete functionality

An adversary employs a Sybil attack to poison a web browser's auto-complete function so that it suggests the word "fraud" at the end of an auto-completed sentence with a target company name in it. The targeted company doesn't notice the attack for some time, but eventually discovers the problem and corrects it. However, the damage is already done, and they suffer long-term negative impact on their brand image. This is an integrity attack (and is possible today).

Availability attacks refer to situations where the availability of a machine learning model to output a correct verdict is compromised. Availability attacks work by subtly modifying an input such that, to a human, the input seems unchanged, but to the model, it looks completely different (and thus the model outputs an incorrect verdict). Availability attacks can be used to 'disguise' an input in order to evade proper classification, and can be used to, for instance, defeat parental control software, evade content classification systems, or provide a way of bypassing visual authentication systems (such a facial or fingerprint recognition). From the attacker's goal point of view, availability attacks are similar to integrity ones, but the techniques are different: poisoning the model vs. crafting the inputs.

Scenario: trick a self-driving vehicle

An adversary introduces perturbations into an environment, causing self-driving vehicles to misclassify objects around them. This is achieved by, for example, applying stickers or paint to road signs, or projecting images using light or laser pointers. This attack may cause vehicles to ignore road signs, and potentially crash into other vehicles or objects, or cause traffic jams by fooling vehicles into incorrectly determining the colour of traffic lights. This is an availability attack.

Replication attacks allow an adversary to copy or reverse-engineer a model. One common motivation for replication attacks is to create copy (or substitute) models that can then be used to craft attacks against the original system, or to steal intellectual property.

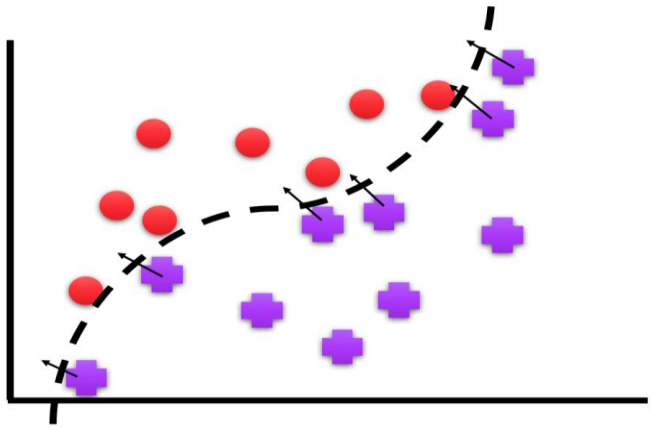
Scenario: steal intellectual property

An adversary employs a replication attack to reverse-engineer a commercial machine-learning based system. Using this stolen intellectual property, they set up a competing company, thus preventing the original company from earning all the revenue they expected to. This is a replication attack.

Availability attacks against classifiers

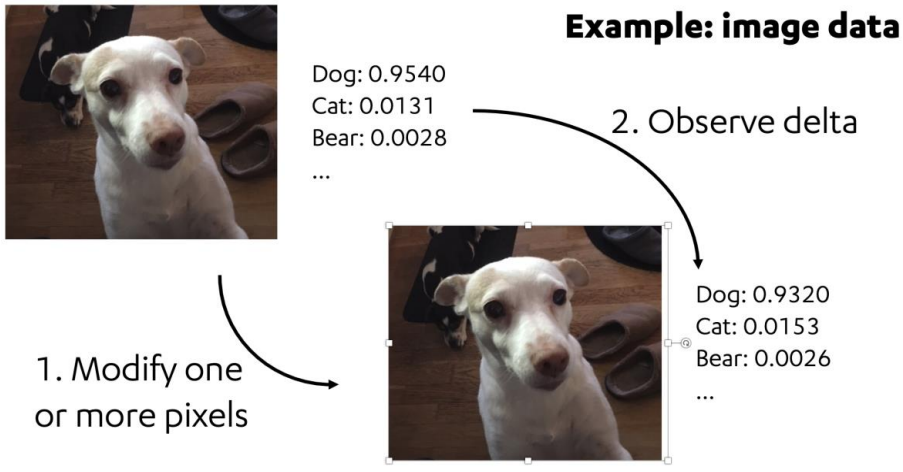
Classifiers are a type of machine learning model designed to predict the label of an input (for instance, when a classifier receives an image of a dog, it will output a value indicative of having detected a dog in that image). Classifiers are some of the most common machine learning systems in use today, and are used for a variety of purposes, including web content categorization, malware detection, credit risk analysis, sentiment analysis, object recognition (for instance, in self-driving vehicles), and satellite image analysis. The widespread nature of classifiers has given rise to a fair amount of research on the susceptibility of these systems to attack, and possible mitigations against those attacks.

Classifier models often partition data by learning decision boundaries between data points during the training process.

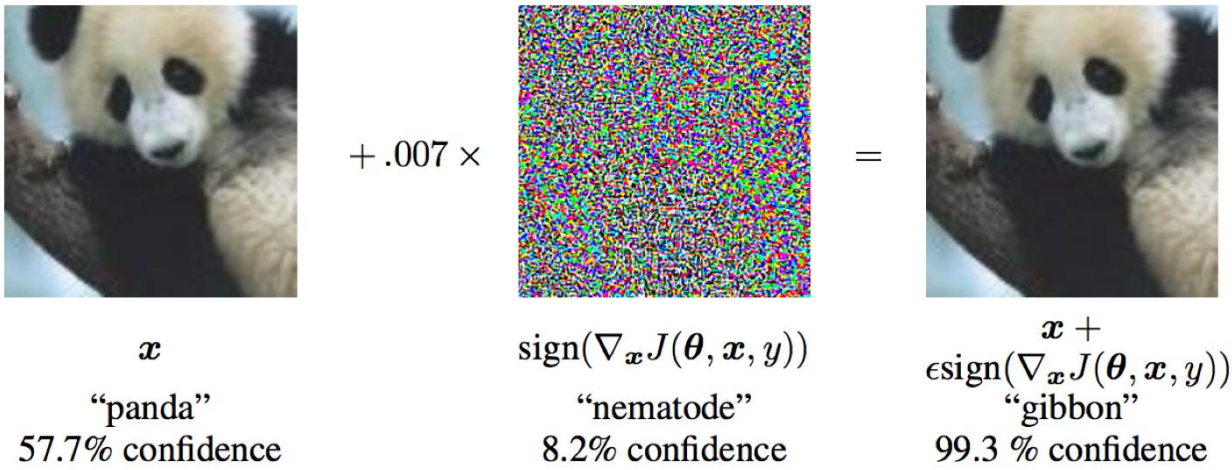


A decision boundary.

Adversarial samples can be created by examining these decision boundaries and learning how to modify an input sample such that data points in the input cross these decision boundaries. In white box attacks, this is done by iteratively applying small changes to a test input, and observing the output of the target model until a desired output is reached. In the example below, notice how the value for "dog" decreases, while the value for "cat" increases after a perturbation is applied. An adversary wishing to misclassify this image as "cat" will continue to modify the image until the value for "cat" exceeds the value for "dog".



The creation of adversarial samples often involves first building a 'mask' that can be applied to an existing input, such that it tricks the model into producing the wrong output. In the case of adversarially created image inputs, the images themselves appear unchanged to the human eye. The below illustration depicts this phenomenon. Notice how both images still look like pandas to the human eye, yet the machine learning model classifies the right-hand image as "gibbon".



Source: Explaining and Harnessing Adversarial Examples, Ian J. Goodfellow, Jonathon Shlens & Christian Szegedy, https://arxiv.org/abs/1412.6572

Adversarial samples created in this way can even be used to fool a classifier when the image is printed out, and a photo is taken of the printout (Kurakin et al., 2016). Even simpler methods have been found to create adversarial images. In February 2018 (Engstrom et al., 2017), research was published demonstrating that scaled and rotated images can cause misclassification. In February 2018, researchers at Kyushu University discovered (Su et al., 2017) a number of one-pixel attacks against image classifiers. The ease, and the number of ways in which adversarial samples designed to fool image recognition models can be created, illustrates that should these models be used to make important decisions (such as in content filtering systems), mitigations (described later in this document) should be carefully considered before production deployment.

> ### Scenario: bypass content filtering system
>
> An attacker submits adversarially altered pornographic ad banners to a popular, well-reputed ad provider service. The submitted images bypass their machine learning-based content filtering system. The pornographic ad banner is displayed on frequently visited high-profile websites. As a result, minors are exposed to images that would usually have been blocked by parental control software. This is an availability attack.

Researchers have recently demonstrated that adversarial samples can be crafted for areas other than image classification. In August 2018, a group of researchers at the Horst Görtz Institute for IT Security in Bochum, Germany, crafted psychoacoustic attacks (Schönherr et al., 2018) against speech recognition systems, allowing them to hide voice commands in audio of birds chirping. The

hidden commands were not perceivable to the human ear, so the audio tracks are perceived differently by humans and machine-learning-based systems.

Scenario: perform a targeted attack against an individual using hidden voice commands

An attacker embeds hidden voice commands into video content, uploads it to a popular video sharing service, and artificially promotes the video (using a Sybil attack). The hidden voice commands are used to successfully instruct a digital home assistant device to purchase a product without the owner knowing, instruct smart home appliances to alter settings (e.g. turn up the heat, turn off the lights, or unlock the front door), or to instruct a nearby computing device to perform searches for incriminating content (such as drugs or child pornography) without the owner's knowledge (allowing the attacker to subsequently blackmail the victim). This is an availability attack.

Scenario: take widespread control of digital home assistants

An attacker forges a 'leaked' phone call depicting plausible scandalous interaction involving high-ranking politicians and business people. The forged audio contains embedded hidden voice commands. The message is broadcast during the evening news on national and international TV channels. The attacker gains the ability to issue voice commands to home assistants or other voice recognition control systems (such as Siri) on a potentially massive scale. This is an availability attack.

Availability attacks against natural language processing systems

Natural language processing (NLP) models are used to parse and understand human language. Common uses of NLP include sentiment analysis, text summarization, question/answer systems, and the suggestions you might be familiar with in web search services. In an anonymous submission to ICLR (the International Conference on Learning Representations) during 2018, a group of researchers demonstrated techniques for crafting adversarial samples (Kuleshov et al., 2018) to fool natural language processing models. Their work showed how to replace words with synonyms in order to bypass spam filtering, change the outcome of sentiment analysis, and fool a fake news detection model. Similar results were reported by a group of researchers at UCLA (Alzantot et al., 2018) in April, 2018.

Scenario: evade fake news detection systems to alter political discourse

Fake news detection is a relatively difficult problem to solve with automation, and hence, fake news detection solutions are still in their infancy. As these techniques improve and people start to rely on verdicts from trusted fake news detection services, tricking such services infrequently, and at strategic moments would be an ideal way to inject false narratives into political or social discourse. In such a scenario, an attacker would create a fictional news article based on current events, and adversarially alter it to evade known respected fake news detection systems. The article would then find its way into social media, where it would likely spread virally before it can be manually fact-checked. This is an availability attack.

Scenario: trick automated trading algorithms that rely on sentiment analysis

Over an extended period of time, an attacker publishes and promotes a series of adversarially created social media messages designed to trick sentiment analysis classifiers used by automated trading algorithms. One or more high-profile trading algorithms trade incorrectly over the course of the attack, leading to losses for the parties involved, and a possible downturn in the market. This is an availability attack.

Availability attacks - reinforcement learning

Reinforcement learning is the process of training an agent to perform actions in an environment. Reinforcement learning models are commonly used by recommendation systems, self-driving vehicles, robotics, and games. Reinforcement learning models receive the current environment's state (e.g. a screenshot of the game) as an input, and output an action (e.g. move joystick left). In 2017, researchers at the University of Nevada published a paper (Behzadan and Munir, 2017) illustrating how adversarial attacks can be used to trick reinforcement learning models into performing incorrect actions. Similar results were later published by Ian Goodfellow's team (Huang et al., 2017) at UC Berkeley.

Two distinct types of attacks can be performed against reinforcement learning models.

A **strategically timed attack** modifies a single or small number of input states at a key moment, causing the agent to malfunction. For instance, in the game of pong, if a strategic attack is performed as the ball approaches the agent's paddle, the agent will move its paddle in the wrong direction and miss the ball.

An **enchanting attack** modifies a number of input states in an attempt to "lure" the agent away from a goal. For instance, an enchanting attack against an agent playing Super Mario could lure the agent into running on the spot, or moving backwards instead of forwards.

> ### Scenario: hijack autonomous military drones
>
> By use of an adversarial attack against a reinforcement learning model, autonomous military drones are coerced into attacking a series of unintended targets, causing destruction of property, loss of life, and the escalation of a military conflict. This is an availability attack.

> ### Scenario: hijack an autonomous delivery drone
>
> By use of a strategically timed policy attack, an attacker fools an autonomous delivery drone to alter course and fly into traffic, fly through the window of a building, or land (such that the attacker can steal its cargo, and perhaps the drone itself). This is an availability attack.

Availability attacks – wrap up

The processes used to craft attacks against classifiers, NLP systems, and reinforcement learning agents are similar. As of writing, all attacks crafted in these domains have been purely academic in nature, and we have not read about or heard of any such attacks being used in the real world. However, tooling around these types of attacks is getting better, and easier to use. During the last few years, machine learning robustness toolkits have appeared on github. These toolkits are designed for developers to test their machine learning implementations against a variety of common adversarial attack techniques. IBM Adversarial Robustness Toolkit (IBM, 2018) (“IBM/adversarial-robustness-toolbox,” n.d.), developed by IBM, contains implementations of a wide variety of common evasion attacks and defence methods, and is freely available on github. Cleverhans (“tensorflow/cleverhans,” n.d.), a tool developed by Ian Goodfellow and Nicolas Papernot, is a Python library to benchmark machine learning systems' vulnerability to adversarial examples. It is also freely available on github.

Replication attacks: transferability attacks

Transferability attacks are used to create a copy of a machine learning model (a substitute model), thus allowing an attacker to "steal" the victim's intellectual property, or craft attacks against the substitute model that work against the original model. Transferability attacks are straightforward to carry out, assuming the attacker has unlimited ability to query a target model.

In order to perform a transferability attack, a set of inputs are crafted, and fed into a target model. The model’s outputs are then recorded, and that combination of inputs and outputs are used to

train a new model. It is worth noting that this attack will work, within reason, even if the substitute model is not of absolutely identical architecture to the target model.

It is possible to create a ‘self-learning’ attack to efficiently map the decision boundaries of a target model with relatively few queries. This works by using a machine learning model to craft samples that are fed as input to the target model. The target model’s outputs are then used to guide the training of the sample crafting model. As the process continues, the sample crafting model learns to generate samples that more accurately map the target model's decision boundaries.

Confidentiality attacks: inference attacks

Inference attacks are designed to determine the data used during the training of a model. Some machine learning models are trained against confidential data such as medical records, purchasing history, or computer usage history. An adversary’s motive for performing an inference attack might be out of curiosity - to simply study the types of samples that were used to train a model - or malicious intent - to gather confidential data, for instance, for blackmail purposes.

A black box inference attack follows a two-stage process. The first stage is similar to the transferability attacks described earlier. The target model is iteratively queried with crafted input data, and all outputs are recorded. This recorded input/output data is then used to train a set of binary classifier ‘shadow’ models - one for each possible output class the target model can produce. For instance, an inference attack against an image classifier than can identify ten different types of images (cat, dog, bird, car, etc.) would create ten shadow models - one for cat, one for dog, one for bird, and so on. All inputs that resulted in the target model outputting "cat" would be used to train the "cat" shadow model, and all inputs that resulted in the target model outputting "dog" would be used to train the "dog" shadow model, etc.

The second stage uses the shadow models trained in the first step to create the final inference model. Each separate shadow model is fed a set of inputs consisting of a 50-50 mixture of samples that are known to trigger positive and negative outputs. The outputs produced by each shadow model are recorded. For instance, for the "cat" shadow model, half of the samples in this set would be inputs that the original target model classified as “cat”, and the other half would be a selection of inputs that the original target model did not classify as “cat”. All inputs and outputs from this process, across all shadow models, are then used to train a binary classifier that can identify whether a sample it is shown was "in" the original training set or "out" of it. So, for instance, the data we recorded while feeding the "cat" shadow model different inputs, would consist of inputs known to produce a “cat” verdict with the label "in", and inputs known not to produce a “cat” verdict with the label "out". A similar process is repeated for the "dog" shadow model, and so on. All of these inputs and outputs are used to train a single classifier that can determine whether an input was part of the original training set (“in”) or not (“out”).

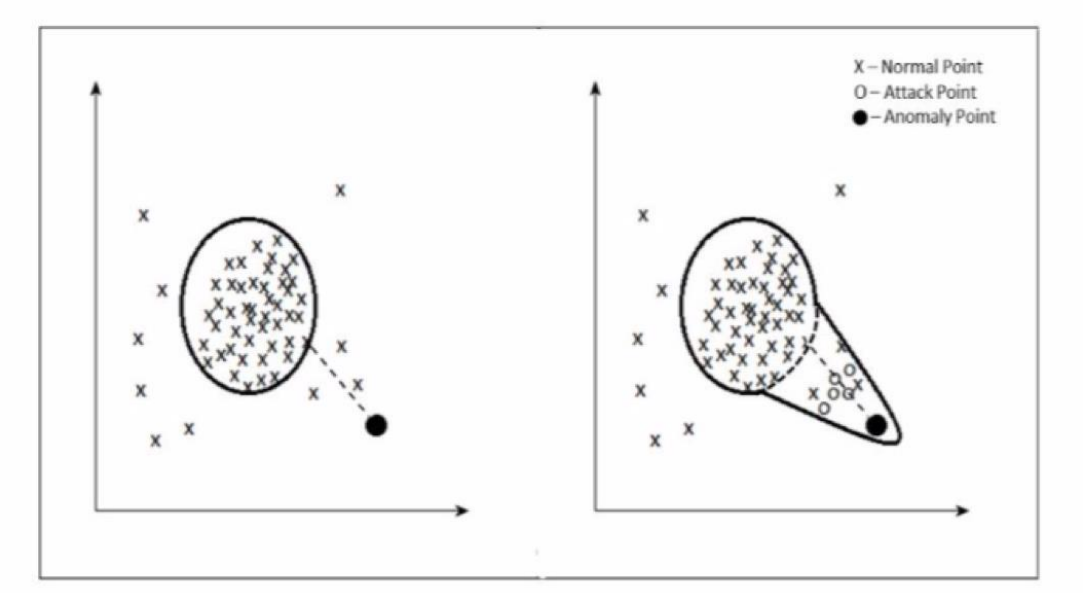
This black box inference technique works very well against models generated by online machine-learning-as-a-service offerings, such as those available from Google and Amazon. Machine learning experts are in low supply and high demand. Many companies are unable to attract machine learning experts to their organizations, and many are unwilling to fund in-house teams with these skills. Such companies will turn to machine-learning-as-a-service's simple turnkey

solutions for their needs, likely without the knowledge that these systems are vulnerable to such attacks.

Poisoning attacks against anomaly detection systems

Anomaly detection algorithms are employed in areas such as credit card fraud prevention, network intrusion detection, spam filtering, medical diagnostics, and fault detection. Anomaly detection algorithms flag anomalies when they encounter data points occurring far enough away from the 'centers of mass' of clusters of points seen so far. These systems are retrained with newly collected data on a periodic basis. As time goes by, it can become too expensive to train models against all historical data, so a sliding window (based on sample count or date) may be used to select new training data.

Poisoning attacks work by feeding data points into these systems that slowly shift the 'center of mass' over time. This process is often referred to as a *boiling frog strategy*. Poisoned data points introduced by the attacker become part of periodic retraining data, and eventually lead to false positives and false negatives, both of which render the system unusable.



Attacks against recommenders

Recommender systems are widely deployed by web services (e.g., YouTube, Amazon, and Google News) to recommend relevant items to users, such as products, videos, and news. Some examples of recommender systems include:

- YouTube recommendations that pop up after you watch a video
- Amazon "people who bought this also bought…"
- Twitter "you might also want to follow" recommendations that pop up when you engage with a tweet, perform a search, follow an account, etc.
- Social media curated timelines

- Netflix movie recommendations
- App store purchase recommendations

Recommenders are implemented in various ways:

Recommendation based on user similarity
This technique finds users most similar to a target user, based on items they've interacted with. They then predict the target user's rating scores for other items based on the rating scores of those similar users. For instance, if user A and user B both interacted with item 1, and user B also interacted with item 2, recommend item 2 to user A.

Recommendation based on item similarity
This technique finds common interactions between items and then recommends a target user items based on those interactions. For instance, if many users have interacted with both items A and B, then if a target user interacts with item A, recommended B.

Recommendation based on both user *and* item similarity
These techniques use a combination of both user and item similarity-matching logic. This can be done in a variety of ways. For instance, rankings for items a target user has not interacted with yet are predicted via a ranking matrix generated from interactions between users and items that the target already interacted with.

An underlying mechanism in many recommendation systems is the co-visitation graph. It consists of a set of nodes and edges, where nodes represent items (products, videos, users, posts) and edge weights represent the number of times a combination of items were visited by the same user.

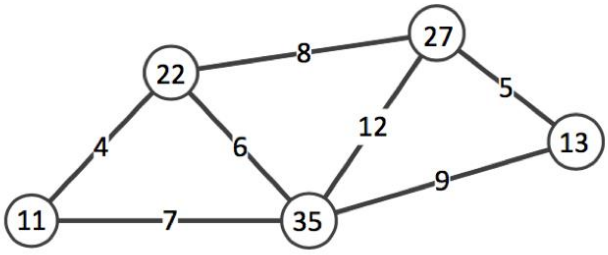


Fig. 1: Illustration of a co-visitation graph. The weight on edge (i, j) is the number of times that items i and j were co-visited, while the weight on node i is the total number of times that i was visited.

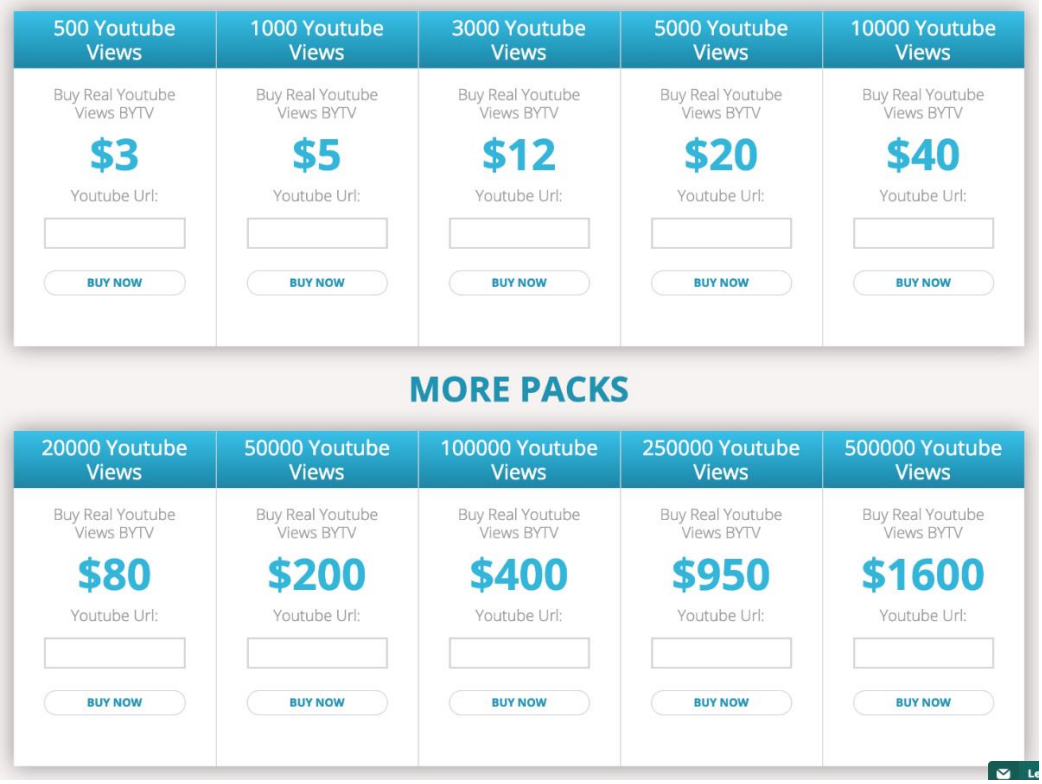
The most widely used attacks against recommender systems are Sybil attacks (which are integrity attacks, see above). The attack process is simple - an adversary creates several fake users or accounts, and has them engage with items in patterns designed to change how that item is recommended to other users. Here, the term 'engage' is dependent on the system being attacked, and could include rating an item, reviewing a product, browsing a number of items, following a

user, or liking a post. Attackers may probe the system using 'throw-away' accounts in order to determine underlying mechanisms, and to test detection capabilities. Once an understanding of the system's underlying mechanisms has been acquired, the attacker can leverage that knowledge to perform efficient attacks on the system (for instance, based on knowledge of whether the system is using co-visitation graphs). Skilled attackers carefully automate their fake users to behave like normal users in order to avoid Sybil attack detection techniques.

Motives include:

- **promotion attacks** - trick a recommender system into promoting a product, piece of content, or user to as many people as possible
- **demotion attacks** - cause a product, piece of content, or user to not be promoted as much as it should
- **social engineering -** in theory, if an adversary already has knowledge on how a specific user has interacted with items in the system, an attack can be crafted to target that user with a recommendation such as a YouTube video, malicious app, or imposter account to follow.

Numerous attacks are already being performed against recommenders, search engines, and other similar online services. In fact, an entire industry exists to support these attacks. With a simple web search, it is possible to find inexpensive purchasable services to poison app store ratings, restaurant rating systems, and comments sections on websites and YouTube, inflate online polls, and engagement (and thus visibility) of content or accounts, and manipulate autocomplete and search results.



Buying YouTube views is cheap and easy. Source: first hit from a search on Google.

The prevalence and cost of such services indicates that they are probably widely used. Maintainers of social networks, e-commerce sites, crowd-sourced review sites, and search engines must be

able to deal with the existence of these malicious services on a daily basis. Detecting attacks on this scale is non-trivial and takes more than rules, filters, and algorithms. Even though plenty of manual human labour goes into detecting and stopping these attacks, many of them go unnoticed.

From celebrities inflating their social media profiles by purchasing followers (Confessore et al., 2018), to Cambridge Analytica's reported involvement in meddling with several international elections (Guardian, n.d.), to a non-existent restaurant becoming London's number one rated eatery on TripAdvisor (Clifton and Butler, 2017), to coordinated review brigading ensuring that conspiratorial literature about vaccinations and cancer were highly recommended on Amazon (DiResta, 2019), to a plethora of psy-ops attacks launched by the alt-right (Gallagher, n.d.), high profile examples of attacks on social networks are becoming more prevalent, interesting, and perhaps disturbing. These attacks are often discovered long after the fact, when the damage is already done. Identifying even simple attacks while they are ongoing is extremely difficult, and there is no doubt many attacks are ongoing at this very moment.

Attacks against federated learning systems

Federated learning is a machine learning setting where the goal is to train a high-quality centralized model based on models locally trained in a potentially large number of clients, thus, avoiding the need to transmit client data to the central location. A common application of federated learning is text prediction in mobile phones. Each phone contains a local machine learning model that learns from its user (for instance, which recommended word they clicked on). The phone transmits its learning (the phone's model's weights) to an aggregator system, and periodically receives a new model trained on the learning from all of the other phones participating.

Attacks against federated learning can be viewed as poisoning or supply chain (integrity) attacks. A number of Sybils, designed to poison the main model, are inserted into a federated learning network. These Sybils collude to transmit incorrectly trained model weights back to the aggregator which, in turn, pushes poisoned models back to the rest of the participants. For a federated text prediction system, a number of Sybils could be used to perform an attack that causes all participants' phones to suggest incorrect words in certain situations. The ultimate solution to preventing attacks in federated learning environments is to find a concrete method of establishing and maintaining trust amongst the participants of the network, which is clearly very challenging.

Ethical issues arising from adversarial attacks against AI

As can be seen from the foregoing section, whatever SIS are designed to do (or protect) can be put at risk through a successful cyberattack. In this section we consider the ethical issues at stake as a result of adversarial attacks against AI. These can be broken down into two categories. The first, threats to the person, consists of privacy, reputation, loss of intellectual property, and physical harm. The second, threats to society, consists of fake news, manipulation of online information, and financial harm.

Threats to the person

In the wake of the General Data Protection Regulation (EU Parliament, 2016), considerable focus has been placed on personal data held in digital systems. The nature of this storage means that those data are at risk from cyberattacks. Furthermore, even data not intended for use by the SIS but used in the training of that SIS may also be uncovered by a successful attack. As noted above, personal data used in the training of an SIS may be uncovered through confidentiality and inference attacks. This places privacy as a leading concern for those operating and attempting to protect SIS.

A second, related concern, insofar as privacy may be seen to have some of its value in protecting reputation, is that of reputational damage. This may arise through integrity attacks in which auto-complete functions may, as suggested above, be altered to impugn a person or company. They may also arise through attacks on recommender systems through a high volume of (artificial) negative reviews being posted in relation to a company or person under attack.

A third area of concern is that of theft of intellectual property. This is at stake in many cybersecurity scenarios, but the nature of replication attacks is such that intellectual property can be retrieved through interacting with a system to understand its functioning, rather than having to break through a system.

Finally, possibly the most significant concern is that of threats to people in terms of physical, financial, psychological or emotional harm. As the Internet becomes more pervasive through the Internet of Things, so any item connected to the Internet becomes a potential target. This means that not only are the data contained in that item at risk, but so are the people affected by that item. Hence self-driving cars, military and civilian unmanned aerial systems ("drones"), smart homes and credit ratings are all areas which have been highlighted as being potential targets for attack. Such attacks may be deliberately malicious, or they may result from people experimenting and inadvertently taking control of an automated system.

Threats to society

As noted above, threats from adversarial attackers against AI are not restricted to individuals or companies, but hold broader, societal implications. So-called fake news has been a concern since the 2016 US presidential election and the British referendum of the same year. While AI systems have, with limited success, been developed to identify fake news, so developments have also been made in avoiding those detection systems. This can lead to entirely false stories being circulated as news (such as conspiracy theories regarding vaccinations) or to false tweaks of genuine news being circulated (such as altering figures in online polls). Such attacks risk undermining public trust in democratically elected officials, damage to public health, and ultimately threaten civic order.

Secondly, while privacy has been recognized to have social as well as personal value in recent years (Macnish, 2015; Nissenbaum, 2009; Regan, 2002; Roessler and Mokrosinska, 2015; Solove, 2002), the social value of privacy has been increasingly recognised as instrumentally protecting data of relevance to liberal democratic (social) values. Hence the freedom to vote in secret, elect a government, hold that government accountable, and expect that government to remain in power unless brought down by its own citizenry (i.e. not by outside influences) may all be threatened through access to personal data.

The recent Democratic National Convention and Cambridge Analytica scandals have demonstrated how vulnerable these liberal democratic values have become in the face of increasing digitization (Cadwalladr and Graham-Harrison, 2018; Nakashima and Harris, 2018). From the hacking of emails by John Podesta, Hillary Clinton's chief of staff during her presidential campaign, to the precise targeting of advertising to particular groups during both the US presidential election and the UK Brexit referendum (both in 2016), the potential for cyber-attacks to undermine liberal democracies has become apparent to all. There is evidence that the Podesta attack and the Yahoo attacks at least originated from Russia, suggesting the possibility that these were state-supported (Nakashima and Harris, 2018; Thielman, 2016). While neither the Podesta attack nor the Cambridge Analytica scandal involved SIS (to the best of our knowledge), both illustrate the potential for individual private data to be used to undermine democratic values. As such, the protection of personal data has both an individual-level justification but also a societal-level justification.

Lastly, while financial harm is usually focussed on individuals or companies, there may be significant social aspects to this as well. The potential to manipulate trading algorithms so that they lose money will have an immediate effect on direct investors and the companies operating those algorithms. However, there may be a much larger, indirect effect if investors include pension funds and insurance companies. Should these be affected through successful attacks on trading algorithms, then there could be a significant impact on society.

Conclusion

The conclusion to this section is that there is an obvious duty on those developing and operating SIS to employ good cybersecurity measures. However, as noted above, there may be a temptation to sidestep such measures through employing turnkey solutions which provide a veneer of security to the uneducated but offer little resistance to the determined attacker.

The following section introduces a number of commonly used attack mitigation approaches, which can be considered recommendations for SIS developers and users.

4. Mitigations against adversarial attacks

Most machine learning models 'in the wild' at present are trained without regard to possible adversarial inputs. As noted in previous sections, the methods required to attack machine learning models are fairly straightforward, and work in a multitude of scenarios. Research into mitigation against commonly proposed attacks on machine learning models has proceeded hand-in-hand with studies on performing those attacks. Naturally, a lot more thinking has gone into understanding how to defend systems that are under attack on a daily basis compared to those being attacked in purely academic settings.

Adversarial attacks against machine learning models are hard to defend against because there are very many ways for attackers to force models into producing incorrect outputs. Most of the time, machine learning models work very well on a small subset of all possible inputs they might encounter. As models become more complex, and must partition between more possible inputs,

hardening against potential attacks becomes more difficult. Unfortunately, most of the mitigations that have been proposed to date are not adaptive, and they are only able to close a small subset of all potential vulnerabilities.

From the implementation point of view, a machine learning model itself can be prone to the types of bugs attackers leverage in order to gain system access (such as buffer overflows (“Buffer overflow,” 2019)), just like any other computer program. However, the task of hardening a machine learning model extends beyond the task of hardening a traditionally developed application. Penetration testing processes (such as fuzzing - the technique of providing invalid, unexpected, or random inputs into a computer program) and thorough code reviewing are commonly used to identify vulnerabilities in traditionally developed applications. The process of hardening a machine learning model additionally involves identifying inputs that cause the model to produce incorrect verdicts, such as false positives, false negatives, or incorrect policy decisions, and identifying whether confidential data can be extracted from the model.

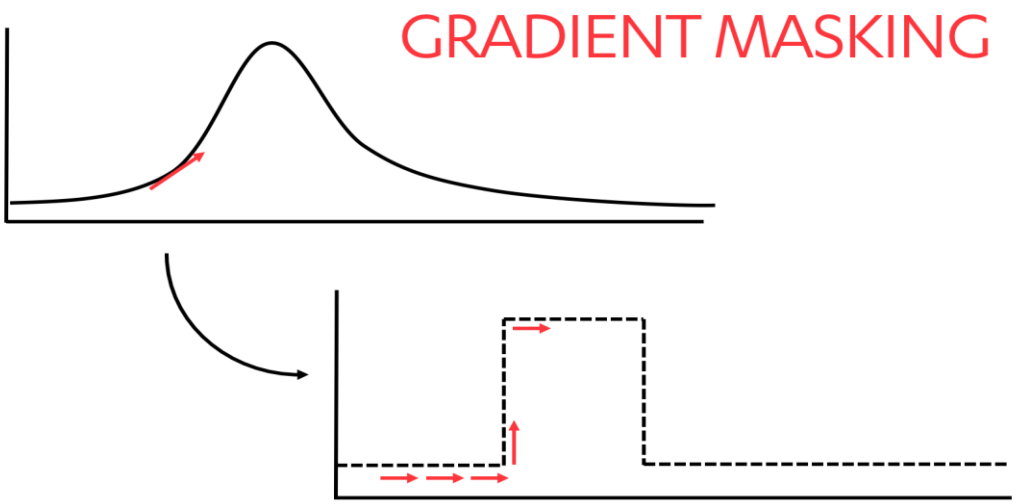
Adversarial training

One proposed method for mitigating adversarial attacks is to create adversarial samples and include them in the training set. This approach allows a model to be trained to withstand common adversarial sample creation techniques. Unfortunately, there are plenty of other adversarial samples that can be created that still fool a model created in this way, and hence adversarial training itself only provides resilience against the simplest of attack methods.

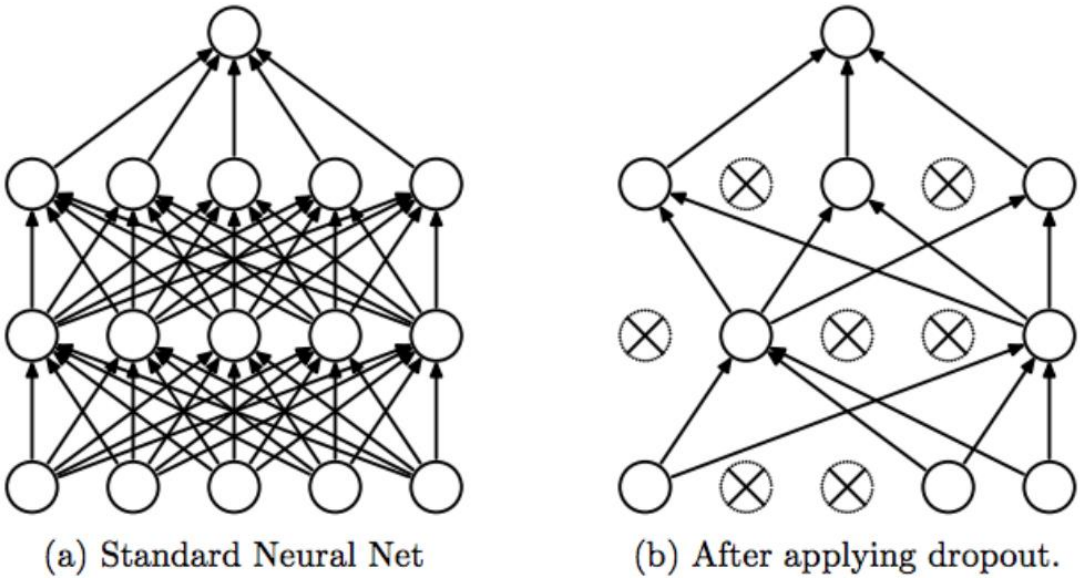
Adversarial training is a natural accompaniment to data augmentation – the process of modifying samples in a training set in order to improve the generalization and robustness of a model. For instance, when training an image classifier, data augmentation is achieved by flipping, cropping, and adjusting the brightness of each input sample, and adding these to the training set.

Gradient masking

Gradient masking is a method designed to create models that are resistant to white box probing for decision boundaries, typically in neural network-based models. Mapping a target model's decision boundaries involves crafting new input samples based on the gradient observed across outputs from previously crafted samples. Gradient masking hampers this process by creating sharper decision boundaries as illustrated below.



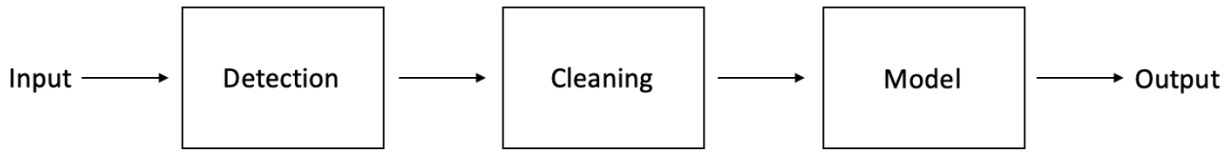
Commonly referenced techniques for masking gradients include defensive distillation and defensive dropout. Defensive distillation (Papernot et al., 2015) is a process whereby a second model is created from the output of one or more initially trained models. The second model is trained on modified Softmax ("Softmax function," 2019) output values of the first model (as opposed to the hard labels that were used to train the initial model). Dropout (Budhiraja, 2016) – the process of randomly disabling a portion of the model's cells – is a method commonly used during model training as a regularization technique to encourage models to generalize better. Defensive dropout (Wang et al., 2018) applies the dropout technique during the model inference phase. Stochastic activation pruning (Dhillon et al., 2018) is another gradient masking technique similar to defensive dropout. We note that gradient masking techniques do not make a model resistant to adversarial samples in general.



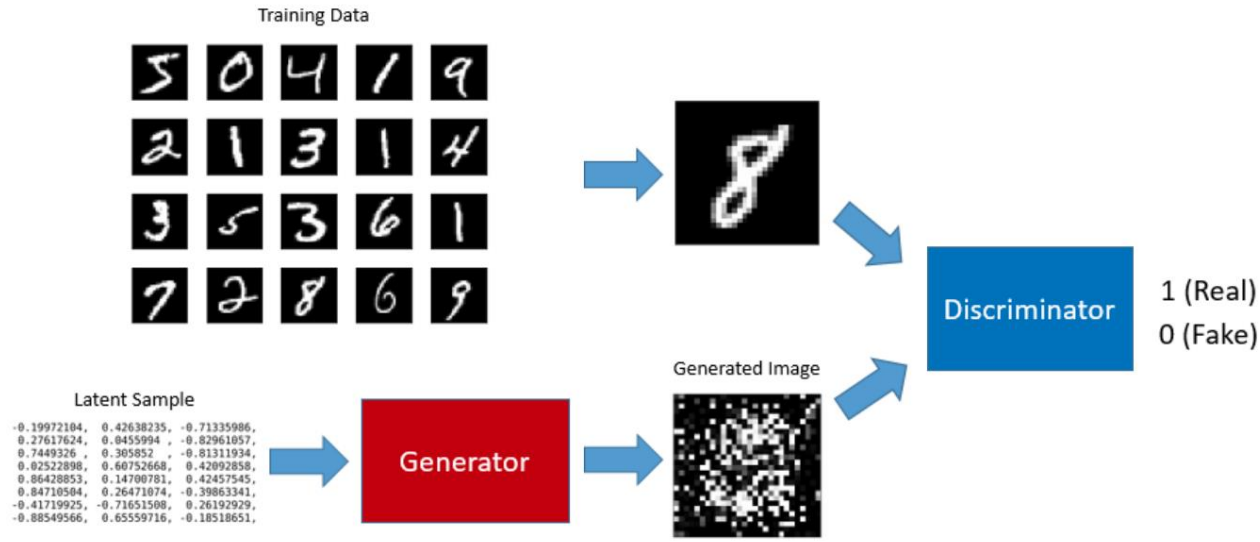
Dropout in neural networks. Source: https://medium.com/@amarbudhiraja/https-medium-com-amarbudhiraja-learning-less-to-learn-better-dropout-in-deep-machine-learning-74334da4bfc5

Detecting and cleaning adversarial inputs

A machine learning model can be shielded from adversarial inputs by placing safeguard mechanisms between the public interface to the model's input and the model itself. These mechanisms detect and clean adversarial perturbations in the raw input, prior to it reaching the model. Detection and cleaning can be performed in separate steps, or as part of a single step.



Generative Adversarial Networks (GANs) (Shibuya, 2017) are a type of machine learning model designed to generate images, or other types of data. Training a GAN involves training two neural network models simultaneously. One model, the generator, attempts to generate samples (e.g. images) from random noise. A second model, the discriminator, is fed both real samples and the samples created by the generator model. The discriminator decides which samples are real, and which are fake. As training proceeds, the generator gets better at fooling the discriminator, while the discriminator gets better at figuring out which samples are real or fake. At the end of training, the generator model will be able to accurately generate samples (for instance, convincing high-resolution photographs), and the trained discriminator model will be able to accurately detect the difference between real and fake inputs. Thus, discriminator models can be used to detect adversarial perturbations.



GAN training mechanism. Source: https://towardsdatascience.com/understanding-generative-adversarial-networks-4dafc963f2ef

Suggested cleaning methods include using the output of the GAN generator model as the input to the target model, using a separate mechanism to generate an image similar to the original input, or modifying the input image to remove perturbations.

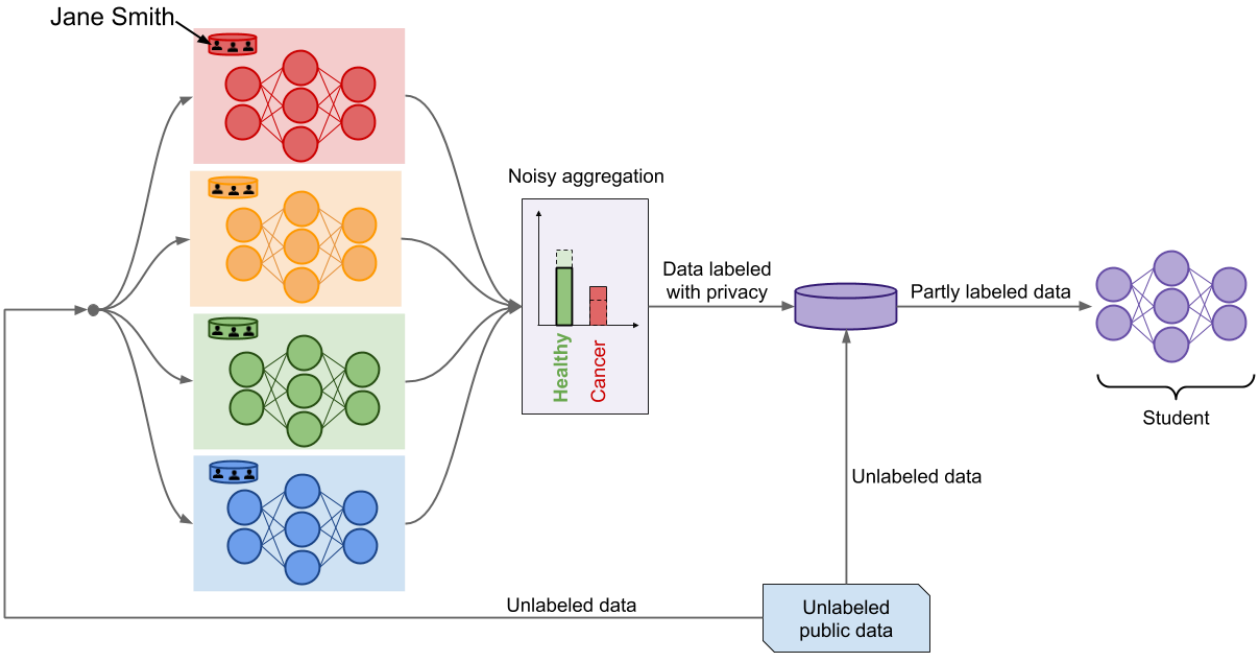
This two- (or three-) step process can actually be accomplished using a single step. Introspective neural networks are classifiers that contain built-in discriminator and generator components as part of a single end-to-end model. Trained models can be used both as a generator, and a classifier, and are resistant to adversarial inputs due to the presence of the discriminator component.

Another proposed solution (Guo et al., 2017), replaces the 'detect and clean' approach with a simple sanitization step that normalizes inputs prior to their reaching the safeguarded model.

Differential privacy

Differential privacy is a general statistical technique that aims to provide means to maximize the accuracy of query responses from statistical databases while measuring (and thereby hopefully minimizing) the privacy impact on individuals whose information is in the database. It is one proposed method for mitigating against confidentiality attacks.

One method for implementing differential privacy with machine learning models is to train a series of models against separate, unique portions of training data. At inference time, the input data is fed into each of the trained models, and a small amount of random noise is added to each model's output. The resulting values become 'votes', the highest of which becomes the output. A detailed description of differential privacy, and why it works, can be found here (Papernot and Goodfellow, 2018).



One possible implementation of differential privacy for machine learning models. Source: http://www.cleverhans.io/privacy/2018/04/29/privacy-and-machine-learning.html

Differential privacy is a hot topic at the moment, and online services such as OpenMined ("OpenMined," n.d.) have sprung up to facilitate the generation of privacy-protected models based on this technique.

Cryptographic techniques for privacy-preserving model training and inference

Cryptographic techniques are a natural choice for ensuring confidentiality and integrity, and there is a growing interest in applying those techniques to data and model protection in machine learning. Several cryptographic methods have been successfully utilized, individually and in combination, for scenarios where data and model owners are different entities which do not trust each other. The two main use cases can be Informally described as follows:

(i) model training, when multiple data owners either provide their data to a single party constructing a model or exchange parts of their data for learning a model in a distributed fashion;

(ii) inference, when a trained model is used for processing inputs provided by data owners to produce an output, such as a classification decision or a prediction.

Confidentiality of data is clearly a concern in both cases, and, in addition, model stealing concerns often need to be addressed in scenario (ii). Conceptually, both (i) and (ii) can be considered instances of the secure multi-party computation problem (secure MPC), where a number of parties want to jointly compute a function over their inputs while keeping the inputs private. This problem has been extensively studied in the cryptographic community since 1970s, and a number of protocols have recently found application in machine learning scenarios. We will now introduce several popular approaches.

Homomorphic encryption makes it possible to compute functions on encrypted data. This enables data owners to encrypt their data and send the encrypted inputs to a model owner and, possibly, other data owners. The model is then applied to the encrypted inputs (or, more generally, a desired function is computed on the encrypted inputs), and the result is communicated to appropriate parties, which can decrypt it and obtain desired information, for example, the model output in scenario (ii). So-called fully homomorphic encryption enables parties to compute a broad class of functions, covering essentially all cases of practical interest. However, all the currently developed fully homomorphic encryption techniques are very computationally expensive and their use is limited. A more practical alternative is so-called semi-homomorphic encryption methods. While those are suitable only for computing narrower classes of functions, they are utilized, usually in combination with other techniques, in several machine learning applications, for example, in collaborative filtering.

Secret sharing-based approaches can be used to distribute computation among a set of non-colluding servers, which operate on cryptographically derived shares of data owner inputs (thus, having no information about the actual inputs) and generate partial results. Such partial results can then be combined by another party to obtain the final result. One example of this approach is a privacy-preserving system for performing Principle Component Analysis developed on top of the ShareMind technology by Cybernetica.

Garbled circuit protocols, based on the *oblivious transfer* technique, are used for secure two-party computation of functions presented as Boolean circuits and can be employed in scenario (ii). In such protocols, one party prepares a garbled (encrypted) version of a circuit that implements the function to be computed, garbles their own input, and collaborates with the other party to garble their input in a privacy-preserving manner. The other party uses then the garbled circuit and inputs for computing a garbled output and collaborates with the first party to derive the actual

function output. Garbling methods are often used for privacy-preserving machine learning in combination with semi-homomorphic encryption.

There are several noteworthy limitations of the use of cryptographic techniques in machine-learning-based systems. In particular, most of such techniques are applicable only to certain (a small number of) machine learning algorithms and are computationally expensive. Besides, one has to carefully check assumptions, which the security guarantees of cryptographic methods are based on. For example, the non-collusion assumption in secret sharing-based approaches may or may not be plausible in specific applications.

Despite the limitations and challenges, a number of platforms and tools for privacy-preserving machine learning have been developed (such as Faster CryptoNets and Gazelle), and this remains a domain of active theoretical and applied research.

Defending against poisoning attacks

Poisoning attacks have been popular for many years. Some of the largest tech companies in the world put a great deal of effort into building defences against these attacks. Mitigation strategies against poisoning attacks can be grouped into four categories - rate limiting, regression testing, anomaly detection, and human intervention.

Rate limiting strategies attempt to limit the influence entities or processes have over the model or algorithm. Numerous mechanisms exist to do this. The defender can:

- Take steps to ensure that a small group of entities, including IPs or users, cannot account for a large fraction of the model training data.
- Put mechanisms in place to prevent over-weighting of false positives and false negatives reported by users.
- Limit the number of examples that each user can contribute, for instance, by the use of decaying weights.
- Slow potential attacks or suspicious activity via mechanisms such as CAPTCHA.
- Give higher weighting to registered, or 'high-quality' users.
- Calculate validity scores for registered accounts based on relevant metrics such as activity patterns, connecting IP addresses, behaviour, and so on.

In order to curb poisoning attacks, regression testing is a useful practice. It is less likely that attacks might slip through the cracks if newly trained models are checked against baseline standards. Good regression testing practices include:

- Compare each newly trained model to the previous one to estimate how much has changed. Alert on larger than expected changes and inspect the training data if an alert happens.
- Use A/B testing to compare the output of a previous and new model on the real-world inputs.
- Implement continuous testing against a dataset containing attacks and normal behavioural data that a model must accurately handle.

Anomaly detection methods can be useful in finding suspicious usage patterns. Maintainers of social networks and other online services prone to poisoning attacks should be able to implement

fairly intelligent anomaly detection methods using metadata they have available. These can include:

- IP-based anomaly detection.
- Heuristic analyses (look for 'unpopular' items that suddenly have many co-visitations with other items).
- Analysis of temporal dynamics of visits and co-visits.
- Implementation of one or more of the many proposed graph-based Sybil attack detection methods.

Although data analysis techniques and machine learning methods can be used to detect some suspicious activity, understanding how attacks are being carried out, and finding edge cases that are being abused is an activity most suited to humans. Much of the manual work required to defend against poisoning attacks relies on the creation of hand-written rules, and human moderation.

Whenever humans are involved in moderation activities and the processing of data, ethical considerations apply (Newton, 2019). Often, companies are faced with decisions (Matsakis, 2018) such as what data a human moderator can work with, how good or bad content is defined (Kennedy, 2018), how to write rules that automatically filter 'bad' content, and how to handle feedback. These issues often force companies to tread a fine line between political beliefs and definitions of free speech (Goggin, 2018). However, as long as attackers are human, it will take other humans to think as creatively as the attackers in order to defend their systems from attack. This fact will not change in the near future.

5. Conclusions

As more and more important decisions are made with the aid of machine-learning-powered systems, it will become crucial for us to be able to explain how those models make decisions, understand whether flaws or biases exist in those models, and determine whether and to what extent the outputs of those models can be affected by attacks.

The understanding of flaws and vulnerabilities inherent in the design and implementation of systems built on machine learning and the means to validate those systems and to mitigate attacks against them are still in their infancy, complicated – in comparison with traditional systems – by the lack of explainability to the user, heavy dependence on training data, and oftentimes frequent model updating. This field is attracting the attention of researchers, and is likely to grow in the coming years. As understanding in this area improves, so too will the availability and ease-of-use of tools and services designed for attacking these systems.

Complex problems can sometimes only be solved with the application of sophisticated machine learning models. However, such models are difficult to harden against attack. When designing systems that use machine learning models, engineers should carefully consider their choice of a particular architecture, based on understanding of potential attacks and on clear, reasoned trade-off decisions between model complexity, explainability, and robustness.

The use of machine learning methods and technologies are well within the capabilities of the engineers that build malware and its supporting infrastructure. Tools in the offensive cyber security space already use machine learning techniques, and these tools are as available to malicious actors as they are to security researchers and specialists. Since it is almost impossible to observe how malicious actors operate, no evidence of the use of such methods have yet been witnessed (although some speculation exists to support that possibility). Thus, we speculate that by-and-large, machine learning techniques are still not being utilized heavily for malicious purposes.

As we witness today in conventional cyber security, complex attack methodologies and tools initially developed by highly resourced threat actors, such as nation states, eventually fall into the hands of criminal organizations and then common cyber criminals. This same trend can be expected for attacks developed against machine learning models.

Text synthesis, image synthesis, and video manipulation techniques have been strongly bolstered by machine learning in recent years. Our ability to generate fake content is far ahead of our ability to detect whether content is real or faked. As such, we expect that machine-learning-powered techniques will be used for social engineering and disinformation in the near future. Disinformation created using these methods will be sophisticated, believable, and extremely difficult to refute.

It is important to bear in mind that methods of defending machine-learning-based systems against attacks and mitigating malicious use of machine learning may lead to serious ethical issues. For instance, tight security monitoring may negatively affect users' privacy and certain security response activities may weaken their autonomy.

AI researchers, engineers, businesses, regulators, policy makers, and indeed all of us will need to understand and be prepared to deal with the societal impact and diverse ethical issues that will accompany the ever-increasing presence of smart information systems in our lives.

Acknowledgements

We thank UCLan Cyprus for Quality Assurance comments and edits. We would also like to thank Matti Aksela and the Artificial Intelligence Center of Excellence team at F-Secure for comments and helpful discussions.

Bibliography

AJL -ALGORITHMIC JUSTICE LEAGUE [WWW Document], n.d. . AJL -ALGORITHMIC JUSTICE Leag. URL https://www.ajlunited.org/ (accessed 2.25.19).

AlgorithmWatch [WWW Document], n.d. . AlgorithmWatch. URL https://algorithmwatch.org (accessed 2.25.19).

algotransparency.org [WWW Document], n.d. URL http://algotransparency.org/ (accessed 2.25.19).

Alzantot, M., Sharma, Y., Elgohary, A., Ho, B.-J., Srivastava, M., Chang, K.-W., 2018. Generating Natural Language Adversarial Examples. ArXiv180407998 Cs.

Amazon SageMaker [WWW Document], n.d. . Amaz. Web Serv. Inc. URL https://aws.amazon.com/sagemaker/ (accessed 2.25.19).

american fuzzy lop [WWW Document], n.d. URL http://lcamtuf.coredump.cx/afl/ (accessed 2.25.19).

Ananny, M., 2016. Toward an Ethics of Algorithms Convening, Observation, Probability, and Timeliness. Sci. Technol. Hum. Values 41, 93–117. https://doi.org/10.1177/0162243915606523

Azure Machine Learning Service [WWW Document], n.d. URL https://azure.microsoft.com/en-us/services/machine-learning-service/ (accessed 2.25.19).

BABEL Generator [WWW Document], n.d. URL https://babel-generator.herokuapp.com/ (accessed 2.25.19).

Barocas, S., 2014. Data mining and the discourse on discrimination, in: Data Ethics Workshop, Conference on Knowledge Discovery and Data Mining.

Behzadan, V., Munir, A., 2017. Vulnerability of Deep Reinforcement Learning to Policy Induction Attacks. ArXiv170104143 Cs.

Bergen, M., De Vynck, G., Palmeri, C., 2019. Nestle, Disney Pull YouTube Ads, Joining Furor Over Child Videos.

Better Language Models and Their Implications [WWW Document], 2019. . OpenAI Blog. URL https://blog.openai.com/better-language-models/ (accessed 2.27.19).

Bolukbasi, T., Chang, K.-W., Zou, J., Saligrama, V., Kalai, A., 2016. Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings. ArXiv160706520 Cs Stat.

Booth, R., 2014. Facebook reveals news feed experiment to control emotions. The Guardian.

Bozdag, E., 2013. Bias in algorithmic filtering and personalization. Ethics Inf. Technol. 15, 209–227.

Brewster, T., 2016. Who's Better At Phishing Twitter, Me Or Artificial Intelligence? [WWW Document]. URL https://www.forbes.com/sites/thomasbrewster/2016/07/25/artificial-intelligence-phishing-twitter-bots/#6453c6e176e6 (accessed 2.25.19).

Brexit: The Uncivil War, 2019. . Wikipedia.

Britt, M.A., Rouet, J.-F., Blaum, D., Millis, K., 2019. A Reasoned Approach to Dealing With Fake News. Policy Insights Behav. Brain Sci. 6, 94–101. https://doi.org/10.1177/2372732218814855

Broderick, R., 2018. Meet The 29-Year-Old Trying To Become The King Of Mexican Fake News [WWW Document]. BuzzFeed News. URL https://www.buzzfeednews.com/article/ryanhatesthis/meet-the-29-year-old-trying-to-become-the-king-of-mexican (accessed 2.25.19).

Budhiraja, A., 2016. Learning Less to Learn Better — Dropout in (Deep) Machine learning. Medium. URL https://medium.com/@amarbudhiraja/https-medium-com-amarbudhiraja-learning-less-to-learn-better-dropout-in-deep-machine-learning-74334da4bfc5 (accessed 2.25.19).

Buffer overflow, 2019. . Wikipedia.

Burrell, J., 2016. How the machine 'thinks': Understanding opacity in machine learning algorithms. Big Data Soc. 3, 2053951715622512.

Cadwalladr, C., 2019. This is GENIUS. Yesterday @nigel_farage helped launch Turning Point UK, a US-funded far right fake student movement. Cue: a shitload of spoof Turning Point twitter accounts creating total chaos...

https://twitter.com/Neeerts/status/1091824686575640579 …. @carolecadwalla. URL https://twitter.com/carolecadwalla/status/1092150658000670721 (accessed 2.25.19).
Cadwalladr, C., Graham-Harrison, E., 2018. Revealed: 50 million Facebook profiles harvested for Cambridge Analytica in major data breach. The Guardian.
Calders, T., Kamiran, F., 2010. Classification with no discrimination by preferential sampling, in: Proc. 19th Machine Learning Conf. Belgium and The Netherlands.
Calders, T., Kamiran, F., Pechenizkiy, M., 2009. Building classifiers with independency constraints, in: Data Mining Workshops, 2009. ICDMW'09. IEEE International Conference On. IEEE, pp. 13–18.
Chaker, N., 2019. AI for Recruiting: the Good, the Bad, and the Unknown [WWW Document]. URL https://beamery.com/blog/ai-recruiting (accessed 2.26.19).
Chan, C., Ginosar, S., Zhou, T., Efros, A.A., 2018. Everybody Dance Now. ArXiv180807371 Cs.
Chaplain, C., 2019. Esther McVey tweets false claim all EU members must adopt the Euro from next year, then deletes it [WWW Document]. inews.co.uk. URL https://inews.co.uk/news/brexit/esther-mcvey-brexit-twitter-fake-claim-eu-members-euro-2020/ (accessed 3.13.19).
Chaslot, G., 2019. YouTube announced they will stop recommending some conspiracy theories such as flat earth. I worked on the AI that promoted them by the *billions*. Here is why it's a historic victory. Thread. 1/https://bit.ly/2MMXNGn. @gchaslot. URL https://twitter.com/gchaslot/status/1094359564559044610 (accessed 2.25.19).
Chollet, F., 2018. What worries me about AI. Fr. Chollet. URL https://medium.com/@francois.chollet/what-worries-me-about-ai-ed9df072b704 (accessed 2.25.19).
Claburn, T., 2017. Lyrebird steals your voice to make you say things you didn't – and we hate this future [WWW Document]. URL https://www.theregister.co.uk/2017/04/24/voice_stealing_lyrebird/ (accessed 2.26.19).
Clifton, J., Butler, O., 2017. I Made My Shed the Top Rated Restaurant On TripAdvisor. Vice. URL https://www.vice.com/en_uk/article/434gqw/i-made-my-shed-the-top-rated-restaurant-on-tripadvisor (accessed 2.26.19).
Cloud ML Engine [WWW Document], n.d. . Google Cloud. URL https://cloud.google.com/ml-engine/ (accessed 2.25.19).
Confessore, N., Dance, G.J.X., Harris, R., Hansen, M., 2018. The Follower Factory. N. Y. Times.
Context-free grammar, 2019. . Wikipedia.
Crawford, K., 2016. Can an algorithm be agonistic? Ten scenes from life in calculated publics. Sci. Technol. Hum. Values 41, 77–92.
Crawford, K., 2013. The Hidden Biases in Big Data [WWW Document]. Harv. Bus. Rev. URL https://hbr.org/2013/04/the-hidden-biases-in-big-data (accessed 1.4.17).
CycleGAN Project Page [WWW Document], n.d. URL https://junyanz.github.io/CycleGAN/ (accessed 2.25.19).
Dam, G., 2019. Senator @marcorubio, an important transformer exploded in Bolívar and that, in part, again collapsed the Venezuelan Electric System; however it was not in a dam, much less german. My name is Germán Dam, I am one of the journalists who published the information. https://twitter.com/marcorubio/status/1104506183484870657 …. @GEDV86. URL https://twitter.com/GEDV86/status/1104511307259281409 (accessed 3.13.19).
Day, S.E., 2019. I'm an Engineer who loves America. I just figured something out I want to talk about it. Let's talk about weaponized bots, algorithm exploitation, countermeasures, and

counter-countermeasures. <A MEGATHREAD>. @smartereveryday. URL https://twitter.com/smartereveryday/status/1091833011262423040 (accessed 2.25.19).
Decision tree, 2019. . Wikipedia.
Deep learning, 2019. . Wikipedia.
Denial-of-service attack, 2019. . Wikipedia.
Dhillon, G.S., Azizzadenesheli, K., Lipton, Z.C., Bernstein, J., Kossaifi, J., Khanna, A., Anandkumar, A., 2018. Stochastic Activation Pruning for Robust Adversarial Defense. ArXiv180301442 Cs Stat.
DiResta, R., 2019. How Amazon's Algorithms Curated a Dystopian Bookstore. Wired.
Dual-use technology, 2019. . Wikipedia.
Engstrom, L., Tran, B., Tsipras, D., Schmidt, L., Madry, A., 2017. A Rotation and a Translation Suffice: Fooling CNNs with Simple Transformations. ArXiv171202779 Cs Stat.
EU Parliament, 2016. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation) (Text with EEA relevance), OJ L.
Europe, C. of, 2019. Council of Europe warns about the risk of algorithmic processes being used to manipulate social and political behaviours [WWW Document]. URL https://search.coe.int/directorate_of_communications/Pages/result_details.aspx?ObjectId=090000168092de6a (accessed 2.26.19).
FAT ML [WWW Document], n.d. URL http://www.fatml.org/ (accessed 3.4.19).
Ferenstein, G., 2014. Predicting Love And Breakups With Facebook Data. TechCrunch. URL http://social.techcrunch.com/2014/02/14/facebook-love-data/ (accessed 2.25.19).
Fussell, S., 2017. Why Can't This Soap Dispenser Identify Dark Skin? [WWW Document]. Gizmodo. URL https://gizmodo.com/why-cant-this-soap-dispenser-identify-dark-skin-1797931773 (accessed 2.11.19).
Gallagher, E., 2019a. Social media automation & information warfare by the Venezuelan opposition [WWW Document]. Noteworthy - J. Blog. URL https://blog.usejournal.com/social-media-automation-information-warfare-by-the-venezuelan-opposition-9cdb407492f8 (accessed 2.25.19).
Gallagher, E., 2019b. Mexico: Coordinated inauthentic behavior on Facebook & Twitter. Medium. URL https://medium.com/@erin_gallagher/mexico-coordinated-inauthentic-behavior-on-facebook-twitter-a670280d02fc (accessed 2.25.19).
Gallagher, E., 2019c. Nobody told them that bridge has never been open I guess. https://twitter.com/RepDWStweets/status/1104813875747278848 …. @3r1nG. URL https://twitter.com/3r1nG/status/1105097557972275203 (accessed 3.13.19).
Gallagher, E., 2019d. I recently tweeted that there has been a social media fog hanging over Venezuela for a long time... here is why I said that:https://medium.com/@erin_gallagher/social-media-automation-information-warfare-by-the-venezuelan-opposition-9cdb407492f8 …. @3r1nG. URL https://twitter.com/3r1nG/status/1090751257915084802 (accessed 2.25.19).
Gallagher, E., 2019e. So far, I've found 70 MAGA accounts that tweet hundreds of times per day. 49 of them have profile photos of women. Collectively, they have over 4.1 million followers and tweet on average 20,192 tweets per day. Chart of the top 18, together they average 10,873 tweets per day:pic.twitter.com/3mLMWR3Nqw. @3r1nG. URL https://twitter.com/3r1nG/status/1087077553863618566 (accessed 2.25.19).

Gallagher, E., 2017. Fake Honduran Twitter: the digital campaign against Berta Cáceres and COPINH. Medium. URL https://medium.com/@erin_gallagher/fake-honduran-twitter-the-digital-campaign-against-berta-c%C3%A1ceres-and-copinh-3d1ea62e61ab (accessed 2.25.19).

Gallagher, E., n.d. Erin Gallagher [WWW Document]. Medium. URL https://medium.com/@erin_gallagher (accessed 2.26.19).

Gebru, T., Morgenstern, J., Vecchione, B., Vaughan, J.W., Wallach, H., Daumeé III, H., Crawford, K., 2018. Datasheets for Datasets. ArXiv180309010 Cs.

Geoghegan, P., 2019. Revealed: The dark-money Brexit ads flooding social media [WWW Document]. openDemocracy. URL https://www.opendemocracy.net/uk/brexitinc/peter-geoghegan/revealed-dark-money-brexit-ads-flooding-social-media (accessed 2.25.19).

Gershgorn, D., n.d. Companies are on the hook if their hiring algorithms are biased [WWW Document]. Quartz. URL https://qz.com/1427621/companies-are-on-the-hook-if-their-hiring-algorithms-are-biased/ (accessed 2.25.19a).

Gershgorn, D., n.d. Facebook says it has a tool to detect bias in its artificial intelligence [WWW Document]. Quartz. URL https://qz.com/1268520/facebook-says-it-has-a-tool-to-detect-bias-in-its-artificial-intelligence/ (accessed 2.25.19b).

Glenn, T., Monteith, S., 2014. New measures of mental state and behavior based on data collected from sensors, smartphones, and the Internet. Curr. Psychiatry Rep. 16, 523.

Goggin, B., 2018. A top Patreon creator deleted his account, accusing the crowdfunding membership platform of “political bias” after it purged conservative accounts it said were associated with hate groups [WWW Document]. URL https://nordic.businessinsider.com/sam-harris-deletes-patreon-account-after-platform-boots-conservatives-2018-12/ (accessed 2.26.19).

Guardian, T., n.d. The Cambridge Analytica Files. The Guardian.

Guo, C., Rana, M., Cisse, M., van der Maaten, L., 2017. Countering Adversarial Images using Input Transformations. ArXiv171100117 Cs.

hashcat - advanced password recovery [WWW Document], n.d. URL https://hashcat.net/hashcat/ (accessed 2.25.19).

Hildebrandt, M., 2011. Who Needs Stories if You Can Get the Data? ISPs in the Era of Big Number Crunching. Philos. Technol. 24, 371–390. https://doi.org/10.1007/s13347-011-0041-8

Hildebrandt, M., Koops, B.-J., 2010. The challenges of ambient law and legal protection in the profiling era. Mod. Law Rev. 73, 428–460.

Holland, S., Hosny, A., Newman, S., Joseph, J., Chmielinski, K., 2018. The Dataset Nutrition Label: A Framework To Drive Higher Data Quality Standards. ArXiv180503677 Cs.

Hope, D., 2018. How AI Is Transforming Lending And Loan Management [WWW Document]. SmartData Collect. URL https://www.smartdatacollective.com/how-ai-is-transforming-lending-and-loan-management/ (accessed 2.26.19).

Huang, S., Papernot, N., Goodfellow, I., Duan, Y., Abbeel, P., 2017. Adversarial Attacks on Neural Network Policies. ArXiv170202284 Cs Stat.

IBM, 2018. IBM Adversarial Robustness Toolbox [WWW Document]. IBM Res. Blog. URL https://www.ibm.com/blogs/research/2018/04/ai-adversarial-robustness-toolbox/ (accessed 2.26.19).

IBM/adversarial-robustness-toolbox [WWW Document], n.d. . GitHub. URL https://github.com/IBM/adversarial-robustness-toolbox (accessed 2.26.19).

Ilyas, A., Engstrom, L., Athalye, A., Lin, J., 2018. Black-box Adversarial Attacks with Limited Queries and Information. ArXiv180408598 Cs Stat.

International, P., 2017. Texas Media Company Hired By Trump Created Kenyan President's Viral "Anonymous" Attack Campaign Against Rival, New Investigation Reveals [WWW Document]. Priv. Int. URL http://privacyinternational.org/feature/954/texas-media-company-hired-trump-created-kenyan-presidents-viral-anonymous-attack (accessed 3.14.19).

Internet of things, 2019. . Wikipedia.

Jánošík, J., 2019. ML-era in cybersecurity: A step toward a safer world or the brink of chaos? [WWW Document]. WeLiveSecurity. URL https://www.welivesecurity.com/2019/02/22/ml-era-cybersecurity-step-toward-safer-world-brink-chaos/ (accessed 2.27.19).

Johnson, J.A., 2006. Technology and pragmatism: From value neutrality to value criticality. Soc. Sci. Res. Netw. Retrieved Httppapers Ssrn Comabstract 2154654.

Kao, J., 2017. More than a Million Pro-Repeal Net Neutrality Comments were Likely Faked [WWW Document]. Hacker Noon. URL https://hackernoon.com/more-than-a-million-pro-repeal-net-neutrality-comments-were-likely-faked-e9f0e3ed36a6 (accessed 2.25.19).

Karpathy, A., 2015. The Unreasonable Effectiveness of Recurrent Neural Networks [WWW Document]. URL http://karpathy.github.io/2015/05/21/rnn-effectiveness/ (accessed 2.12.19).

Kennedy, J., 2018. Facebook Dublin embroiled in scandal over content moderation failures [WWW Document]. Silicon Repub. URL https://www.siliconrepublic.com/companies/facebook-cpl-content-moderation-dublin (accessed 2.26.19).

Kim, H., Garrido, P., Tewari, A., Xu, W., Thies, J., Nießner, M., Pérez, P., Richardt, C., Zollhöfer, M., Theobalt, C., 2018. Deep Video Portraits. ArXiv180511714 Cs.

Kitchin, R., 2017. Thinking critically about and researching algorithms. Inf. Commun. Soc. 20, 14–29.

Kleinman, Z., 2018. IBM launches bias detector for AI.

Kloft, M., Laskov, P., n.d. A "Poisoning" Attack Against Online Anomaly Detection 2.

Knight, W., n.d. Microsoft is creating an oracle for catching biased AI algorithms [WWW Document]. MIT Technol. Rev. URL https://www.technologyreview.com/s/611138/microsoft-is-creating-an-oracle-for-catching-biased-ai-algorithms/ (accessed 2.25.19).

Kraemer, F., Van Overveld, K., Peterson, M., 2011. Is there an ethics of algorithms? Ethics Inf. Technol. 13, 251–260.

Kuleshov, V., Thakoor, S., Lau, T., Ermon, S., 2018. Adversarial Examples for Natural Language Classification Problems.

Kurakin, A., Goodfellow, I., Bengio, S., 2016. Adversarial examples in the physical world. ArXiv160702533 Cs Stat.

Kwon, D., 2017. Can Facebook's Machine-Learning Algorithms Accurately Predict Suicide? [WWW Document]. Sci. Am. URL https://www.scientificamerican.com/article/can-facebooks-machine-learning-algorithms-accurately-predict-suicide/ (accessed 2.25.19).

Lapowsky, N.T., Issie, 2018. How Russian Trolls Used Meme Warfare to Divide America. Wired.

Leese, M., 2014. The new profiling: Algorithms, black boxes, and the failure of anti-discriminatory safeguards in the European Union. Secur. Dialogue 45, 494–511.

Logistic regression, 2019. . Wikipedia.

Lowe, R., 2019. OpenAI's GPT-2: the model, the hype, and the controversy [WWW Document]. Data Sci. URL https://towardsdatascience.com/openais-gpt-2-the-model-the-hype-and-the-controversy-1109f4bfd5e8?sk=bc319cebc22fe0459574544828c84c6d (accessed 2.27.19).

Macnish, K., 2015. An Eye for an Eye: Proportionality and Surveillance. Ethical Theory Moral Pract. 18, 529–548. https://doi.org/10.1007/s10677-014-9537-5
Macnish, K., 2012. Unblinking eyes: the ethics of automating surveillance. Ethics Inf. Technol. 14, 151–167. https://doi.org/10.1007/s10676-012-9291-0
MacWhirter, J., 2019. YouTube removes advert for far-right Britain First. The Guardian.
Markov chain, 2019. . Wikipedia.
Matsakis, L., 2018. YouTube's Content Moderation Is an Inconsistent Mess. Wired.
Mayer, J., 2018. New Evidence Emerges of Steve Bannon and Cambridge Analytica's Role in Brexit.
Mayer-Schonberger, V., Cukier, K., 2017. Big Data: The Essential Guide to Work, Life and Learning in the Age of Insight. John Murray, London.
Mitchell, M., Wu, S., Zaldivar, A., Barnes, P., Vasserman, L., Hutchinson, B., Spitzer, E., Raji, I.D., Gebru, T., 2019. Model Cards for Model Reporting. Proc. Conf. Fairness Account. Transpar. - FAT 19 220–229. https://doi.org/10.1145/3287560.3287596
Mittelstadt, B.D., Allo, P., Taddeo, M., Wachter, S., Floridi, L., 2016. The ethics of algorithms: Mapping the debate. Big Data Soc. 3, 2053951716679679. https://doi.org/10.1177/2053951716679679
Naik, G., Bhide, S.S., 2014. Will the future of knowledge work automation transform personalized medicine? Appl. Transl. Genomics 3, 50–53.
Nakashima, E., Harris, S., 2018. How the Russians hacked the DNC and passed its emails to WikiLeaks. Wash. Post.
Neff, G., 2019. Silicon Valley engineers say about AI bias "We can fix this". No, no you can't. Really. This problem is bigger than you, and the solutions are social, not just one team of engineers, one company, or one country. @ginasue. URL https://twitter.com/ginasue/status/1097921774174552066 (accessed 2.26.19).
Newell, S., Marabelli, M., 2015. Strategic opportunities (and challenges) of algorithmic decision-making: A call for action on the long-term societal effects of 'datification.' J. Strateg. Inf. Syst. 24, 3–14.
Newlin, B., 2019. Seattle man who stabbed his brother to death with 4-foot sword thought he was a lizard, police say [WWW Document]. WDIV. URL https://www.clickondetroit.com/news/national/seattle-man-who-stabbed-his-brother-to-death-with-4-foot-sword-thought-he-was-a-lizard-police-say (accessed 2.25.19).
Newton, C., 2019. The secret lives of Facebook moderators in America [WWW Document]. The Verge. URL https://www.theverge.com/2019/2/25/18229714/cognizant-facebook-content-moderator-interviews-trauma-working-conditions-arizona (accessed 2.26.19).
Neyland, D., 2016. Bearing account-able witness to the ethical algorithmic system. Sci. Technol. Hum. Values 41, 50–76.
Nissenbaum, H.F., 2009. Privacy in Context: Technology, Policy, and the Integrity of Social Life. Stanford University Press, Stanford, Calif.
Norteño, C., 2019. Check out #RegimeChange4France, the latest goofy invention of far-right Twitter. This appears to be (among other things) a pro-LePen hashtag campaign being conducted largely by individuals who do not actually vote in France. #AltWankers cc: @ZellaQuixotepic.twitter.com/nfsyRIn3wr. @conspirator0. URL https://twitter.com/conspirator0/status/1090444227589427201 (accessed 2.25.19).
NVIDIA, 2019. Synthesizing and manipulating images with conditional GANs. NVIDIA Corporation.
Oberoi, G., 2018. Exploring DeepFakes [WWW Document]. Gaurav Oberoi. URL https://goberoi.com/exploring-deepfakes-20c9947c22d9 (accessed 2.25.19).

O'Kane, S., 2019. The things that happen at the fringes of Tesla world are so weird. Like today when a twitter account for a supposed "Senior Journalist at Bloomberg" named "Maisy Kinsley" followed a few too many Tesla short-sellers. Account is already suspended.pic.twitter.com/3GO9Uhf9aH. @sokane1. URL https://twitter.com/sokane1/status/1111023838467362816 (accessed 3.28.19).

OpenAI, 2019. We've trained an unsupervised language model that can generate coherent paragraphs and perform rudimentary reading comprehension, machine translation, question answering, and summarization — all without task-specific training: https://blog.openai.com/better-language-models/ …pic.twitter.com/360bGgoea3. @OpenAI. URL https://twitter.com/OpenAI/status/1096092704709070851 (accessed 2.27.19).

OpenMined [WWW Document], n.d. URL https://www.openmined.org/ (accessed 2.26.19).

Orphanides, K.G., 2019. On YouTube, a network of paedophiles is hiding in plain sight. Wired UK.

Papernot, N., Goodfellow, I., 2018. Privacy and machine learning: two unexpected allies? [WWW Document]. Cleverhans-Blog. URL http://cleverhans.io/privacy/2018/04/29/privacy-and-machine-learning.html (accessed 2.26.19).

Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z.B., Swami, A., 2016. Practical Black-Box Attacks against Machine Learning. ArXiv160202697 Cs.

Papernot, N., McDaniel, P., Wu, X., Jha, S., Swami, A., 2015. Distillation as a Defense to Adversarial Perturbations against Deep Neural Networks. ArXiv151104508 Cs Stat.

Phishing, 2019. . Wikipedia.

Plaut, M., 2018. Cambridge Analytica and the digital war in Africa [WWW Document]. URL https://www.newstatesman.com/world/2018/03/cambridge-analytica-facebook-elections-africa-kenya (accessed 2.25.19).

Police warned about using algorithms, 2017.

Ray, T., 2019. Fear not deep fakes: OpenAI's machine writes as senselessly as a chatbot speaks [WWW Document]. ZDNet. URL https://www.zdnet.com/article/fear-not-deep-fakes-openais-machine-writes-like-a-chatbot-speaks-senselessly/ (accessed 2.27.19).

Raymond, A., 2014. The Dilemma of Private Justice Systems: Big Data Sources, the Cloud and Predictive Analytics (SSRN Scholarly Paper No. ID 2469291). Social Science Research Network, Rochester, NY.

Regan, P., 2002. Privacy as a Common Good in the Digital World. Inf. Commun. Soc. 5, 382–405.

Reigstad, L., 2017. How an Austin Ad Agency Helped the Alt-Right Rise Again in Germany [WWW Document]. Tex. Mon. URL https://www.texasmonthly.com/the-daily-post/how-an-austin-ad-agency-helped-the-alt-right-rise-again-in-germany/ (accessed 3.14.19).

Responsible AI Licenses (RAIL) [WWW Document], n.d. . Responsible AI Licens. RAIL. URL https://www.licenses.ai/ (accessed 3.4.19).

Riemer, K., n.d. On Rewarding 'Bullshit': Algorithms Should Not Be Grading Essays [WWW Document]. Undark. URL https://undark.org/article/rewarding-bullshit-algorithms-classroom/ (accessed 2.25.19).

Robitzski, D., 2018. AI can now manipulate people's movements in fake videos [WWW Document]. Futurism. URL https://futurism.com/ai-can-now-manipulate-peoples-movements-in-fake-videos (accessed 2.26.19).

Roessler, B., Mokrosinska, D. (Eds.), 2015. Social Dimensions of Privacy: Interdisciplinary Perspectives. Cambridge University Press, New York.

Sandvig, C., Hamilton, K., Karahalios, K., Langbort, C., 2014. Auditing algorithms: Research methods for detecting discrimination on internet platforms. Data Discrim. Convert. Crit. Concerns Product. Inq. 1–23.
Schermer, B.W., 2011. The limits of privacy in automated profiling and data mining. Comput. Law Secur. Rev. 27, 45–52.
Schönherr, L., Kohls, K., Zeiler, S., Holz, T., Kolossa, D., 2018. Adversarial Attacks Against Automatic Speech Recognition Systems via Psychoacoustic Hiding. ArXiv180805665 Cs Eess.
Scott, M., 2018. Cambridge Analytica helped 'cheat' Brexit vote and US election, claims whistleblower [WWW Document]. POLITICO. URL https://www.politico.eu/article/cambridge-analytica-chris-wylie-brexit-trump-britain-data-protection-privacy-facebook/ (accessed 2.25.19).
Shibuya, N., 2017. Understanding Generative Adversarial Networks [WWW Document]. Data Sci. URL https://towardsdatascience.com/understanding-generative-adversarial-networks-4dafc963f2ef (accessed 2.27.19).
Shushman, C., Lee, M., Kurenkov, A., 2019. In Favor of Developing Ethical Best Practices in AI Research [WWW Document]. SAIL Blog. URL http://ai.stanford.edu/blog/ethical_best_practices/ (accessed 3.4.19).
Softmax function, 2019. . Wikipedia.
Solomon, S., 2018. Cambridge Analytica Played Roles in Multiple African Elections [WWW Document]. VOA. URL https://www.voanews.com/a/cambridge-analytica-played-roles-in-multiple-african-elections/4309792.html (accessed 2.25.19).
Solove, D.J., 2002. Conceptualizing Privacy. Calif. Law Rev. 90, 1087–1155.
Speer, R., 2017. How to make a racist AI without really trying [WWW Document]. ConceptNet Blog. URL http://blog.conceptnet.io/posts/2017/how-to-make-a-racist-ai-without-really-trying/ (accessed 2.25.19).
SpinnerChief [WWW Document], n.d. URL http://www.spinnerchief.com/ (accessed 2.25.19).
Stark, M., Fins, J.J., 2013. Engineering medical decisions: computer algorithms and the manipulation of choice. Camb. Q. Healthc. Ethics 22, 373–381.
Stoecklin, M., 2018. DeepLocker: How AI Can Power a Stealthy New Breed of Malware. Secur. Intell. URL https://securityintelligence.com/deeplocker-how-ai-can-power-a-stealthy-new-breed-of-malware/ (accessed 2.26.19).
Stuchbery, M., 2019. The UK right loves this youth group. But it has a worrying US history | Mike Stuchbery. The Guardian.
Stuxnet, 2019. . Wikipedia.
Su, J., Vargas, D.V., Kouichi, S., 2017. One pixel attack for fooling deep neural networks. ArXiv171008864 Cs Stat.
Sybil attack, 2019. . Wikipedia.
takaesu, isao, 2019. Source code about machine learning and security. Contribute to 13o-bbr-bbq/machine_learning_security development by creating an account on GitHub.
Tashea, J., 2017. Courts Are Using AI to Sentence Criminals. That Must Stop Now. Wired.
tensorflow/cleverhans [WWW Document], n.d. . GitHub. URL https://github.com/tensorflow/cleverhans (accessed 2.26.19).
Thalen, M., 2019. Mikael Thalen on Twitter: "I've gone down a black hole of the latest DeepFakes and this mashup of Steve Buscemi and Jennifer Lawrence is a sight to behold… https://t.co/nK4tXnwWS6" [WWW Document]. URL https://twitter.com/MikaelThalen/status/1090349932266094593 (accessed 2.12.19).

Thielman, S., 2016. Yahoo hack: 1bn accounts compromised by biggest data breach in history. The Guardian.

This Person Does Not Exist [WWW Document], n.d. URL https://thispersondoesnotexist.com/ (accessed 2.27.19).

Tutt, A., 2016. An FDA for algorithms.

Van Wel, L., Royakkers, L., 2004. Ethical issues in web data mining. Ethics Inf. Technol. 6, 129–140.

Wang, S., Wang, X., Zhao, P., Wen, W., Kaeli, D., Chin, P., Lin, X., 2018. Defensive Dropout for Hardening Deep Neural Networks under Adversarial Attacks. Proc. Int. Conf. Comput.-Aided Des. - ICCAD 18 1–8. https://doi.org/10.1145/3240765.3264699

Wang, T.-C., Liu, M.-Y., Zhu, J.-Y., Tao, A., Kautz, J., Catanzaro, B., 2017. High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs. ArXiv171111585 Cs.

Wexler, J., n.d. The What-If Tool: Code-Free Probing of Machine Learning Models. Google AI Blog. URL http://ai.googleblog.com/2018/09/the-what-if-tool-code-free-probing-of.html (accessed 2.25.19).

WhiteHatBox [WWW Document], n.d. URL https://www.whitehatbox.com/ (accessed 2.25.19).

Wiener-Bronner, D., 2014. More Computer-Generated Nonsense Papers Pulled From Science Journals [WWW Document]. The Atlantic. URL https://www.theatlantic.com/technology/archive/2014/03/more-computer-generated-nonsense-papers-pulled-science-journals/358735/ (accessed 2.25.19).

Zarsky, T., 2016. The trouble with algorithmic decisions: An analytic road map to examine efficiency and fairness in automated and opaque decision making. Sci. Technol. Hum. Values 41, 118–132.

Zarsky, T., 2013. Transparent Predictions (SSRN Scholarly Paper No. ID 2324240). Social Science Research Network, Rochester, NY.

Zhang, C., 2018. How to generate realistic yelp restaurant reviews with Keras | DLology [WWW Document]. URL https://www.dlology.com/blog/how-to-generate-realistic-yelp-restaurant-reviews-with-keras/ (accessed 2.25.19).

Zhu, J.-Y., Park, T., Isola, P., Efros, A.A., 2017. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. ArXiv170310593 Cs.

Zwetsloot, R., Dafoe, A., 2019. Thinking About Risks From AI: Accidents, Misuse and Structure [WWW Document]. Lawfare. URL https://www.lawfareblog.com/thinking-about-risks-ai-accidents-misuse-and-structure (accessed 3.4.19).