Augmenting Machine Learning with Argumentation

Matt Bishop
University of California at Davis
mabishop@ucdavis.edu

Carrie Gates
Securelytix
cgates@securelytix.com

Karl Levitt
University of California at Davis
knlevitt@ucdavis.edu

ABSTRACT

The information security community is haunted by the failure of an appropriate break-the-glass access control at the United States Center for Disease Control that led to an estimated additional 1.2 million deaths in North America in 2036. In this paper we review what caused the security failures in this system and argue that, by combining human intelligence with multiple technological approaches to create a system that emphasizes human approaches to guide analysis, the failures that occurred will not recur. We also leverage people and technologies to identify and fill gaps in the training data to minimize the threat of unexpected events. While we use this scenario as our running example, we note that our approach is generalizable to a broader problem space where machine learning approaches have been deployed to make decisions.

CCS CONCEPTS

• **Security and privacy → Access control**; *Information flow control*; *Usability in security and privacy*;

KEYWORDS

break the glass, argumentation, pandemic

ACM Reference Format:
Matt Bishop, Carrie Gates, and Karl Levitt. 2018. Augmenting Machine Learning with Argumentation. In *New Security Paradigms Workshop (NSPW '18), August 28–31, 2018, Windsor, United Kingdom.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3285002.3285005

1 INTRODUCTION

Machine learning has been deployed in numerous contexts over the past two decades. Abstracting away from the specific problem domains, machine learning has been used to classify data (such as in facial recognition systems), discover previously unknown associations (such as in data exploration), and to make recommendations based on historical data (such as in recommender systems for sales). One example problem domain where machine learning has been deployed is in break-the-glass systems, which we use as our running example through-out this paper.

Break the glass systems have been deployed for nearly a decade. These systems provide the ability to circumvent access control systems in the case of emergencies. Originally conceived of as a necessary extension to access control, they have evolved to become intelligent systems that determine not only that the access control system needs to be over-ridden, but also determine who has the appropriate credentials to over-rule the access control restrictions.

Current break the glass systems are designed based on deep learning neural networks, having learned the appropriate escalation responses by using previous emergencies as input. Deployed systems have been trained on a wide range of emergency cases within their particular vertical. For example, and relevant to this paper, is that systems used within the medical community have all been trained using emergencies and break the glass situations within that community. Once trained on the generic cases, systems are specialized with further training examples specific to their deployed environment and based on the access control systems within that environment.

Sadly, this system suffered a catastrophic security failure during the 2036 infection, leaving an additional 1.2 million Americans dead because of the black-box nature of the system. A post-mortem on this event showed that the system in place at the CDC had learned to find the nearest person with appropriate credentials to perform any break the glass actions, rather than finding the most appropriate person based on the situation at hand. While this approach is generally sufficient for most emergencies, in this case it resulted in the delay and incorrect distribution of vaccines.

We review what caused the security failures in this system and propose using argumentation and other artificial intelligence methods augmented by human intelligence rather than black-box artificial intelligence learning systems to make decisions in break-the-glass scenarios for high-risk environments.

Argumentation is a model where a system evaluates different "arguments" to determine the best one. Given a scenario, the system puts forward an argument and the resulting conclusion. A counter-argument is then provided, along with a new conclusion. The system continues in this fashion until all scenarios (arguments) have been exhausted. The best argument is then chosen based on the context of the scenario.

In the next section, we describe the details of the 2036 infection. We then provide relevant background and literature in Section 3, and an analysis of the failures of the system. Section 4 presents an alternative approach. Section 5 describes how this approach would be applied to the 2036 scenario. Underlying problems are outlined in Section 6, as well as suggestions for reducing or coping with them. We compare our approach to related approaches in Section 7. We propose generalizations in Section 8 and then conclude in Section 9.

2 2036 INFECTION

In 2036, doctors reported an increase in an illness that initially had the same symptoms as the avian flu. However, after approximately a month, the symptoms worsened, and within two months, around

NSPW '18, August 28–31, 2018, Windsor, United Kingdom

ACM ISBN 978-1-4503-6597-0/18/08…$15.00
https://doi.org/10.1145/3285002.3285005

65% of the people infected died. The disease spread rapidly, with thousands of death and illnesses.

At that point, the Centers for Disease Control became involved. Determining how the disease spread took considerable time; unlike avian flu, most of the patients appeared to have had neither direct nor indirect contact with infected animals or contaminated environments.

The first approach to treating the disease, prevention, required developing a vaccine to increase immunity of the disease. Such a vaccine typically takes a long time to develop. The alternative, which can be done much more quickly, is to develop an anti-viral drug to treat those already ill. Given the numbers of people with the disease, the CDC co-ordinated development of this drug, which was based on a similar but less virulent disease. It did not stop the disease, but it kept infected people alive. The anti-viral medication was developed within a few months.

As the CDC got ready to distribute the anti-viral drug, the epidemiology experts deployed around the country reported data leading the CDC to determine that the illness seemed to follow patterns in the weather. The CDC began to co-ordinate with emergency response groups to manage the distribution of the drug. As is standard practice, all the distribution workers were given the anti-viral drug. Unfortunately, the drug did not prevent the illness; it simply saved the patient's life. So many workers got very sick with the virus. Although they would not die, they were incapacitated during the critical time when the anti-viral drug had to be distributed.

The CDC had recently adopted an automated system to assist with the distribution of medicine; the system would prepare the shipping labels, maps, and distribution guides so that, when required, the medicine could be distributed quickly. But access to medicines to be used in a crisis was tightly controlled to ensure it was distributed appropriately.

In this case, the AI system used a "break-the-glass" algorithm to determine whom to contact to authorize the distribution. In the past, all such authorizing people were physically present when they acted, so the algorithm used examined who was closest to the medicine, so they could immediately begin the distribution. In this case, the list began with three people; if none of these could be reached, it would proceed to the next group. The first was a doctor who was an expert in infectious diseases. The second was a medical researcher, in this particular case the one who identified the similar disease leading to the creation of the anti-viral drug. Farther down the list was the epidemiologist who determined the spread was correlated with weather patterns, followed by a meteorologist who worked on predicting changes in weather. She worked at a site about 5 hours away from the distribution point.

Performing its analysis, the AI system determined the doctor was physically closest to the distribution point, and could get there much more quickly than anyone else. So it notified her that she needed to open the area and begin sending the medicine out. She did so. It is estimated that the anti-viral drug saved over 500,000 lives.

A retrospective root cause analysis of the CDC's performance examined the co-ordination of the manufacturing of the anti-viral drug and its distribution. The analysis showed the former worked as well as possible. But the latter did not. The distribution pattern did not reflect changes in the spread of the disease due to the delay in sending the medicine out - a short delay, but a critical one. Had the meteorologist, who understood how weather patterns would change, been the distributor, it is estimated that by changing the planned pattern of distribution, about 1,500,000 lives would have been saved.

The problem was that the AI system used machine learning to determine whom to call in the break-the-glass scenario. In the past, the person closest to the distribution point was the right person, as the goal was to get the medicine out quickly, and the spread was not affected by short delays. In this case, though, that assumption, and history, played false.

3 BACKGROUND

The break-the-glass concept in access control was first introduced by Povey in 1999 [31]. At this point the concept was a suggestion (embedded in the concept of "optimistic security"), but since became more standard, particularly in literature relating to access to health records. By 2006, there were implementations of this concept in healthcare systems [7]. We note that Ferreira *et al.* [7] stated at that time that "user intervention in defining security procedures is crucial to its successful implementation and use." Extensions to break-the-glass concepts were subsequently centered around changing environments, such as moving health records into cloud-based storage and supporting alternative information delivery models [19].

Two key papers emerged during this time frame, both initially relatively obscure. The first provided additional context around break-the-glass scenarios by including time and location into the access control model [9]; however, this model still utilized role hierarchies in addition to spatiotemporal information in order to determine who had permissions for breaking the glass in any given scenario. The second paper [20] defined a language that allowed one to infer knowledge gaps and knowledge conflicts, thus providing more context around break-the-glass scenarios. A third paper was published shortly thereafter that was prescient, but that never made an impact — Bishop et al. [3] published a paper that argued against using black box approaches to life-and-death situations.

The late 2010's saw a revolution in the usage of machine learning and artificial intelligence models. These models moved from academia into mainstream applications, fueling recommender systems, advertising targeting, political campaigns, and information security systems. Health care systems did not remain untouched by this, and researchers started building on the spatiotemporal break-the-glass [9] and the language inference [20] papers using standard intelligence approaches. The first breakthrough came in 2022 with a paper by Sutton [37] that provided a comprehensive review of health care situations where break-the-glass was employed, resulting in a dataset that could be leveraged by machine learning researchers. A first approach that leveraged Naive Bayes was used to predict if the current medical situation warranted providing break-the-glass access [36]. Support vector machines were also applied to this same problem and data set, with only marginally improved results [33]. Both of these approaches determined if breaking the glass should be allowed by a given individual (a binary decision), but were not flexible enough to determine that, in the case that breaking-the-glass is warranted, who should then have access to the resulting information or be tasked with making the required

decisions. Thus the precision and accuracy were artificially high, but when deployed in emergency situations the result was that several people needed to try to break-the-glass before one that the system accepted was allowed [21]. (Fortunately these systems were tested in mock scenarios before deployment, uncovering these flaws before any systems were deployed and so not impacting on any emergency situations.) In 2028, a deep-learning algorithm that provided both functionality was developed [18]. A company formed around this technology, who tested it and deployed it with success at numerous health-related facilities, including hospitals, clinics, pharmacies, and the CDC [8]. It was this algorithm that was in place during the 2036 infection.

We rely on one other thrust of work in machine learning. Many of the machine learning algorithms have drawn conclusions in a manner that can be described but not explained. Some work had been done on interpretability. Palczewska and her colleagues examined developing patterns from the influence of various features of the training dataset [28]; the patterns allowed a model of the analysis performed by the machine learning system to be developed and assessed. Zhang and Zhu [44] used visualization as the basis for building models of convolutional neural networks analysis that could be explained. Pearl [24, 29] models causality using probability theory, graph theory, and structural equation models; this approach also leads to an understanding of how the inferences obtained from machine learning come about.

Argument-based machine learning means that, through argumentation, an expert can state her knowledge easily and unambiguously, leading to the construction of a knowledge base that machine learning algorithms use for texting and training [23]. As an example, Gómez and Chesñevar [10] point out that most machine learning algorithms are based on quantitative reasoning, and thus require training data to establish the desired functions, whereas argumentation is qualitative reasoning. They suggest that the knowledge base serving as background knowledge for argumentation can be built from training data. More to our point, they propose applying an argumentation theory in a machine-learning context. Our approach is similar.

3.1 Post-Mortem

An analysis of the failures of this algorithm during a post-mortem of the response to the 2036 infection found that the algorithm had actually learned to find the subject that was closest to the location where decisions needed to be made [22]. Given that the training data consisted of only cases where the closest medical personnel was able to address the emergency, this was not a surprising result. However, when deployed in real access control systems, particularly the CDC's, many more personnel were included, not just medical personnel, because at that point it was recognized that a variety of specialities were required to solve many medical problems. It was this gap between training data and real data that resulted in the failure of the AI-based access control system.

The post-mortem considered first what failed in the process. The approach chosen was to model the process of preparing the medicine for distribution and distributing it. From that model, the analysts generate a fault tree that showed what sequences of failures would cause the process to fail. By identifying single points of failure, the analysts can design additional steps to compensate for the failure, and prevent it from recurring given the same circumstances. The advantage of this approach is that the reasons for the failures are irrelevant to locating the weak points of the process. If a step fails, why does not affect the effect the failure has on the result of the process.

The disadvantage of this type of analysis is that the model must reflect the process accurately and completely. In practice, one focuses on specific parts of the process in order to do a thorough analysis of the steps of interest. One refines the steps into substeps, and those further, until the model reflects how the process actually works. In this case, the fault was known, and so the analysis could focus on modeling the specific part of the process — the distribution — that broke down.

The second disadvantage is that, although the reasons for the failures of steps leading to a failure of the process are not relevant to the process' failure, they are very relevant to *why* the process failed. There is a difference between a transformer going off-line because a breaker was thrown, and a hurricane destroying the power distribution system using that transformer. In both cases, the failure is the same: people lose power. But recovery for these two events is very different. In one case, a transformer is reset or replaced, which (presumably) takes place relatively quickly. But restoring the power distribution system will take at best days, and possibly weeks or months. Thus, the cause affects recovery — and in the case of our scenario, it is critical for the failed process of distributing the medicine to recover, that is to begin distribution, as quickly as possible.

Therein lies the problem. Recovery mechanisms must take into account the reasons for failure, and so the causes of the failures must be anticipated to some degree. The countermeasures must also be known and be feasible. But in all previous cases, the distribution involved medical personnel who worked with the medication or who had supervised medicine distribution. The machine learning algorithm's training data set did not include the weather, and all the primary distribution personnel getting ill, as none of the developers thought about it; and as previous situations were never affected by the weather, the machine learning algorithm never learned to include it. In some sense, this is a form of data poisoning, but the problem is not deliberate corruption of the training set; the problem is one of omission.

Thus, the machine learning system failed because of incomplete training data, and because of a novel situation.

4 APPROACH

Unique situations arise repeatedly in life. In most cases, one can solve problems created by these situations by generalizing lessons learned from similar yet different situations. A child's hand is burned when she touches a hot stove; she refuses to put her hand over a candle because she feels the heat of the candle, and it reminds her of the heat of the stove. But sometimes generalizations are not obvious. Dry ice is cold, so the lesson learned does not apply, and if a child has never encountered that level of coldness, they might believe that something cold cannot hurt them. Dry ice is fundamentally different than a stove or candle, because it involves cold, not heat. And yet when the child touches it, she will get a burn.

The problem is when unique situations arise in critical operations such as matters of life and death, and previous experience fails to generalize to include the situation. If they are analogous to other such situations, they can be handled in a similar way. But in other cases, the similarity is deceiving, and following the same path as the earlier incident will cause failure, as happened here. Thus, we must find an approach that takes into account the factors of the incident and yet not rely too heavily on the past.

In order to do this, we propose a system that has three key differences from existing systems:

(1) Given that humans are still the gold standard when it comes to reasoning and problem solving, especially in new and previously unseen circumstances, we propose a system that more closely emulates human thought processes in terms of logic (argumentation) rather than at the biological level (neural networks).
(2) Humans are still superior at creativity and "thinking outside the box" and so we leverage these capabilities through the use of red teaming the break-the-box system in order to better ensure that all possible contexts have been considered. We note that this is explicitly different from the majority of deep learning aproaches which focus nearly exclusively on historical data. (There are exceptions to this, such as in [12] and [17], however these approaches are largely limited to academic work and have not seen deployment in products.)
(3) We propose the usage of a combination of technologies, rather than focusing exclusively on a single technology (e.g., deep learning). Specifically, we make the case for a system based on argumentation but still leveraging neural network techniques to help develop and improve the system.

4.1 Argumentation

Using predicate calculus or other logics would be ideal, because we can reason from a set of hypotheses to reach a conclusion, and rigorously validate that conclusion. The problem is that interests compete, and so the weighting of the facts is uncertain; we do not know which should dominate. So we need a method that allows for incomplete and imprecise knowledge, and further allows us to reconstruct the chain of reasoning that led to the conclusion, here determining whom to notify.

Most forms of machine learning do not allow this reconstruction. Some forms are straightforward enough to allow a reviewer to determine why a particular classification was reached; for example, k-nearest-neighbor is intuitively clear. But it is also less accurate than other methods, and more complex mechanisms such as neural nets do not provide the reasoning that leads them to draw particular conclusions. As noted in Section 3, while work has progressed in network interpretability for approximately two decades, it is still not at the stage of providing human-understandable reasoning of decisions, particularly for complex environments that do not involve image processing. They are therefore in a sense "black boxes" validated by having them operate on test data with known conclusions, and comparing the results from the machine learning algorithms with those known conclusions. Thus, a different approach is needed.

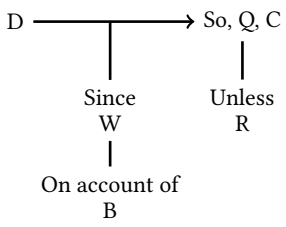

Figure 1: Structure of an argument

The approach we propose is argumentation, supplemented by red teaming by both human teams and generative adversarial networks (discussed in section 6).

4.2 Background on Argumentation

We use as our basis Toumlin's study of argumentation [38]. He defines an argument as a possibly qualified claim that is proposed, and the supporting data and reasoning. More precisely an *argument* puts forward a conclusion, or claim, supported by facts and reasoning based on those facts. His view is that the reasoning to arrive at a conclusion, and the facts upon which that reasoning relies, is important, as well as the conclusion. Further, conclusions can be invalidated by arguments using different facts and reasoning. Toulmin [38] We use the following terms.

- Data D are the facts upon which the argument is based.
- The warrant W is the reasoning that uses the data to support the claim.
- Backing B is information that supports the reasoning itself (as opposed to the data to which the reasoning is applied).
- A claim C is the proposition being put forth.
- A qualifier Q is a limitation or statement of strength for the claim.
- A rebuttal R is a statement of circumstances upon which the reasoning of the warrant would be incorrect or irrelevant.

Figure 1 illustrated the relationship of these terms.

As an example, consider the question of whether Matt is a registered student at UC Davis. The facts (D) are that he attends classes on campus, turns in homework, and takes the exams. Since (W) someone who does that is generally a registered student, on account of (B) people taking a class wanting to receive credit for work done, so, (Q) presumably (C) Matt is a registered student, unless (R) he is an auditor. Figure 2 shows this in the framework of an argument.

Dung [6] expanded on this by developing a theory around the acceptability of arguments. He defines an *argumentation framework* as a set of arguments and a binary relation on that set. The binary relation, "attacks", means that the first argument is a challenge to, or contradiction of, the second. For example, a parent asks a child if she ate the cookie. The child replies that she did not. The parent then points out there are cookie crumbs around the child's mouth. Treating these as arguments, the second attacks the first (because it contradicts the first) and the third attacks the second. In this framework, an argument ends when no arguments attack one of the claims, or when all but one of the arguers surrenders.

We will apply these ideas to the scenario to see how such a catastrophe might be avoided.

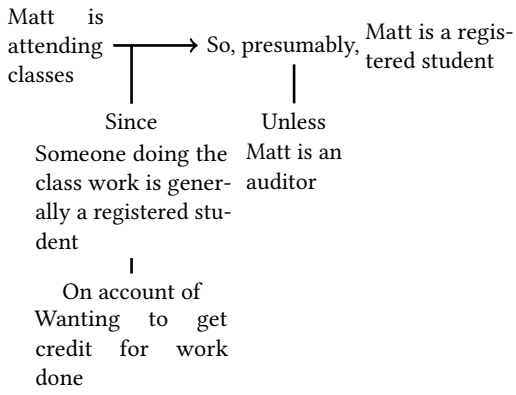

Figure 2: Example of an argument

5 APPLICATION

In our case, the goal is to determine who to contact to supervise the distribution of the medicine.

We first gather facts. The relevant facts for the distribution of medicine are the nature of the medicine, especially its volatility, potency, and handling requirements; the manner of distribution; and the factors that affect how it is to be distributed. Relevant to these are how the disease spread, and who can begin the distribution.

Our goal is to minimize the spread of the disease. To this end, we give weight to arguments, in the following way (least weight to most); the justification for this ordering is the goal of minimizing the spread of the disease.

(1) Arguments not involving people knowledgeable about the spread of the disease or how to run a large-scale distribution of medicine;
(2) Arguments supporting people knowledgeable about the spread of the disease;
(3) Arguments supporting people knowledgeable about how to run a large-scale distribution of medicine; and
(4) Arguments supporting people knowledgeable about both the spread of the disease and how to run a large-scale distribution of medicine.

Our argumentation proceeds as follows.

D1. The default vaccine distribution process failed.
W1. A person needs to intervene.
C1. Someone must be notified to intervene (the "break the glass" component)
D2. There is an ordered list of people, and their expertise, to notify.
W2. Someone is needed to determine how the medicine is to be distributed.
C2. Select a person based upon their expertise and availability.
D3. The first person on the list is an expert in infectious diseases.
W3. The disease is an infectious one.
C3. Notify the first person on the list.

So far, the argument has been straightforward. But now, other factors come into play.

D4. The spread of the disease is controlled by environmental factors (specifically, the weather).
W4. The first person on the list is not an expert on how environmental factors affect the spread of the disease.
B4. Being an expert in how infectious diseases affect people does not necessarily mean an understanding of how the disease spreads.
R4. The first person on the list has the most expertise on how the disease spreads.
C4. Someone other than the first person on the list should be notified.

Claim C4 attacks claim C3. Given the goal is to minimize the spread of the disease, and the weighting of the arguments given above, C4 has greater weight than C3, so C4 is more believable than C3. Note that R4, the rebuttal to W4, is read as if prefixed by "Unless", so it is stating a negative. We now examine who should be called, and incorporate another factor.

D5. The vaccine must be handled carefully to retain its full potency.
W5. The medical researcher who created the vaccine knows how delicate it is, and how to handle it.
C5. The medical researcher should be notified.

But another consideration arises: the environmental impact upon the spread of the disease.

D6. Whoever is notified must understand weather patterns.
W6. The weather affects how the disease is spread.
B6. The virus is transmitted through the air.
C6. The epidemiologist and meteorologist understand how the disease spreads.

Of the two people identified in C6, which should be notified? Our goal comes into play.

D7. Whoever is notified must be able to forecast how the wind blows.
W7. Future weather patterns control how the virus will spread in the future.
B7. The virus is spread through the air.
Q7. Usually
C7. Meteorologists can predict the weather accurately.
D8. The meteorologist can predict the weather accurately (claim C7)
W8. The goal is to stop (ideally) or limit (realistically) the spread of the disease.
C8. The meteorologist should be notified.

We now have two claims that attack one another, C5 and C8. To determine which is more credible, we must weigh the risk to the distributors against the risk that the disease will spread more rapidly than otherwise. With only these claims, the weighting C8 — which deals with how the disease will spread – has greater weight than C5. But another factor comes into play: the danger of mishandling the medicine. The key is to ensure the medicine is handled properly.

D9. Being exposed to too much of the vaccine will make a person ill.
W9. The distributors must be well to distribute the medicine.
R9. The medicine is handled improperly.
Q9. Usually

C9. The distributors will not become ill.

Again, R9 should be read as if prefixed by "Unless".

So argumentation says that C9 attacks C5, but no argument attacks C8. Hence C8 is the convincing argument. Hence the meteorologist who works with the CDC should be notified.

Ignoring the weights given to the arguments by the ordering described above, this particular argumentation sequence has an interesting feature: a loop. The argument culminating in C5 attacks the argument culminating in C8, because it asserts something contrary to C8. But argument culminating in C8 does the same against C5. In our case, we had an additional argument, the one that culminated in C9, that also attacked C5 but not C8. Therefore, as C5 was attacked by two different arguments, and C8 by one, the argument culminating in C8 is the more credible argument.

5.1 Perturbations

In the above arguments, the goal of distributing the medicine is to get it to the people who are currently uninfected. This can be seen from the use of weather prediction: where will the airborne virus go next? Consider though a different goal: the goal is to get the vaccine to people in areas where the disease is prevalent, to treat those who are ill first. In this case, the future spread of the disease becomes less important than the areas where the disease is currently prevalent. So, now, we have other arguments to consider.

D10. Whoever is notified must understand where the virus has spread.
W10. The epidemiologist has been studying the spread of the disease.
B10. The data provided by CDC field workers shows spread of the disease.
C10. The epidemiologist should be notified.

Now this argument attacks both arguments culminating in C5 and C8. Further, those two also attack this one. Again, the argument culminating in C9 attacks that culminating in C5, but not C8. So we examine the arguments for C8 and C10. We note that C8 is based on stopping the spread of the disease, which has greater weight than C5.

D11. Distribution requires knowledge of where the disease has spread.
W11. The goal is to get the vaccine to people who are already infected.
C11. Someone who knows where the disease has spread should be notified.

And now this argument attacks C8 but not C10. So we accept the argument culminating in C10.

Now, suppose environmental factors affect the effectiveness of the vaccine. In that case, the distribution is controlled by deciding which is more critical, the spread (and accept in some places the vaccine will be less effective) or the effectiveness (in which case some folks won't get the vaccine and so will become ill). In the former case, the epidemiologist would be identified as the person to notify first (see the argument ending in C10).[1] In the latter case, a new argument is introduced:

D10. The person notified must understand how the medicine interacts with the environment to which it is distributed.
W10. The vaccine loses its effectiveness if environmental factors are not right.
C10. Someone who developed the vaccine and best understands its interactions should be notified.

This attacks the arguments ending in C8 and C10, but not C5, and is not itself attacked. Hence we accept it, and consequently the argument culminating in C5.

5.2 Summary

Applying this to the infrastructure that the CDC and its machine learning system have developed, one would need to gather facts — information about how a disease might spread, and the expertise of people on the "break-the-glass" call list. Once the machine learning algorithm determines that someone on the list must be notified, the argumentation system takes over. The information about the crisis is weighted either by the machine learning system or the argumentation system based upon the stated and predetermined goals, and based on those weights and argumentation determines whom to call. In essence, the machine learning algorithm determines that something is unknown and so "breaks the glass"; argumentation is then used to arbitrate between the selection of specific actions because the data is incomplete.

6 UNDERLYING PROBLEMS

This approach has four main problems. None are insoluble, but all must be handled in some fashion.

6.1 Training

First, the factors relevant to the goals must be determined. This requires an analysis of the goals, and from that the specific constraints must be derived. These constraints will dictate what environmental and other factors affect the process meeting the stated goal. From these factors, one can deduce the requirements, which in turn identify the nature of the facts needed for argumentation to succeed. Much of this process can probably be automated.

However, while much of the process itself can likely be automated, determining the factors relevant to the goals requires deeper domain knowledge. Using the CDC case as an example here, if an argumentation system had been developed that had not considered weather as a possible vector for disease spread, then the correct arugment would never have been generated. The difficulty with this process lies in ensuring that the development team has considered not only all of the usual cases, but also all of the possibly edge cases as well, regardless of how improbable those cases may be. The end goal is to have a complete model of the knowledge required to make appropriate decisions for the deployment domain. As with machine learning models, it is possible to have an initial base model that can be provided (a *teacher* model for transfer learning, see eg [32] for examples in the neural network domain), but that model will still need to be configured for its specific environment. We expect that to address this, developers will follow a spiral model of some form that consists of development and testing (see subsection below), which will further inform development, repeated until a stable system has been produced.

[1] If stopping the spread were more critical, then the meteorologist would be notified, as argued by C8).

6.2 What I Say Versus What I Do

Second, there is a gap between what people do and what they are aware of doing. In this context, the discrepancies are important to understand because they may, or may not, affect the ability of the organization to follow the process identified as necessary to meet the stated goals. This requires interacting with the staff and other domain experts to identify factors and procedures, and then iterate until the models of the process and the identification of the facts is as complete as possible [5, 45]. This technique has been used successfully to improve software development, medical, and election processes [2, 27, 43].

6.3 Testing

How to properly test such a system needs to be determined. One way is to have the equivalent of a red team exercise in which an outside team of experts would develop a series of scenarios that can be used to test the break the glass system. This red team would need to understand the context and environment in which the system was deployed, as well as the organizational culture. Further, they would need the education and expertise to be able to conceive of edge cases based on what they had studied and experienced. And even with such a testing process in place, the tests are limited by the imagination of the testers. It is possible that the designers and testers will overlook causes of failures that have potentially dire consequences; however, they are more likely to cover possibilities that a system should consider than a machine learning approach based on limited historical and training data.

The structure of the test is that of a table-top exercise. The goal of the test is to identify gaps in the training sets, and any assumptions that the argumentation engine begins with — such as the weighting used to choose between conflicting arguments. The adversaries here are people with knowledge of the processes (as described in Section 6.2). Their goal is to generate scenarios that the machine learning and argumentation system can try to handle — and then they can determine if the matter were handled correctly. The system we propose would be tailored for interpretability, so the adversaries could see *why* the system made the choices it did. The state of the art for interpretability has advanced sufficiently to enable this.

Counterfactuals are a useful tool here. A *counterfactual* is a claim about matters that are not believed to exist when the claim is made [4]. It has been used to examine issues in disparities in test scores among various populations [16], diversity initiatives in business organizations [42], and consent in experiments that involve deception [41]. In terms of computer security, the concept of "think like an attacker" is the ability to create counterfactuals that describe what the attacker can do. Risk analysis also is counterfactual analysis, because the analyst examines possible future events, many of which will prevent others from occurring. Herley and Pieters point out their necessity in computer security [14], and the table-top exercise is no exception.

During the table-top exercise, the participants will be asked to imagine catastrophic situations as well as problems arising from failures or misunderstandings. These may be based on events in the past, but with suitable modifications to exacerbate the situation. The advantage to this approach is the results of the exercise can then be compared to the results of the real situation, to see what changes would have more closely met the needs of the counterfactual situation. They may also draw on their imagination to create scenarios, and evaluate the results with respect to some metric (for example, the number of people who get ill, the difficulty of procuring the medicine, and perhaps the effect of political or bureaucratic constraints).

A variant of this is to test the break the glass system symbolically. Here, testers create scenarios that are fed directly to the argumentation system, just as was done with question-answer systems and expert systems [13]. These scenarios have outcomes that the testers believe are appropriate. The argumentation system's responses would then be tested against the known "good" results. This is similar to the red team exercise, except that exercise would test the machine learning component as well — does it "break the glalss" at appropriate times? Here, the testers assume the glass has been broken already, that they have all known scenarios, and are testing only against these known scenarios.

A third approach to testing leverages generative adversarial networks (GANs) [11]. GANs consist of two models, a generator and a discriminator. Traditionally, the disriminator learns to distinguish between real and generated (or fake or malicious) data. Meanwhile, the generator attempts to create fake data that has characteristics such that the discriminator classifies the data as real. First developed for images, GANs were subsequently modified for use with text (e.g., [35]). While this approach saw widespread use initially with trying to detect malicious training data and, later, with countering the spread of fake news [34], its popularity waned and its usage remained confined to these niche areas.

Here, we propose using this same framework to test the argumentation system. The goal of the discriminator is broadened beyond a binary response to one of determining the appropriate authority in a break-the-glass situation. More specifically, the discriminator is the augmented argumentation system we propose. The generator, in contrast, takes known scenarios and attempts to modify them so that the discriminator provides a different authority as a response. When the generator has achieved this goal, a person is charged with looking at the results to determine if the new scenario created by the generator is realistic and, if so, if the argumentation system responded appropriately or needs to be modified.

6.4 Explanations and Reasoning

Given this is an unusual ("break-the-glass") situation, understanding why the system determines whom to contact, and how that determination was made, is important both to understand the effects of the algorithms used, and to enable an evaluation of how well the system works. If a human sees the result before it is acted upon, an explanation also serves as a check to ensure the system does not make unreasonable decisions.

Pieters [30] describes three different types of explanations:

- *Traces* present a detailed record of reasoning steps.
- *Justifications* present a logical argument for the action.
- *Strategies* are high-level approaches for solving a class of problems.

For our purposes, the first two of these are of interest. When reporting a result or taking some action, both the underlying machine learning system (or other system) and the break-the-glass

system need to provide justification in a form a human observer can understand. They also should log the traces of events and intermediate decisions leading up to the final result or action. The latter may be too much to review before taking action. However, at any *post mortem*, the auditors can use the traces to determine what information, and what steps in the chain of intermediate decisions, led to the result or action.

For the argumentation system, the chain of decisions will be straightforward; nevertheless, the facts leading to each decision in the chain may not be obvious and so must be recorded in the trace. For the underlying machine learning system, obtaining a trace is straightforward; merely record all the values at each part of the internal evaluation engine (for example, at each node of a neural net). The problem is using these to develop a justification for a decision. The trace indicates what happened, but does not explain the effects of the inputs and outputs, and that leads to a lack of justification — which is critical to understanding why the break-the-glass occurred.

6.5 Attacks on the System

An additional complication occurs if the databases containing the information about the spread of the disease are corrupted maliciously. In that case, the assigned weights will probably be incorrect, resulting in erroneous actions by the machine learning system (here, failing to recognize this is a "break-the-glass" situation) or the argumentation system (here, failing to recognize the criticality of future weather patterns). The argumentation system can take this possible attack into account by obtaining evidence about whether such an attack has occurred, and using argumentation to determine how to handle such an attack (assuming it occurred).

6.6 Conflicts in the System

Being largely a collection of rules, an argumentation system can be incorrect. As alluded to in Section 5, in an application, an argument can get into situations that are analogous to deadlock, where two claims attack one other, or into a loop where the arguments culminating in a claim attack the arguments culminating in another claim and vice versa. In essence, the rules can lead to situations that violate a liveness property. Besides deadlock and loops, liveness can be violated if situations arise in which no warrant applies; for example, if the arguments are incomplete.

Also, an automated argumentation system can provide claims that are not what the human designer of the system intended, analogous to a violation of a safety property.

In the verification literature, safety corresponds to "nothing bad happens" and liveness to "something good happens".

Liveness can be checked by dynamic analysis as the argumentation proceeds either by human intervention or automated analysis, similar to on-line deadlock detection in multi-processing. Safety can be checked by human intervention that notes unexpected claims or through assertions on expected claims that are continuously checked.

Static analysis can, in principle, also be used since the rules for an argumentation process are in essence a program and the analysis corresponds to program verification. O'Keefe and O'Leary [25] survey verification methods for expert system rules, a good starting point for the static analysis of argumentation systems. Analogous to program verification, static analysis for an argumentation system can in principle guarantee that liveness or safety are assured for all input.

As discussed above, errors in the rules that constitute an argumentation system can lead to unexpected claims or to the argumentation process failing to terminate.

Besides the "logic" of the rules, the weights assigned to arguments could be in error, given that they are human-set. Although, it is difficult to formally verify the weights since they are largely subjective they can be to a certain extent analyzed. A higher weight argument would have clauses that cover those of a lower weight argument. And, as indicated in Section 5, the justifications for a higher weight would, in some sense be more important than those for a lower weight argument.

7 COMPARISON TO RELATED APPROACHES

While much has been written in the mainstream media regarding the failures in the systems that led to the large number of deaths during this infection, little has been done in the academic arena. Within the media, the main question has centered around why there wasn't more human intervention. Thus we consider the first alternative approach to be less automation.

A more desirable approach is to have a human monitor the distribution process and intervene when a "break-the-glass" situation occurs. In fact, *two* people were supposed to be monitoring the process, and when they agreed intervention was necessary, they were to intervene. The problem that the review found was the lack of co-ordination between the two monitors. Each thought the other would indicate when they saw a problem, and so waited to react. The review group recommended reorganizing the monitoring so that one monitor could "break the glass", but also keeping the automated system to work with the monitors and to "break the glass" if for some reason the human did not.

We note that the review recommended keeping the automated system — this is important in case one or both of the monitors become ill or are otherwise unavailable (e.g., at the hospital with a loved one). Thus, we still feel that the system proposed in this paper is important and needs to be developed to provide such a back-up system. (We also hope that this approach can be taken in other instances where there is an over-reliance on machine learning based on historical data that may not cover all eventualities, such as in many government systems, SCADA systems and financial systems.)

To date, no alternative approaches to using deep learning for break-the-glass scenarios have been proposed in the academic literature. We thus hypothesize an alternative based on our background literature survey. Specifically, it seems reasonable to expect that an approach that combines human reasoning with neural networks should be possible. As identified in Section 4, there is previous work that combines human learning with deep learning to create hybrid systems that perform better. These combinations typically take one of two forms: (1) leveraging human intuition to create better training sets (which we use here in some sense in terms of the red team models) and (2) combining expert systems-type approaches with deep learning and neural networks. In either of these cases,

the aim is to improve on the limited black-box nature of neural networks by leveraging human intuition. While these approaches are bound to be better than the current historical-only learning approach — especially the first combination — they do not have the added advantage of allowing for human understanding of the logic and argument process. The ability to easily represent the logic of a particular decision allows the system designer to determine if there is missing information or flaws in the logic, thus allowing more effective debugging and a system that more closely adheres to both policy and expectations.

Other possible approaches include alternative forms of machine learning, such as decision trees and logops (logical operations, first published by Holmes and O'Kelly-Davis [15]); however these approaches also suffer from the heavy use of historical data. Logops is perhaps the closest machine learning approach to argumentation, which consists of a series of statistical analyses that determine the appropriate next steps but in a manner that allows for the skipping of intermediate steps (see Olivaw[26] for a detailed description). The next steps can be thought of as similar to arguments, but with a statistical analysis being used to determine the arguments to be given the greatest weight. It is the statistical analysis that is the greatest weakness in these scenarios as it is historically based only, and determining appropriate statistical weightings to arguments is difficult to do without real data.

8 GENERALIZABILITY

As we noted in the introduction, the use of argumentation instead of machine learning extends beyond break-the-glass scenarios. In particular, this approach is particularly useful for scenarios where a decision based on logical deduction is required. That is, it is not useful in all instances where machine learning is employed. For example, machine learning works well for image processing tasks such as facial recognition, where an argumentation system would be cumbersome at best (e.g., the arguments would end up being descriptions of facial qualities that would still require image processing to detect). But there are other areas where machine learning is perhaps being used inappropriately, such as in situations where a logical decision needs to be made and where it is easily conceivable that historical events do not capture all possibilities.

Argumentation is particularly well-suited for situations that require auditability, such as when there is an intersection between technology and society. Extending the infection scenario further, a situation can be envisioned where the correct public policy might involve sacrificing some number of people, rather than attempting to save everyone. Argumentation is useful here for two reasons:

(1) The logic that the system has used to determine the result (e.g., who should be contacted in a break-the-glass situation) can be presented in an easily understood way to the system operator. Thus the logic can also be reviewed by a person to ensure that no information has been missed, and that the end result makes sense. This further ensures some level of accountability, as policies can be structured such that a person needs to approve the result from the algorithm. This is particularly useful in situations where the "right" answer is not necessarily the ethical answer, and so having this oversight is crucial.
(2) The arguments in this system can be structured to take into consideration policies (such as public policy) rather than be based solely on historical information where policies would only be learned implicitly. This does, however, come with a couple of caveats. First, an appropriate weighting needs to be assigned to any policies such that a system can also represent when it might be appropriate to break the policy, or when the policy should over-rule other possible results. Secondly, this underscores the need for having an appropriate development team and an appropriate red team, such that the contributors in each of these are not solely technology people but rather encompass those who set policy as well.

There are two additional requirements on argumentation that, as future work, will help the system generalize further. The first is that some approach is needed to update rules in the system as missing rules are discovered and to update existing rules with new information. For example, public policy often changes with changes in government, and so an approach that allows for updating these policies (hopefully without simultaneously requiring extensive red team testing) is required.

The second area of future work is on system self-awareness. While it is noted above that accountability can be built into the system by requiring a human operator to concur with the system's recommendation, ideally the system would be able to determine when it might need additional information. Confidence scores might be useful here, with a low confidence answer from the system resulting in the equivalent of the system knowing that it needs to ask for help rather than provide an automated response. Given that an automated system can not understand the ethics of a given situation, integrating the oversight of an operator to validate the system is necessary.

9 CONCLUSION

In this paper we describe an approach — argumentation — that can be used to replace certain "black box" machine learning algorithms, focusing on an example scenario for breaking the glass in access control systems. Argumentation is a modeling approach that uses logic to determine the best outcome by creating arguments, developing conclusions, and then following up with counter arguments and thus counter conclusions. The context is applied to determine which argument has the most weight, and thus which conclusion should be chosen.

Most effective machine learning approaches are essentially "black box" in the sense that why they arrived at their outputs is unknown. The methods used by machine learning are of course known — but why was one weight assigned in the interior of a neural net rather than another weight? This intuition causes people to not understand why the machine learning systems produce the outputs — they accept that the outputs are right, or rather that errors are negligible.

Unfortunately, unexpected events sometimes occur, and then the training and history fail. That is what happened here. Indeed, the best red team testers look at how the security mechanisms and procedures work, tease out the underlying assumptions, and then take actions that violate them. They bank on the defenders believing their assumptions will always hold, or — even better —

being ignorant of their assumptions. Mimicry attacks [40] work this way; they analyze the algorithms, sometimes indirectly, and then introduce perturbations or changes that cause the algorithms to fail because the attackers mimic acceptable sequences of actions, but in such a way that they compromise the system. Social and political structures are also vulnerable to this type of attack, as Saul Alinsky [1] and Sun Tzu [39] have demonstrated.

Indeed, this was one of the causes for the multitude of computer security disasters seen over the past ten years. In the first part of the 2020 decade, security firms and agencies believed the attacks were becoming so sophisticated and changing so rapidly that only machine learning could provide effective defenses, and so they integrated those systems into the security software and systems they sold and used. Adversaries exploited this by finding attacks that were not present in the training and testing data sets. The resulting compromises of systems were not in and of themselves disastrous; what was disastrous was the unwillingness of people to believe they had been compromised, because the machine learning system reported no successful attacks. Instead, the data and problems that resulted were ascribed to non-cybersecurity problems. The point is that black box approaches, such as deep learning, learn solely from previous experience and are thus unable to "think outside the box", or take into consideration new information that has not been part of any previously learned scenario (its training set). The result of this failure can be catastrophic, particularly in life or death situations.

The application of argumentation in future break-the-glass systems provides an assist to situations which the machine learning systems cannot handle. They also force the site to examine their systems, procedures, and environments to provide weighting factors and ancillary data that the argumentation system can use to arbitrate between arguments. In this way, argumentation systems can help prevent disastrous scenarios such as what happened during the 2036 epidemic.

ACKNOWLEDGEMENTS

This material is based upon work supported by the National Science Foundation under Grant Number OAC-1739025 to the University of California at Davis. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. We thank the National Science Foundation for this support.

We thank Dr. Peter Yellowlees of the University of California Davis Medical School and Dr. Michael Bishop of the Pioneers Memorial Healthcare District Hospital for reviewing our 2036 medical scenario and providing valuable feedback based on their experience in the medical profession. We also thank the anonymous reviewers who provided insightful comments, Ben Edwards for his helpful feedback during the pre-workshop shepherding process, Simon Foley for his helpful feedback during the post-workshop shepherding process, and the workshop attendees for their valuable suggestions. All these have resulted in a far better paper than the original.

REFERENCES

[1] Saul D. Alinsky. 1989. *Rules for Radicals.* Vintage Books, New York, NY, USA.

[2] George S. Avrunin, Lori A. Clarke, Leon J. Osterweil, Stefan C. Christov, Bin Chen, Elizabeth A. Henneman, Philip L. Henneman, Lucinda Cassells, and Wilson Mertens. 2010. Experience Modeling and Analyzing Medical Processes: UMass/Baystate Medical Safety Project Overview. In *Proceedings of the First ACM International Health Informatics Symposium*. ACM, New York, NY, USA, 316–325.

[3] Matt Bishop, Carrie Gates, and Karl Levitt. 2018. Augmenting Machine Learning with Argumentation. In *Proceedings of the 2018 New Security Paradigms Workshop*. ACM, New York, NY, USA, 50–55.

[4] Kenneth T. Broda-Bahm. 1995. Counterfactual Possibilities: Constructing Counter-to-Fact Causal Claims. *Contemporary Argumentation and Debate* 16 (1995), 73–85.

[5] Stefan Christov, Bin Chen, George S. Avrunin, Lori A. Clarke, Leon J. Osterweil, David Brown, Lucinda Cassells, and Wilson Mertens. 2007. Rigorously Defining and Analyzing Medical Processes: An Experience Report. In *Proceedings of the 2007 International Conference on Model Driven Engineering Languages and Systems (Lecture Notes in Computer Science)*, Vol. 5002. Springer, Berlin, 118–131.

[6] Phan Minh Dung. 1995. On the Acceptability of Arguments and Its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and n-Person Games. *Artificial Intelligence* 77, 2 (Sept. 1995), 231–357.

[7] Anna Ferreira, Ricardo Cruz-Correia, Luis Antunes, Pedro Farinha, E. Oliveira-Palhares, David W Chadwick, and Altamiro Costa-Pereira. 2006. How to Break Access Control in a Controlled Manner. In *Proceedings of the 19th IEEE International Symposium on Computer-Based Medical Systems*. IEEE, Los Alamitos, CA, USA, 847–854.

[8] Amy Fowler and Will Kane. 2031. Dark Horse Healthcare Promotes Deep Learning Across Healthcare Industry. *The New York Times* (Sep. 30, 2031), B1, B4. [*future publication*]

[9] Emmanouil Georgakakis, Stefanos A Nikolidakis, Dimitrios D Vergados, and Christos Douligeris. 2011. Spatio Temporal Emergency Role Based Access Control (STEM-RBAC): A Time and Location Aware Role Based Access Control Model with a Break the Glass Mechanism. In *Proceedings of the 2011 IEEE Symposium on Computers and Communications*. IEEE, Los Alamitos, CA, USA, 764–770.

[10] Sergio Alejandro Gómez and Carlos Iván Chesñevar. 2004. *Integrating Defeasible Argumentation and Machine Learning Techniques*. Technical Report arXiv:cs/0402057v2 [cs.AI]. CoRR.

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems*, Vol. 27. Curran Associates, Inc., Red Hook, NY, USA, 2672–2680.

[12] Thomas Gradgrind. 2024. Supplementing Limited Training Data with Human Intuition in Deep Learning. In *Proceedings of the 20th International Conference on Machine Learning and Data Mining*. IEEE Computer Society, Los Alamitos, CA, USA, 129–147. [*future publication*]

[13] C. Cordell Green. 1969. Theorem Proving by Resolution as a Basis for Question-Answering Systems. In *Proceedings of the Fourth Annual Machine Intelligence Workshop*. Edinburgh University Press, Edinburgh, Scotland, 183–208.

[14] Cormac Herley and Wolter Pieters. 2015. "If You Were Attacked, You'd Be Sorry": Counterfactuals as Security Arguments. In *Proceedings of the 2015 New Security Paradigms Workshop*. ACM, New York, NY, USA, 112–123.

[15] Mycroft Holmes and Manuel Garcia O'Kelly-Davis. 2025. Logical Operations (LogOps) — A New Approach to Machine Learning. *Advances in Neural Information Processing Systems* 16, 5 (2025), 1271–1283. [*future publication*]

[16] Odis Johnson and Michael Wagner. 2017. Equalizers or Enablers of Inequality? A Counterfactual Analysis of Racial and Residential Test Score Gaps in Year-Round and Nine-Month Schools. *Annals of the American Academy of Political and Social Science* 674, 1 (2017), 240–261.

[17] Wyoming Knott and Adam Selene. 2026. Leveraging Lessons from Expert Systems to Improve Deep Learning. *Expert Systems with Applications* 78, 12 (2026), 29–39. [*future publication*]

[18] Ruth Leavitt, Jeremy Stone, and Mark Hall. 2028. Deep Learning for Health Record Access Control. *IEEE Journal of Biomedical and Health Informatics* 32, 1 (2028), 4–21. [*future publication*]

[19] Ming Li, Shucheng Yu, Kui Ren, and Wenjing Lou. 2010. Securing Personal Health Records in Cloud Computing: Patient-Centric and Fine-Grained Data Access Control in Multi-Owner Settings. In *Proceedings of the 2010 International Conference on Security and Privacy in Communication Systems (Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering)*, Vol. 50. Springer, Berlin, Heidelberg, Germany, 89–106.

[20] Srdjan Marinovic, Naranker Dulay, and Morris Sloman. 2014. Rumpole: An Introspective Break-Glass Access Control Language. *ACM Transactions on Information and System Security* 17, 1 (2014), 2.

[21] Andrew Martin. 2028. Testing AI-Based Access Control for Electronic Health Records. *IEEE Proceedings on Software Development* 165, 1 (2028), 16–23. [*future publication*]

[22] Leonard McCoy and Beverly Crusher. 2037. *Failures in 2036 Infection Response*. NIH Publication 37-0071. National Institute of Health, Washington, DC. [*future publication*]

[23] Martin Možina, Jure Žabkar, and Ivan Bratko. 2007. Argument Based Machine Learning. *Artificial Intelligence* 171, 10–15 (July 2007), 922–937.

[24] Leland Gerson Neuberg. 2003. Review of 'Causality: Models, Reasoning, and Inference'. *Econometric Theory* 19, 4 (Aug. 2003), 675–685.

[25] Robert M. O'Keefe and Daniel E. O'Leary. 1993. Expert System Verification and Validation: A Survey and Tutorial. *Artificial Intelligence Review* 7, 1 (Feb. 1993), 3–42.

[26] R. Daneel Olivaw. 2034. A Survey of Coordinated Attacks and Collaborative Intrusion Detection. *Computers & Security* 53, 2 (2034), 117–138. [*future publication*]

[27] Leon J. Osterweil, Matt Bishop, Heather M. Conboy, Huong Phan, Borislava I. Simidchieva, George S. Avrunin, Lori A. Clarke, and Sean Peisert. 2017. Iterative Analysis to Improve Key Properties of Critical Human-Intensive Processes: An Election Security Example. *ACM Transactions on Privacy and Security* 20, 2 (March 2017), 5:1–5:31.

[28] Anna Palczewska, Jan Palczewski, Richard Mrachese Robinson, and Daniel Neagu. 2014. *Integration of Reusable Systems.* Advances in Intelligent Systems and Computing, Vol. 263. Springer, Cham, Switzerland, Chapter Interpreting Random Forest Classification Models Using a Feature Contribution Method, 193–218.

[29] Judea Pearl. 2009. *Causality: Models, Reasoning and Inference* (2nd ed.). Cambridge University Press, Cambridge, UK.

[30] Wolter Pieters. 2011. Explanation and Trust: What to Tell the User in Security and AI? *Ethics and Information Technology* 13, 1 (March 2011), 53–64.

[31] Dean Povey. 1999. Optimistic Security: A New Access Control Paradigm. In *Proceedings of the 1999 New Security Paradigms Workshop.* ACM, New York, NY, USA, 40–45.

[32] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. 2014. NN Features Off-the-Shelf: An Astounding Baseline for Recognition. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops.* IEEE Computer Society, Los Alamitos, CA, USA, 512–519.

[33] Jay Score. 2024. Applying SVMs to Predict Appropriate Access Control Policies. *IEEE Transactions on Dependable and Secure Computing* 21, 5 (2024), 533–545. [*future publication*]

[34] Hari Seldon and Gaal Dornick. 2023. Generative Adversarial Networks in Society: A Survey. *Proceedings of the 17th International AAAI Conference on Web and Social Media* 23 (2023), 505–514. [*future publication*]

[35] Rakshith Shetty, Bernt Schiele, and Mario Fritz. 2018. A4NT: Author Attribute Anonymity by Adversarial Training of Neural Machine Translation. In *Proceedings of the 27th USENIX Security Symposium.* USENIX Association, Berkeley, CA, USA, 1633–1650.

[36] Hermann Strangelove. 2023. Naive-Bayes to Break-the-Glass in RBAC-Administered Databases. In *Proceedings of the 39th Annual Computer Security Applications Conference.* IEEE Computer Society, Los Alamitos, CA, USA, 1–10. [*future publication*]

[37] William Sutton. 2022. A Review of Break-the-Glass Scenarios and Real-World Consequences. *Journal of American Medical Informatics Association* 29, 4 (2022), 2–17. [*future publication*]

[38] Stephen E. Toulmin. 2003. *The Uses of Argument.* Cambridge University Press, Cambridge, UK.

[39] Sun Tzu. 1983. *The Art of War.* Delacorte Press, New York, NY, USA.

[40] David Wagner and Paolo Soto. 2002. Mimicry Attacks on Host-based Intrusion Detection Systems. In *Proceedings of the Ninth ACM Conference on Computer and Communications Security.* ACM, New York, NY, USA, 255–264.

[41] Alan T. Wilson. 2015. Counterfactual Consent and the Use of Deception in Research. *Bioethics* 29, 7 (Sept. 2015), 470–477.

[42] Leon Windscheid, Lynn Bowes-Sperry, Jens Mazei, and Michèle Morner. 2017. The Paradox of Diversity Initiatives: When Organizational Needs Differ from Employee Preferences. *Journal of Business Ethics* 145, 1 (Sept. 2017), 33–48.

[43] Alexander Wise, Aaron G. Cass, Barbara Staudt Lerner, Eric K. McCall, Leon J. Osterweil, and Jr. Stanley M. Sutton. 2000. Using Little-JIL to Coordinate Agents in Software Engineering. In *Proceedings of the Fifteenth IEEE International Conference on Automated Software Engineering.* IEEE, Piscataway, NJ, USA, 155–163.

[44] Quan-shi Zhang and Song-chun Zhu. 2018. Visual Interpretability for Deep Learning: A Survey. *Frontiers of Information Technology & Electronic Engineering* 19, 1 (Jan. 2018), 27–39.

[45] Didar Zowghi and Chad Coulin. 2005. Requirements Elicitation: A Survey of Techniques, Approaches, and Tools. In *Engineering and Managing Software Requirements.* Springer, Berlin, Heidelberg, 19–46.