An Exploration of Decision Making Models in the Face of Untrusted Data

Sarah Kinsey

Abstract

In various domains ranging from security and conservation to public health and planning, an ever-increasing amount of artificial intelligence approaches are being deployed in the real world. With real world deployments come additional complexities, challenges, and vulnerabilities. This work examines these considerations from three broad directions: security games, data-based decision making, and adversarial learning. For security games, we're primarily concerned with scenarios involving deliberate deception, where one agent manipulates data to alter a strategy formed by its adversary. Similarly, we study data-based decision making, where data is used by a learning model to make some decision. This data provides a vector for attack, which could be taken advantage of by an adversary. To investigate the various threat models, we draw on adversarial learning research which studies how attacks can be carried out when such an opening exists, in addition to providing defences. Understanding these three areas will provide a comprehensive view of how decision making models can perform when the data upon which they rely is compromised, enabling further research to create more robust systems.

Introduction

Artificial intelligence has been applied to a variety of domains, including security, conservation, public health, and city planning. In all of these domains, additional challenges (such as imperfect data and deceptive behavior) arise when considering real-world deployments. This survey paper explores these concerns from three directions: security games, data-based decision making, and adversarial learning. While, at first glance, these areas seem rather disparate, they each provide valuable perspectives on the challenges of working in realistic (i.e. messy and insecure) settings.

As security games model interactions between adversaries, it is natural to consider deceptive behavior. Most well established is deception from the defender side (Rabinovich et al. 2015; Zhuang, Bier, and Alagoz 2010; Guo et al. 2017), for example, concealing the allocation of defensive resources. More recently, a lot of work has been done investigating deception from the attacker side (Nguyen et al. 2019; Nguyen and Sinha 2022; Gan et al. 2019a; Zhang, Wang, and Zhuang 2021), such as acting deceptively during a data collection phase so that the defender will create an exploitable strategy. Naturally, defenses against such deception (which must rely on the limited data available to the defender) have also been studied.

Our next area of interest, data-based decision making, does not fall under the umbrella of game theory but it follows a similar pattern to the security game settings we are interested in. Namely, performing some learning on data in order to then make a decision. In security games, this takes the form of one agent observing behavior to form a model of an adversary and create an effective strategy against it. Data-based decision making, however, covers a broader variety of approaches united by a common theme: their goal is to make a decision based on data that will be unobservable at test time.

Due to that restriction, a model must be built that can predict said unobservable data from correlated data that *is* directly observable. The traditional approach is simple: train a model to maximize prediction accuracy. However, in real world settings, it's inevitable that predictions will not be perfect. Thus, work (Wilder, Dilkina, and Tambe 2018; Donti, Amos, and Kolter 2017; Wilder et al. 2020) has investigated incorporating the end goal (high quality decisions) directly into the model's training process. As this field involves using AI to make decisions based on real-world data, it's natural to examine the data itself as a source of vulnerability (Kinsey et al. 2023) but adversarial research in this area is almost non-existent.

For inspiration in attacking both of these domains we draw on the area of adversarial learning, which studies attacks on machine learning models as well as defenses. Of particular interest to us is graph adversarial learning, as graph learning problems often fall into the data-based decision making paradigm (e.g. predicting links based on node information and then solving a problem such as bipartite matching on the predicted graph). However, deep learning in general and computer vision are more well researched from an adversarial perspective. Understanding existing adversarial learning research is key to applying these methods to data-based decision making applications, and being able to create robust approaches in this field.

This work will start by providing an overview of security games and discussing the current state of deception research in this field. Next, we will describe data-based decision making and detail existing applications, paying particular atten-

tion to the relatively new decision-focused approach as well as some social good applications that are adjacent to this area of research. Lastly, we investigate adversarial learning. Our work gives particular attention to poisoning attacks and graph based adversarial learning, as those are of particular relevance to both security game deception and attacks on data-based decision making.

Security Games

Our first area of interest lies in security games, which is primarily an area of study using game theory to optimize defensive resources in real-world security problems. We start by describing some real-world applications motivation further interest in this field. Then, we discuss two key models (Stackelberg Security Games and their variant Green Security Games) to provide some context. Next, we discuss some common models of human behavior in these games, which are important when considering how deception can function. Lastly, we discuss the current state of deception research in this domain.

Game theoretic AI approaches have been shown effective in a variety of real world applications, particularly in security and wildlife conservation. One security application, ARMOR (Pita et al. 2008), was deployed to protect the Los Angeles International Airport. This is accomplished via modeling the interaction between adversaries such as terrorists or drug smugglers and airport security as a Stackelberg game, considering terminals and checkpoints as targets top be attacked. Solving this game using their method, DOBSS, yielded strategies that outperformed the existing human devised schedules, while still allowing for manual overrides within the scheduling system.

PROTECT (Shieh et al. 2012) has been deployed by the US Coast Guard to optimize patrols and protect the ports of the United States. Their method models interactions between the Coast Guard and terrorists as a Stackelberg Security Game (Tambe 2011) with the Coast Guard as the defender and the terrorists as attackers. The targets considered are areas of interest in a port (e.g. critical infrastructure) which must be protected via Coast Guard patrols. To model the attacker's behavior, they utilize the Quantal Response model (McKelvey and Palfrey 1995; Yang et al. 2011) which allows for modeling of sub-optimal attacker behavior.

One challenge faced in this domain is the sheer number of targets, which results in an exponential number of potential patrols. To account for this, PROTECT first divides the port into patrol areas, and restricts patrols to covering targets within a single area. Further, they reduce the final number of strategies considered (each strategy corresponding to a patrol allocation) via removing equivalent strategies as well as dominated strategies from the list, resulting in a more compact representation of the strategy space. Patrols created by their system resulted in more consistent coverage of targets as well as more proportional coverage (i.e. more valuable targets are patrolled more frequently).

In the conservation domain, PAWS (Fang et al. 2016) has been deployed in parks in both Uganda and Malaysia to combat poaching. Its approach divides the parks into grids and then models the rangers vs poachers dynamic as a Green Security Game (Fang, Stone, and Tambe 2015), with rangers as the defender and poachers as the attackers. Each grid cell is considered a target, and the value of that target is determined by the animal density in that area. To model the attacker/poacher's behavior, the authors use Subjective Utility Quantal Response (Nguyen et al. 2013) with parameters learned from historical data. Solving the game yields patrol strategies that were proven effective.

Improving on this work, researchers (Xu et al. 2020) investigate an end-to-end approach for creating patrol strategies based on observed poacher data. To do so, they integrate Gaussian processes into an ensemble learner, quantifying the various levels of uncertainty in predictions across different sections of the park. Then, they use this uncertainty in order to build more robust patrol strategies. Experimentally, this method increased detection of poaching by 30%.

Stackelberg Security Games (SSGs)

SSGs (Tambe 2011) consist of at least two players: a defender (the leader in traditional Stackelberg games) and one or more attackers (the followers). The defender's goal is to protect a set of T targets from these attackers, given a limited number of *resources* (K, where $K < T$) that each can be allocated to protect a single target. A defender's pure strategy consists of a one-to-one allocation of resources to targets. A mixed defense strategy, $\mathbf{x}$, is a probability distribution over these pure strategies. This mixed strategy can be represented as a coverage probability vector: $\mathbf{x} = \{x_1, x_2, \ldots, x_T\}$, where $x_i \in [0, 1]$ represents the probability that target i is protected by the defender and $\sum_i x_i \leq K$. In SSGs, the attacker is fully aware of the defender's *mixed* strategy and chooses a target to attack based on this knowledge.

Suppose the attacker decides to attack target i. This action gives each player a reward or a penalty, depending on whether the defender is currently protecting target i. If i is unprotected, the attacker gains reward R_i^a and the defender receives penalty P_i^d. Conversely, if target i is protected, the attacker takes penalty $P_i^a < R_i^a$ and the defender gains reward $R_i^d > P_i^d$. Given coverage probability x_i, the *expected* utilities for the defender and the attacker resulting from an attack on target i can be formulated as follows:

$$U_i^d(x_i) = x_i R_i^d + (1 - x_i) P_i^d$$
$$U_i^a(x_i) = x_i P_i^a + (1 - x_i) R_i^a$$

Solution Concepts and Equilibrium The standard solution concept for Stackelberg games is the Strong Stackelberg Equilibium (SSE). Note that this may be different from the Nash equilibrium as Stackelberg games are non-simultaneous. To be considered a SSE, a pair of attacker/defender strategies must satisfy three conditions:

- The defender's strategy, x, is the best response to the attacker's strategy, g:

$$U_d(x, g(x)) \geq U_d(x', g(x')) \forall x'$$

- The attacker's strategy, g, is the best response to the defender's strategy, x:

$$U_a(x, g(x)) \geq U_a(x, g'(x)) \forall g'(x)$$

- If any ties exist (strategies with equal expected utility for the attacker), the attacker breaks ties in favor of the defender:

$$U_d(x, g(x)) \geq U_d(x, g'(x)) \forall g'(x) \in T$$

Where T is the set of attacker strategies with equal expected attacker utility.

This equilibrium is guaranteed to exist (von Stengel 2004). The Weak Stackelberg Equilibrium (WSE), on the other hand, is not. This equilibrium concept is the same as the SSE with the third condition inverted. That is, the attacker breaks ties by choosing the *worst* strategy for the defender. As the WSE is not guaranteed to exist, the SSE is used as the standard solution, with the justification that the attacker can be induced to choose the best strategy for the defender by trivial adjustments to the defender's strategy.

Green Security Games

GSGs (Fang, Stone, and Tambe 2015) define a specialized form of SSGs designed to be applicable to conservation problems. This model has two key differences: firstly, GSGs specifically focus on repeated game settings where there are multiple *rounds* of the game being played. In each round, there are multiple episodes. For the duration of a round, the defender commits to a mixed strategy, while each episode considers a single pure strategy drawn from this mixed strategy. As in SSGs, the attacker(s) commit to an attack based on their knowledge of the defender. The second key difference from SSGs in general is that the GSG attacker does not have a perfect knowledge of the defender's mixed strategy. Instead, each attacker is modeled with a memory length parameter, as well as a parameter controlling how much consideration is given to each historical timestep.

Human Behavior Modeling

While modeling attackers as perfectly rational is simple, real world adversaries don't conform to this assumption. Due to many factors including imperfect knowledge of the world and human emotions, attackers are unlikely to choose their targets optimally. To address this, multiple approaches modeling human behavior have been used, including MATCH (Pita et al. 2012) and Quantal Response (McKelvey and Palfrey 1995; Yang et al. 2011).

MATCH uses robust optimization techniques to create defenses that can perform well against attackers of various behavior. Furthermore, it doesn't rely on any sort of attacker behavior modeling. Instead, it's singularly controlled by a parameter which dictates the tradeoff between defender utility when the attacker plays according to best response and robustness to less predictable attackers.

For our work, we focus on Quantal Response and its variants:

Quantal Response (QR). QR is an well-known model describing attacker behavior in SSGs (McKelvey and Palfrey 1995; Yang et al. 2011). Intuitively, QR provides a mechanism for partially rational behavior where higher expected utility targets are attacked more frequently.

Essentially, the probability of attacking target i is given as follows:

$$q_i(\mathbf{x}; \lambda) = \left(e^{\lambda U_i^a(x_i)}\right) \Big/ \left(\sum_j e^{\lambda U_j^a(x_j)}\right) \quad (1)$$

This model describes the attacker with a single parameter, λ, governing its rationality. As λ approaches 0, the attacker becomes completely random. As λ approaches ∞, the attacker becomes perfectly rational.

Maximum Likelihood Estimation For computing λ, the traditional method is to use maximum likelihood estimation over the historical data. This yields the most likely λ matching the observed attack pattern:

$$\lambda^{learnt} = \texttt{argmax}_\lambda \sum_m \sum_i z_i^m \log q_i(x_i^m, \lambda)$$

where x_i^m is the defender's coverage probability of target i at timestep m and z_i^m is the observed number of attacks at that timestep. This allows the defender to learn a single parameter, λ, using historical data. Then, the defender can predict future attacker behavior using that λ, and optimize its defense accordingly.

Defense Against QR Attacker To defend against such an attacker, BRQR (Best Response to Quantal Response) was proposed (Yang et al. 2011). The defender's optimization problem is defined as follows:

$$\max_x \sum_i q_i U_i^d(x_i)$$
$$\text{s.t.} \sum_i x_i \leq T$$
$$0 \leq x_i \leq 1, \forall i$$

As this objective is generally non-convex, finding the global optimum isn't feasible. Instead, the general approach used is to find multiple local optima from different starting points, and to take the best one found (Yang et al. 2011).

Subjective Utility Quantal Response (SUQR). In SUQR, the attacker's perceived utility of attacking each target is calculated differently. Rather than computing the actual expected utility, SUQR uses a linear combination of some information available to the attacker. The attacker considers for each target the coverage probability, the reward for a successful attack, and the penalty for a defended attack:

$$\hat{U}_i^a = w_1 x_i + w_2 R_i^a + w_3 P_i^a$$

The attack probabilities, then, are given by:

$$q_i(\mathbf{x}; \lambda) = \left(e^{\lambda \hat{U}_i^a(x_i)}\right) \Big/ \left(\sum_j e^{\lambda \hat{U}_j^a(x_j)}\right) \quad (2)$$

Intuitively, this allows attackers to give different weights to the defender coverage, the reward, and the penalty than the objective expected utility calculation does. This model was shown to outperform both regular QR as well as MATCH (Nguyen et al. 2013).

Other Models Another model called CAPTURE (Nguyen et al. 2016) aims to improve upon the shortcomings of SUQR. Firstly, this model considers attacker behavior at each time step to be related to behavior at prior time steps, rather than independent as in QR models. Next, the model incorporates a larger range of domain features (e.g. slope and habitat) than SUQR does. Third, CAPTURE incorporates observational uncertainty on the part of the defender, which is modeled as depending on the domain features, the underlying behavior of the attackers, and the defense strategies during the observation. Lastly, attack probabilities are calculated independently per-target. Experimentally (using real world data), this model was shown to significantly out-perform SUQR.

Noting the complexity and poor interbretability of CAPTURE, researchers were motivated to create a simpler model, INTERCEPT (Kar et al. 2017). Their underlying approach uses decision trees to produce effective and interpretable models. To handle the spatial challenges of the space (e.g. addressing the continuous nature of real world terrain), they draw on criminology's theory of "hot spots" which are points where crime (in this case poaching) is likely to be common. Then, they utilize the distance from expected hot spots as another input to the decision trees. Lastly, they utilized ensemble learning by creating different expert models (limited to 5 for interpretability reasons) that will then vote on the attack likelihood of each target. Experimentally, their model was shown to significantly outperform CAPTURE, despite being far simpler and computationally cheaper.

Game Theoretic Deception

While modeling attacker behavior allows for a better defense, it does present a vulnerability. Namely, that the defender must utilize historical attack data to form a model of the attacker. If a particularly clever attacker were to change its attack pattern, knowing that data collection was in progress, it could alter the learning results and find advantage in the resulting strategy. Recently, research has investigated this kind of deception from the attacker side (Gan et al. 2019b; Nguyen et al. 2019; Zhang, Wang, and Zhuang 2021) in SSGs, and the follower side in general Stackelberg games (Gan et al. 2019a). This type of attack is analogous to a poisoning attack in adversarial learning.

One such work considers multiple *types* of attackers, corresponding to different rewards and penalties for each target. To deceive the defender, an attacker could then pretend to be a different type and play accordingly during the learning phase (Nguyen et al. 2019). Then, after the defender has created its strategy, that attacker can play optimally, gaining advantage from the earlier deception's influence on the resulting strategy.

Addressing this imitative deception has also been studied (Nguyen, Butler, and Xu 2020). This work introduces an exact equilibrium formulation for repeated SSGs, as well as using this formulation to devise an optimal counter to the aforementioned deception. However, the authors note that, given the repeated game setting, considering both historical data and future expected utility exponentially compounds this optimization problem. To address this, they introduce limited memory and limited lookahead heuristics. Their experimental results show that addressing the deception, with or without heuristics, yields significantly better utility for the defender, and worse utility for the attacker, than naively ignoring the deceptive behavior.

Another approach (Nguyen, Sinha, and He 2020) considers a realistic scenario in which the defender must contend with multiple attackers of *unknown* behavior. These attackers are then modeled by the defender with QR, using a single λ to describe the attacker population. The deception, then, takes the form of an attacker playing according to some λ to skew the learning result for the entire population, altering the defender's resulting strategy. Again, after the learning phase, the attacker can play optimally to take advantage of the altered strategy.

Noticing the advantages of this form of deception, researchers were motivated to study counterstrategies (Butler, Nguyen, and Sinha 2021). Their approach relies on characterizing the possible deceptive space of the attacker and then using a maximin optimization to form an effective strategy against it. Using binary search, a defender can find both the minimum and the maximum possible λ parameter for the *non* deceptive attacker population (which was concealed by the deceptive attacker polluting the collected historical data). Then, the defender can optimize its strategy against both attackers (the deceptive, fully rational one and the boundedly rational population) using a maximin over the range found by the binary search.

One limitation of the two previously discussed deception approaches is that they only consider a one-shot game, where the attacker has no incentive to play dishonestly after the initial learning phase. A newer paper (Nguyen and Sinha 2022) explores a repeated game setting where the attacker must consider the longer term. The authors use projected gradient descent to solve the attacker's nested optimization problem and find its deception strategy. Their experimental results (on repeated games of 4 and 8 timesteps) show significantly higher utility for the attacker, and lower utility for the defender, compared to the case where the attacker plays honestly.

While studying attacker deception is relatively new, deception from the defender side has been more well considered (Zhuang, Bier, and Alagoz 2010). One such work (Zhuang, Bier, and Alagoz 2010) described information concealment by the defender. Their setting was a multiple timestep, general sum game in which the defender could invest more resources between timesteps, and the attacker could learn more information based on observing signals and on results of attacks. Their findings showed that it can be in the best interest of the defender to conceal information (e.g. leading attackers to believe that targets are better defended than they actually are).

Similarly, a study shows selectively revealing information can improve outcomes for the defender (Rabinovich et al. 2015). The authors consider a Stackelberg security game setting in which allocation of defender resources to targets may not be visible to the attacker. They then investigate what resource assignments the defender should reveal to the attacker, finding that this selective disclosure can be a pow-

erful deterrent, improving outcomes for the defender. Furthermore, they note that acting on this information (updating their strategy) will still be in the attacker's best interest, even if they know that it was intentionally revealed as a deterrent.

Another work (Guo et al. 2017) compared the utility of *signaling* (openly flaunting defense resources) to concealment, showing that there they can both be advantageous depending on the payoff structure of the game itself. The authors are able to formalize the tradeoff between concealment and signalling/commitment. Furthermore, their results show that the boundary of this tradeoff is close to zero sum.

Data Based Decision Making

In security games, data can be collected by the defender to form a model of the attacker's behavior and optimize a defense against it. Similarly, data based decision making uses some data with the ultimate goal of producing a decision. Despite the differences between these fields, their mutual reliance on data produces similar vulnerabilities. In this section, we first give an overview of the field, and then detail the two most common approaches (two-stage and decision focused) for solving the problem of interest. Afterwards, we discuss applications using these approaches, as well as some social good applications that don't fall neatly into either category.

Data-based decision making refers to a common paradigm in real world artificial intelligence applications in which we are concerned with three related pieces of information: directly observable data (denoted by u), data that will be unobservable at test time (denoted by θ), and a *decision* that must be made (denoted by x). The decision, x, depends directly on θ, which in turn can be predicted based on u. The ultimate goal in a data-based decision making problem is to find an optimal decision to maximize a utility function, abstractly represented as follows:

$$\max_{x \in X} f(x, \theta)$$

where x is the decision variable and $X \subseteq \mathbf{R}^K$ is the set of all feasible decisions. Note that the objective, f, depends directly on the *unobservable* parameter θ, which must be inferred from the correlated observable data, u.

There are two common ways of solving data based decision making problems. First, and most well established, is the two-stage approach. Here, the task is split into two separate steps. The predictive component (i.e. a neural network) is first trained directly to learn the relationship between θ and u, taking u as the input and outputting a prediction for θ, denoted $\hat{\theta}$. Next, we have the planning or optimization step, in which $\hat{\theta}$ is used to optimize the final decision, x.

While the two-stage approach would be optimal if we could perfectly predict θ from u, in realistic settings, errors are inevitable. Using imperfect predictions to optimize our decision may result in compounding errors, and notably worse decision quality. To address this shortcoming, an approach called *decision focused* learning seeks to bridge the disconnect between the end goal of the system and the learning result. That is, rather than training a model for predictive accuracy, it is directly trained to maximize decision quality.

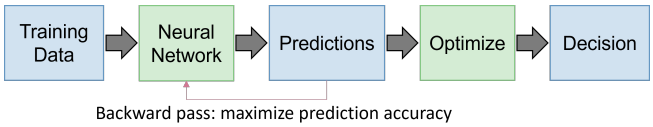

Figure 1: Depiction of a two-stage learner

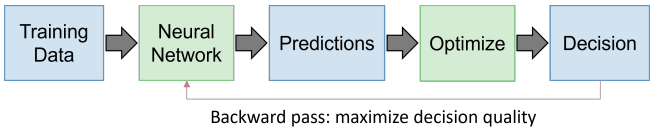

Figure 2: Depiction of a decision focused learner

The naive approach to this would be to have the network directly output x, and bypass prediction of θ entirely. However, in practice, training a neural network to solve optimization problems is a difficult task. Instead, decision focused learning approach still uses the model to predict θ. The innovation here, then, is to differentiate *through* the solution to an optimization problem, allowing the model to be trained based on the solution quality, while still incorporating a convex optimization solver (Wilder et al. 2020).

Two-Stage Formulation

For the two-stage approach, the first stage is predicting the unknown parameter θ from the observed feature vector u. The second stage, then, is to compute the optimal x given the predicted θ (Figure 1). Predicting the *unknown* parameter θ can be done using a parametric model trained for the task, denoted by $\hat{\theta} = g(u, w)$. Here, w represents the model's parameters where the learner seeks an optimal set of model parameters, w^*, that minimize the training loss, abstractly formulated as follows:

$$\min_w \mathcal{L}(\mathcal{D}, w)$$

For example, using mean squared error as the training loss:

$$\mathcal{L}(\mathcal{D}, w) = \frac{1}{n} \sum_i (\theta_i - g(u_i, w))^2$$

Once the model has been trained (yielding w^*), the decision maker can use observed u values to predict a θ value ($\hat{\theta} = g(u, w^*)$), then use that prediction to find an optimal decision by solving the following optimization problem:

$$\max_{x \in X} f(x, g(u, w^*))$$

Decision Focused Formulation

We focus on the problem setting in which the decision optimization is convex, meaning that the objective is convex with respect to the decision variable, x. This convexity setting has been widely considered in previous studies on data-based decision making (Wilder et al. 2020; Wilder, Dilkina, and Tambe 2018; Donti, Amos, and Kolter 2017; Agrawal et al. 2019).

Based on this convexity characteristic, we can leverage the implicit function theorem (Krantz and Parks 2002) to

differentiate through the decision-optimization component (computing $\frac{dx}{d\hat{\theta}}$). The decision-optimization is formulated as a convex optimization problem:

$$\max_x \ f(x, \hat{\theta}) \ \text{ s.t. } \ Ax \leq b$$

Since this is a convex optimization problem, any solution that satisfies the following KKT conditions is optimal:

$$\begin{aligned} &-\nabla_x f(x, \hat{\theta}) + \lambda \cdot \nabla_x (Ax - b) = 0 \\ &\lambda \cdot (Ax - b) = 0 \\ &Ax \leq b, \lambda \geq 0 \end{aligned}$$

where λ is the dual variable. Observe that the first equation indicates that x and λ are functions of $\hat{\theta}$. Based on the implicit function theorem, we can differentiate through the first two equations to obtain the following gradient formulation:

$$\begin{bmatrix} \frac{dx}{d\hat{\theta}} \\ \frac{d\lambda}{d\hat{\theta}} \end{bmatrix} = \begin{bmatrix} \nabla_x^2 f(x, \hat{\theta}) & A^T \\ diag(\lambda)A & diag(Ax - b) \end{bmatrix}^{-1} \begin{bmatrix} \frac{d\nabla_x f(x, \hat{\theta})}{d\hat{\theta}} \\ 0 \end{bmatrix} \tag{3}$$

Solving this system gives us $\frac{dx}{d\theta}$ which then allows us to directly optimize the predictive component of the model for decision quality using standard gradient descent based methods. The primary disadvantage of decision focused approaches is the increased computational cost. Every training instance requires both solving and backpropagating through this optimization, rather than computing the training loss based directly on the model output as in two-stage methods.

Two-Stage Applications

Data-based decision making encapsulates a wide range of applications and approaches. In this section, we'll begin by covering some works following the two-stage approach.

COPE (Wang et al. 2006) uses observed data to construct a convex hull containing expected future traffic demands. Then, they solve a linear program in order to optimize traffic routing. Additionally, their method provides a worst-case guarantee for unprecedented or unpredictable future scenarios.

One application (Xue et al. 2016) uses a two-stage approach for reducing bias in citizen science (i.e. models built on crowd sourced data). More specifically, they consider a bird observation collection app (eBird) and seek to gamify observation collections so that contributors will provide more balanced data. The learning task here is to model the user's preferences (which determine how rewards scale, i.e. less preferred tasks require higher reward) based on observations of past behavior. Next, the decision-optimization component incorporates both the user's and the organizer's goals by transforming the user's goals into constraints on the organizer's objective. Finally, solving this optimization problem yields rewards that were shown to effectively incentivizes the users to explore under-observed areas, resulting in more balanced observations and lower bias.

In wildlife conservation, PAWS (Fang et al. 2016) uses a two-stage approach to protect wildlife and combat poachers. The predictive portion of the task is using past observations of poacher and animal activity to model the poachers' behavior using SUQR. Then, the decision-optimization task is to optimize ranger patrols using that model. Notably, this was the first deployed security game application considering imperfectly rational attackers as previous deployments assumed full rationality.

Another application (Mukhopadhyay and Vorobeychik 2017) seeks to optimize allocation of emergency responders. For the learning task, they use features such as weather, season, and transportation network details to predict both the timing and severity of potential incidents requiring emergency services. Then, ask the decision-optimization component, they use a greedy approach to solve a non-linear, non-convex optimization problem to yield desirable placements for emergency responder facilities.

Two-stage approaches also find application in graph based problems. One such work (Yan and Gregory 2012) considers community detection in the case where edge information is unknown. First, for the predictive task, they utilize edge prediction based on vertex similarity to learn the weights in a previously unweighted graph. Then, as the decision-optimization component, the authors use several standard community detection algorithms and compare results between them. Overall, this work demonstrates the value of making predictions in graph optimization tasks, rather than trying to make do with only the directly observable information.

Decision Focused Applications

While newer than the two-stage approach, a variety of works have considered decision focused approaches to solving data-based decision making problems. One (Wilder, Dilkina, and Tambe 2018) introduces a general formulation for decision focused combinatorial optimization problems, using linear programming and submodular maximization as examples. To bridge the gap between the optimization component and the model, the authors leverage the KKT conditions on the implicit function theorem, as described previously. Their experimental results across three different problems (budget allocation, bipartite matching, and diverse recommendations) demonstrate overall better solution quality than the two-stage approach, despite less accurate predictions from the predictive model. These results suggest that maximizing predictive accuracy is often a poor proxy for maximizing the final decision quality.

Similarly, another work investigates a decision focused approach for stochastic optimization (Donti, Amos, and Kolter 2017). Once again, they utilize the KKT conditions and the implicit function theorem to differentiate through the solution to an optimization problem, using the derived gradient to train the predictive component. The author's experiments consider three different applications. These are a synthetic data inventory stock problem, and two real world applications: energy scheduling and battery load arbitrage. Their results demonstrate both higher utility and lower variance than the corresponding two-stage approaches.

Another paper (Wilder et al. 2020) applies decision focused learning to graph optimization problems. The ap-

proach the authors consider starts with a graph embedding network that encodes the graph's adjacency matrix along with any available node information. Then, as the decision-optimization component, they incorporate a differentiable optimization layer that performs K-means clustering. This generalized approach can be seen as an analogous to many common graph problems, including maximum coverage and community detection. Solving the backwards pass uses the implicit function theorem to compute gradients. However, instead of using the KKT conditions of the optimization problem's solution, they directly characterize the optimization update process and compute gradients accordingly. Another contribution of this work is introducing a heuristic for this computation, significantly reducing the computational complexity of the backwards pass. Essentially, the authors find that, in practice, the K-means cluster assignments change little in each optimization step. When that holds, the gradient of the objective with respect to the cluster assignments can be approximated as the identity matrix.

Observing the computational complexity of decision-focused learning approaches, researchers are motivated to examine heuristics. One such work (Wang et al. 2020) investigates learning surrogates for decision-focused optimization problems, seeking to preserve the advantages of the decision-focused approach while addressing the discouraging compute requirements. The authors utilize a *learnable* reparameterization matrix and incorporate it into the model. This allows for dramatic (but lossy) simplification of the decision-optimization problem, and allows loss based on the final solution quality to train both the predictive component and the reparameterization component. Another advantage of this surrogate approach is that it's less prone to getting stuck in local minima than both the decision-focused and two-stage approaches due to the gradient sparsity alleviating effects of the reparameterization. Experimentally, their results demonstrate the value of the surrogate approach, showing significantly lower runtime and/or significantly better solution quality than the decision-focused approach, and strictly better solution quality than the two-stage approach.

In the security game domain, researchers (Perrault et al. 2019) leverage a decision focused approach to optimize defender utility. The predictive component in this setting is designed to learn the attacker's behavior (e.g. the target weights in SUQR). The decision-optimization component, then, is to optimize the defender's strategy accordingly. While the optimization problem here is generally non-convex, the local region is generally convex for boundedly rational attackers. This allows them to utilize the KKT conditions of the implicit function theorem to compute the gradient, enabling direct optimization of the predictive component based on solution quality. Lastly, their experiments show higher quality solutions across a variety of settings (including real-world human attacker data) than two-stage approaches.

In wildlife conservation, researchers (Xu et al. 2020) were motivated to investigate improving on PAWS by using a decision focused approach rather than the original two-stage approach. Furthermore, they account for uncertainty in observations of poacher behavior by incorporating Gaussian processes into an ensemble learner, which allows them to quantify the uncertainty of observations in each section of a park. Leveraging this knowledge allows them to create more robust strategies, minimizing the harm done by imprecise observations. Experimentally, this end-to-end method increased detection of poaching by 30%, showing the value of decision focused approaches when observed data isn't fully reliable.

Social Good Applications

Though they may not perfectly fit into the "predict-then-optimize" framework, a variety of social good applications follow the general philosophy of data-based decision making. One such work (Zhang et al. 2023) uses a large language model, RoBERTa (Liu et al. 2019b) to automate triage of pregnant people with health concerns in Kenya. The model takes in questions sent over text message and attempts to classify their problem based on a set of pre-defined common concerns among pregnant people. If the predicted problem isn't severe, and the classification confidence is high, an automated response is sent. If either of these things are not true, the problem is referred to human health desk staff. The main challenge in this work lies in the text messages - they contain natural language including slang, and, to further complicate things, mixed English-Swahili text. Their final system shows high classification performance on problems of interest and is able to reduce the workload of the health desk workers.

Another work in maternal healthcare (Mate et al. 2021) also considers an automated messaging system. However, in this work, the goal is to optimize limited intervention resources to prevent dropouts. They use restless multi-armed bandits (RMABs), a reinforcement learning technique, to model the problem. The goal is to predict which participants will benefit most from an intervention, given their behavioral history. To deal with scaling issues and lack of data for new participants, they cluster participants into groups and use a single RMAB for each group. Their results show that the selected interventions were significantly better than randomized interventions, highlighting the benefit of optimizing resources in similar health applications.

Also in public health, research (Killian et al. 2019) has investigated a decision focused approach for targeting interventions for improving adherence to tuberculosis treatment plans. This is accomplished via using various features (such as recent call data and demographic information) of the patients to predict which ones are likely to stop adhering to the treatment plan. The paper considers a random forest as well as an LSTM based model (which proves more effective), as the data is comprised of time series information for each participant. Furthermore, the authors investigate using the same models to predict the *effectiveness* of the interventions, which is where decision focused learning comes in (i.e. training a model directly to maximize the effectiveness of interventions selected, rather than just predicting what participants may need intervention). The decision-focused learning approach yields less accurate predictions, but results in 15% higher total intervention utility than the two-stage counterpart.

Similarly, research (Yadav et al. 2018) has investigated

using AI to optimize interventions among homeless youth to raise HIV awareness. This is done by forming a model of the community structure within the population of homeless youth, and selecting individuals who will most effectively spread information to others. Notably, the models formed of the community are never fully accurate, and uncertainty in some information (such as exactly how likely individuals are to pass on information to each other) has to be accounted for. The authors here perform a pilot study in the real world comparing two methods. First is HEALER (Yadav et al. 2016) which leverages Partially Observable Markov Decision Processes (POMDPs) to select optimal interventions. Notably, using a single POMPD for each possible combination of interventions results in an intractable problem. Thus, HEALER breaks the graph down, and uses multiple nested levels of POMDPs to overcome this challenge. The next method under evaluation is DOSIM (Wilder et al. 2017), which uses a game theoretic approach with robust optimization. To make the problem tractable, the authors use the double oracle approach to find an approximate equilibrium. Notably, this method yields *mixed* strategies (which allows for more robust policy selection), while HEALER only gives pure strategies. In practice, both methods gave similar results, giving 160% more information spread compared to the baseline (degree centrality).

In AI for education, research often involves predicting student performance. One such work (Su et al. 2018) uses a recurrent neural network (a modified LSTM) to perform this task, incorporating *both* the performance history of students as well as the text of the exercises in question. One of the key challenges in this area is known as the "cold start" problem, referring to the difficulty of predicting performance of new students or on new exercises. Their methods outperform baselines, particularly in the cold start setting, by incorporating correlations between exercises. While they don't consider any task *after* this predictive stage, their predictions could be used for objectives such as recommending tutoring, sending automated informative messages, and deciding what subjects should be covered in more depth.i

Adversarial Learning

Our work has considered attacks in both game theoretic and data-based decision making settings. Specifically, we've investigated attack scenarios where an adversary can manipulate some portion of the training data. In the field of adversarial learning, this would be considered a *poisoning attack* (or backdoor attack). By way of contrast, an *evasion attack* (or adversarial example) occurs at test time, seeking to manipulate the model's output for specific samples. *Exploratory attacks* work in another direction entirely, using their attack capabilities to learn more details about the system. Here, we pay extra attention to poisoning attacks as they are the most relevant to our work.

As graph learning problems often follow the data based decision making paradigm (e.g. using node features to predict edges and then performing bipartite matching on the edge predictions) we spend more time on this domain than others. After discussing poisoning attacks and evasion attacks in graph learning and deep learning, we detail the current research into defense and robustness.

Direct attacks to data-based decision making models are relatively unexplored. To the best of our knowledge, our work (Kinsey et al. 2023) is the only paper in this domain. We utilize the metagradient method to optimize poisoning attacks against data-based decision making models, investigating both the two-stage approach and the decision focused approach as targets. Furthermore, we evaluate the effectiveness of using a simpler model (i.e. one trained to directly output the decision) as a vector for generating attacks that will then be transferred. Experimentally, our results are mixed. Directly attacking the decision-focused learner is infeasible due to the computational requirements of solving the attacker's optimization problem. Attacks from the two-stage learner do transfer effectively to the decision focused learner, though generating these attacks is still difficult. Due to the complexity and non-convexity of the attack space, obtaining the global optima is implausible, and even finding a good local optima isn't guaranteed. Attacking the simpler model, on the other hand, is entirely ineffective. Future work in this area should consider approximate metagradients or non metagradient based methods for attacking data-based decision making model.

Attacking Graph Learning

Problems across many domains including social networks, city planning, network security, and biology can be modeled as graphs. This has led to significant study of graph learning in recent years, particularly using deep learning on graphs (Kipf and Welling 2017; Bojchevski and Günnemann 2018; Klicpera, Bojchevski, and Günnemann 2018; Monti et al. 2017). Anomaly detection in this field is well studied (Akoglu, Tong, and Koutra 2014) based on the observation that learning results on the entire graph can be compromised via anomalous individual nodes. However, until more recently, intentional attacks on graph learning problems was an unexplored area of research.

Evasion Attacks One early work in this area investigates adversarial example generation for the link prediction task (Minervini et al. 2017). In their setting, an adversary generates examples maximizing the *inconsistency loss*. This loss is calculated by first identifying constraints on non-adversarial inputs, and then measuring how much a given example violates those constraints. The learner (or discriminator) then makes use of this inconsistency loss as a regularization term when training on generated adversarial examples. Surprisingly, they are able to find efficient closed-form solutions for the adversarial generation task against several popular link prediction models. Their experimental results show that incorporating adversarial examples in this manner improves the performance of these link prediction models, particularly when limited training data exists.

Primarily motivated by network security problems, another work (Chen et al. 2017) investigates attacks on community detection tasks by an adversary without perfect knowledge. The authors introduce two different attacks: *targeted noise injection* and *small community*. As in the name,

targeted noise injection adds some noise to the graph structure, creating new edges in a way that imitates the structure of the true graph. The small community attack, on the other hand, aims to create smaller clusters in the graph by removing edges and/or nodes. For defences, the authors recommend re-training on adversarial examples (altered by the noise injection) and specifically tuning hyperparameters based on performance against the small community attack. Experimentally, they found the attack to dramatically reduce model performance when unaddressed, but that their suggested defences are effective.

In the domain of social networks, researchers have studied attacks on community detection problems. One work (Waniek et al. 2018) considers an attack with the primary goal of obscuring the importance of a single individual (denoted $v^\dagger$) in a community (e.g. concealing the leader of a terrorist cell). The secondary goal of this attacker is to hide the community entirely. They also present simple (such that they could be used by attackers without mathematical or technical requirements) heuristics for both of these goals. For hiding individuals, ROAM (remove one, add many) removes the link between $v^\dagger$ and some v_0, then connects v_0 to up to $budget - 1$ other neighbors of $v^\dagger$. This reduces the closeness centrality of $v^\dagger$ and its degree, while increasing the closeness centrality and the degree of its chosen neighbor, v_0. For hiding communities, their DICE (disconnect internally, connect externally) heuristic first disconnects d (where $d \leq budget$) links within the community, and then creates $budget - d$ links from within the community to outside nodes. Note that this method is concerned with a single community/individual.

For a more global attack on community detection, work (Chen et al. 2019) utilize a genetic algorithm to generate adversarial examples. Their results show that their method outperforms simpler heuristics they propose, Community Detection Attack (CDA) and Degree Based Attack (DBA). CDA randomly selects a node in each community to remove random inter-community links, and add intra-community links. DBA is identical, except instead of randomly selected nodes, it targets the highest degree node in each community. Furthermore, their results show significant transferability of their GA generated adversarial examples across different types of target models.

Many graph learning approaches use some kind of lower dimensional representation of nodes, done via some machine learning node embedding process such as DeepWalk (Perozzi, Al-Rfou, and Skiena 2014), LINE (Tang et al. 2015), or node2vec (Grover and Leskovec 2016). Then, these node embeddings can be used for a variety of downstream tasks. Researchers are thus motivated to investigate attacks on the node embeddings as general purpose adversarial examples. One work (Sun et al. 2018) targets node embeddings and uses link prediction as the downstream task of interest. Their approach makes use of the KKT conditions to differentiate through the node embedding process and then optimizes adversarial graph modifications via projected gradient descent. The authors consider two specific attacks: *integrity attack* which targets specific links and *availability attack* which seeks to maximize overall prediction errors. Both are accomplished by adding or moving edges. Their results show the effectiveness of their technique, even with a budget of relatively few edges. Once again, attacks generated by this method are shown to transfer effectively between different node embedding techniques.

Poisoning Attacks The first work to consider training time attacks on deep learning for graphs (Zügner, Akbarnejad, and Günnemann 2018) targeted the node classification task. Their approach allows for both structural attacks (modifying edges) and feature attacks (modifying node features), and seeks to create unnoticeable perturbations by preserving degree distribution and feature co-occurrence statistics (e.g. ensuring that features never seen together in the original graph don't appear together in the modified graph). To make the computations tractable, the authors target a *surrogate* model to produce their attack, and then transfer it to the final model. Experimental results demonstrate the effectiveness of this attack, transferring successfully to other semi-supervised graph learning methods, and, notably, to the unsupervised method DeepWalk (Perozzi, Al-Rfou, and Skiena 2014).

By way of contrast, another work (Bojchevski and Günnemann 2019) directly targets unsupervised methods for node embedding. This setting presents additional challenges: no labels exist to exploit, and many unsupervised node embedding methods (such as those based on random walks) prevent direct gradient calculations. Instead, they utilize matrix perturbation theory (Stewart 1990) to efficiently approximate the loss function of DeepWalk (Perozzi, Al-Rfou, and Skiena 2014) and compute their attack, which takes the form of added and removed edges. They consider three general attack types: first, the general attack seeking to maximize the node embedding loss; second, a targeted attack seeking to change the classification of a specific node; third, a targeted attack seeking to prevent link prediction between a set of node pairs. Experimentally, they show that their attacks are effective even when the allowed number of edges flipped is low. Furthermore, they once again demonstrate transferability of their attacks between a variety of models.

Investigating poisoning attacks on the node classification task, another work (Zügner and Günnemann 2019) leverages meta learning to *directly* solve the bilevel optimization underlying the poisoning attack. Essentially, this requires unrolling the training process of the classifier (each step of training itself being differentiable) and computing the gradient of the resulting weights with respect to the training data. Rather than considering specific nodes, their goal is to decrease the overall accuracy of the classifier. Additionally, they provide memory efficient heuristics for the metagradient calculation. The experimental results provided demonstrate the effectiveness of the main method, as well as the heuristics, decreasing overall classification performance of the target models even with small perturbations in the training data.

A more recent paper (Zhang et al. 2020a) investigates attacks and defences for graph neural networks under the *label flipping* setting. Here, the attacker's power is limited to

changing the labels of nodes (considered to be binary) in the training set. To solve this attack, the authors come up with a closed-form approximation for the classifier (a GCN here) as well as transforming the discrete components of the attack objective into continuous surrogates. This allows them to avoid directly computing the metagradient, as (Zügner and Günnemann 2019) did. For defence, they propose a self-supervised community labelling task as a regularization method during the training process. Their experiments on several real world datasets demonstrate the value of the attack as well as the effectiveness of their proposed defence.

Targeting classical methods for graph learning (rather than the relatively new methods considered in the previously mentioned works) researchers (Liu et al. 2019a) seek to create a unified framework for poisoning attacks on semi-supervised graph learning problems, particularly focusing on the label propagation method. Their framework considers both classification tasks and regression tasks, and presents novel approaches for solving both. Experimental results demonstrate that, even with very few perturbations, their methods can significantly decrease classification accuracy or increase regression loss.

Two simultaneous works (Zhang et al. 2020b; Xi et al. 2020) first considered *backdoor* attacks on graph neural networks. These are a special case of poisoning attack where the attacker seeks to influence the model to classify test time examples with some *trigger* present as a specific class. Furthermore, the attack is designed to not impact performance on clean test examples (those without the trigger present).

The first of these works (Zhang et al. 2020b) seeks to directly produce a graph neural network that is susceptible to these triggers, given a pre-trained clean GNN and the data that will be used for downstream classification (using the node embeddings produced by the GNN). Interestingly, they tailor the triggers (which take the form of subgraphs) to each graph in question, rather than using a one-size-fits-all approach. Their results show how effective such an attack can be, and they provide analysis of the threat model and its limitations.

The other work (Xi et al. 2020) takes a different approach to this backdoor attack. Rather than trying to produce an altered GNN, this method seeks to alter training data by injecting a trigger (again taking the form of a subgraph) as well as arbitrarily changing the label. For this trigger, they randomly (using various methods to ensure similarities to the real data) generate a subgraph to insert. Interestingly, their results show that fixing this subgraph (one randomized trigger shared across every poisoned training and test instance) barely performs better than each subgraph being individually randomized. In addition, the authors provide a certified defence against this treat model. Their experimental results show the effectiveness of the attack, however, their certified defence is ineffective in some settings, necessitating further study.

In a more recent paper, researchers (Zheng et al. 2022) propose a new approach to backdoor attacks on GNNs, based on *motifs* which are recurrent and statistically significant subgraphs. To select the trigger, then, they analyze the motifs in available graphs, and construct an appropriate trigger. Their experimental results demonstrate more effective attacks than existing methods, as well as ensuring the target model's performance on clean test instances isn't compromised.

Attacking Deep Learning

Attacks to deep learning systems in general are much more well-researched, especially in computer vision, than those targeted against graph learning models.

Evasion Attacks Adversarial examples targeted against deep learning models were initially introduced by researchers (Szegedy et al. 2013) who noticed that imperceptible modifications could cause an image to be misclassified by image classification models (Yuan et al. 2017). Furthermore, they found that adversarial examples generated against one network transferred effectively to other models with different architectures or even different training data sets.

While effective, the method introduced by the previous paper was inefficient, and relied on a linear search to find the best imperceptible perturbation. To address this flaw, the Fast Gradient Sign Method (Goodfellow, Shlens, and Szegedy 2014) was introduced. Intuitively, this method computes the gradient of the classification loss exactly once. Then, each pixel value is modified with the same magnitude, based on the sign of the gradient with respect to that pixel. Similarly, the Fast Gradient Value method (Rozsa, Rudd, and Boult 2016) also computes the gradient exactly once. However, they modify each pixel with the raw gradient value, rather than making modifications of the same magnitude to each pixel. Note that this allows larger per-pixel modifications than the previous method.

Seeking to improve on the weaknesses of single step adversarial example generation (imprecision and relatively easy defense primarily) as well as the weaknesses of traditional iterative methods (getting stuck in local optima, unstable optimization) researchers (Dong et al. 2017) applied momentum to the gradient descent method of optimizing adversarial examples. Additionally, they formulated their attacks against an ensemble of models (via averaging their logit outputs) to generate broadly applicable attacks. Their experimental results demonstrate better attack performance than the single step or iterative (without momentum) methods.

Working in a different direction, DeepFool (Moosavi-Dezfooli, Fawzi, and Frossard 2015) seeks to understand adversarial examples and improve model robustness by efficiently and precisely computing adversarial perturbations. Specifically, their method iteratively linearizes the classification model, and then computes a minimal step to take, repeating until the class of the target instance changes. Intuitively, this method seeks to find the minimal possible perturbation that produces the desired misclassification. Experimentally, they demonstrate that they are able to produce adversarial examples more reliably than previous methods, and training models on their examples significantly improves robustness.

Another creative work (Su, Vargas, and Sakurai 2019) explored the viability of attacking a single pixel. Their method uses differential evolution (a genetic algorithm that does not require computing gradients) to produce this extremely limited attack. Ultimately, they are able to change the classification of 67% of images in the CIFAR-10 dataset, and 16% of the images in the ImageNet dataset, despite only modifying one pixel per image.

Poisoning Attacks One early work investigating poisoning attacks on deep learning models (Muñoz-González et al. 2017) uses the metagradient method to optimize its attack. The intuition here is that, when a model is being trained, each update is itself a differentiable operation. By unrolling these updates, an attacker can compute the gradient of the final weights with respect to the training data, enabling poison attack optimization via gradient descent. To make the gradient calculation tractable, they consider only a few steps of updates to the model while training. Additionally, they find that attacks generated can be transferred effectively to different training algorithms.

Improving on the previous work, MetaPoison (Huang et al. 2020) employs the same metagradient method except on an ensemble of target models, each at different stages of their training process. By averaging the attack gradients over all of them, the authors are able to create robust attacks transfer effectively across models. Another improvement along these lines is Witches' Brew (Geiping et al. 2020) which introduces a *gradient alignment* component to the attack. Overall, they seek to match the direction of the attacker's loss gradient on a target image with the classifier's loss gradient on that image. Intuitively, what this does is ensures that when the classifier takes an optimization step based on that image, it is also reducing the attacker's loss on that image, furthering the poisoning attack's goal.

Rather than directly or approximately trying to solve the bilevel problem underlying the poisoning task, some methods train generative models to directly produce poisoned images. One such work (Yang et al. 2017) uses autoencoders to speed up the poison generation process. Experimentally, the computation time is significantly lower, though their generated attacks are on average less effective than the iteratively produced baseline. Another (Muñoz-González et al. 2019) uses a GAN based model where the generator is trained against a classifier *and* against a discriminator (which seeks to detect the difference between a poisoned instance and a clean one). In contrast with the previous work, this serves to create unnoticeable poisoned instances that are also effective for the attack goal. Furthermore, this enables them to study differences between attackers with various levels of imperceptibility concerns simply by tuning the ratio of the discriminator's loss to the classifier's loss when training the generator.

Another work (Shafahi et al. 2018) pioneered what are called feature collision attacks. Essentially, they seek to misclassify a target image, i, in the test set as some target class, c. Their mechanism for doing this is by manipulating instances of c in the training set such that their feature space representation moves closer to that of i. Additionally, they find that overlaying a mostly transparent watermark of i to the poisoned training set images boosts the power and the transferability of these attacks.

Defense and Robustness

Naturally, much research (Sun et al. 2020) has also been done into making models resistant to such attacks. Interestingly, researchers (Weng, Lee, and Wu 2020) have found a tradeoff between adversarial robustness (against evasion attacks) and backdoor robustness (against poisoning attacks). This suggests that deployed models should be careful to consider both threats lest they increase their vulnerability to one when addressing the other.

Designed to mitigate both poisoning and evasion attacks, researchers (Weber et al. 2022) follow previous work in using randomized smoothing during training. Furthermore, they're able to theoretically analyze the robustness bound against poisoning attacks, proving that their defense is effective. Prior work focused on empirical robustness against poisons; research into certified defenses against *poisoning* attacks is crucial and still sparse. Experimentally, they also show the value of their technique on a variety of datasets.

To address backdoor attacks, work has investigated systematically detecting and covering up the trigger (Udeshi et al. 2022). This method is notable for requiring no insight to the model being used, and no modifications to the training process itself. Instead, they simply test inputs to the trained model to identify any backdoor triggers. Then, they cover the trigger image using the dominant color of the original image to ensure similarity. While they provide no theoretical guarantees, empirically, their method outperforms existing work, even when compared to white box methods.

Another approach (Zeng et al. 2022) uses metagradients to "unlearn" the backdoor triggers after a model has been trained. The general approach of unlearning triggers was well-established before this paper, but compared to the existing techniques, this work is able to accomplish the task an order of magnitude more efficiently. Furthermore, unlike other approaches, their process remains effective in the case where access to clean samples is highly limited.

Yet another direction focuses on training directly on poisons to mitigate their potential effect (Geiping et al. 2022). While this approach was well studied to defend against evasion attacks, this work's contribution was to consider it against training time attacks. Furthermore, they find that it generalizes well against multiple threat models (including highly targeted attacks) and is more resource efficient than comparable methods.

In graph learning, one work (Li et al. 2022) observes that existing attacks tend to prefer similar nodes. Based on that observation, they seek to create a "universal" defence against attacks which could be applied to arbitrary nodes on the graph. Essentially, this method removes or adds edges to key nodes that are believed to be potential attack targets. Unlike prior research, their approach is designed (and shown) to work against targeted attacks.

Another work (Xiao, Li, and Su 2021) tries to identify poisoned edges using Jaccard similarity, taking the ones with the lowest score and then removing them from the graph.

To ensure that the graph structure isn't too damaged by this defense, they utilize the minimum connectivity principle as the termination condition for their algorithm. Their experimental results are encouraging, showing effective defenses against poisoning attacks with notably less performance impact than existing methods.

Using random smoothing, researchers (Wang et al. 2021) were able to provide robustness guarantees for any arbitrary graph neural network against both node classification and graph classification tasks. To compute the perturbation size, they formulate finding the optimal random perturbation magnitude as an optimization problem. Solving this problem exactly is unrealistic so they devise an innovative technique based on analyzing regions within the graph. Experimentally, their certified accuracy results on real-world datasets are encouraging.

Conclusion

In this work, we provided an overview of concerns surrounding real-world data. We investigated security games, which model interactions between adversaries. Defenders often rely on attacker's past behavior to build defenses against them, meaning that savvy attackers could manipulate this data nefariously. In the field of data-based decision making, we investigated various applications of this paradigm, discussed the different approaches to find solutions, and mentioned the lack of adversarial research here so far. Through the field of adversarial learning, we explored various approaches for attacks (primarily poisoning attacks) as well as defense techniques. Combining insights from all these fields could allow us to build more robust data-based decision making systems and reduce the threat of attacks to AI applications deployed in the real world.

References

Agrawal, A.; Amos, B.; Barratt, S.; Boyd, S.; Diamond, S.; and Kolter, J. Z. 2019. Differentiable convex optimization layers. *Advances in neural information processing systems*, 32.

Akoglu, L.; Tong, H.; and Koutra, D. 2014. Graph-based Anomaly Detection and Description: A Survey.

Bojchevski, A.; and Günnemann, S. 2018. Deep Gaussian Embedding of Graphs: Unsupervised Inductive Learning via Ranking. In *International Conference on Learning Representations*.

Bojchevski, A.; and Günnemann, S. 2019. Adversarial Attacks on Node Embeddings via Graph Poisoning. arXiv:1809.01093.

Butler, A. R.; Nguyen, T. H.; and Sinha, A. 2021. Countering Attacker Data Manipulation in Security Games. In Bošanský, B.; Gonzalez, C.; Rass, S.; and Sinha, A., eds., *Decision and Game Theory for Security*, 59–79. Cham: Springer International Publishing. ISBN 978-3-030-90370-1.

Chen, J.; Chen, L.; Chen, Y.; Zhao, M.; Yu, S.; Xuan, Q.; and Yang, X. 2019. GA-Based Q-Attack on Community Detection. *IEEE Transactions on Computational Social Systems*, 6(3): 491–503.

Chen, Y.; Nadji, Y.; Kountouras, A.; Monrose, F.; Perdisci, R.; Antonakakis, M.; and Vasiloglou, N. 2017. Practical Attacks Against Graph-based Clustering. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. ACM.

Dong, Y.; Liao, F.; Pang, T.; Su, H.; Zhu, J.; Hu, X.; and Li, J. 2017. Boosting Adversarial Attacks with Momentum.

Donti, P.; Amos, B.; and Kolter, J. Z. 2017. Task-based end-to-end model learning in stochastic optimization. In *Advances in Neural Information Processing Systems*, 5484–5494.

Fang, F.; Nguyen, T. H.; Pickles, R.; Lam, W. Y.; Clements, G. R.; An, B.; Singh, A.; Tambe, M.; Lemieux, A.; et al. 2016. Deploying PAWS: Field Optimization of the Protection Assistant for Wildlife Security. In *AAAI*, volume 16, 3966–3973.

Fang, F.; Stone, P.; and Tambe, M. 2015. When Security Games Go Green: Designing Defender Strategies to Prevent Poaching and Illegal Fishing. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, 2589–2595. AAAI Press. ISBN 9781577357384.

Gan, J.; Guo, Q.; Tran-Thanh, L.; An, B.; and Wooldridge, M. 2019a. Manipulating a Learning Defender and Ways to Counteract. In *NIPS-19*.

Gan, J.; Xu, H.; Guo, Q.; Tran-Thanh, L.; Rabinovich, Z.; and Wooldridge, M. 2019b. Imitative Follower Deception in Stackelberg Games. In *EC '19*.

Geiping, J.; Fowl, L.; Huang, W. R.; Czaja, W.; Taylor, G.; Moeller, M.; and Goldstein, T. 2020. Witches' Brew: Industrial Scale Data Poisoning via Gradient Matching.

Geiping, J.; Fowl, L.; Somepalli, G.; Goldblum, M.; Moeller, M.; and Goldstein, T. 2022. What Doesn't Kill You Makes You Robust(er): How to Adversarially Train against Data Poisoning. arXiv:2102.13624.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2014. Explaining and Harnessing Adversarial Examples.

Grover, A.; and Leskovec, J. 2016. node2vec: Scalable Feature Learning for Networks.

Guo, Q.; An, B.; Bosansky, B.; and Kiekintveld, C. 2017. Comparing strategic secrecy and Stackelberg commitment in security games. In *IJCAI*.

Huang, W. R.; Geiping, J.; Fowl, L.; Taylor, G.; and Goldstein, T. 2020. MetaPoison: Practical General-purpose Clean-label Data Poisoning. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M. F.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 12080–12091. Curran Associates, Inc.

Kar, D.; Ford, B.; Gholami, S.; Fang, F.; Plumptre, A.; Tambe, M.; Driciru, M.; Wanyama, F.; Rwetsiba, A.; and Nsubaga, M. 2017. Cloudy with a Chance of Poaching: Adversary Behavior Modeling and Forecasting with Real-World Poaching Data. In *AAMAS '17*.

Killian, J. A.; Wilder, B.; Sharma, A.; Choudhary, V.; Dilkina, B.; and Tambe, M. 2019. Learning to prescribe interventions for tuberculosis patients using digital adherence

data. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2430–2438.

Kinsey, S. E.; Tuck, W. W.; Sinha, A.; and Nguyen, T. H. 2023. An Exploration of Poisoning Attacks on Data-Based Decision Making. In Fang, F.; Xu, H.; and Hayel, Y., eds., *Decision and Game Theory for Security*, 231–252. Cham: Springer International Publishing. ISBN 978-3-031-26369-9.

Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. arXiv:1609.02907.

Klicpera, J.; Bojchevski, A.; and Günnemann, S. 2018. Personalized Embedding Propagation: Combining Neural Networks on Graphs with Personalized PageRank. *CoRR*, abs/1810.05997.

Krantz, S. G.; and Parks, H. R. 2002. *The implicit function theorem: history, theory, and applications*. Springer Science & Business Media.

Li, J.; Liao, J.; Wu, R.; Chen, L.; Dan, J.; Meng, C.; Zheng, Z.; and Wang, W. 2022. GUARD: Graph Universal Adversarial Defense. arXiv:2204.09803.

Liu, X.; Si, S.; Zhu, X.; Li, Y.; and Hsieh, C.-J. 2019a. A Unified Framework for Data Poisoning Attack to Graph-based Semi-supervised Learning.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019b. RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv:1907.11692.

Mate, A.; Madaan, L.; Taneja, A.; Madhiwalla, N.; Verma, S.; Singh, G.; Hegde, A.; Varakantham, P.; and Tambe, M. 2021. Field Study in Deploying Restless Multi-Armed Bandits: Assisting Non-Profits in Improving Maternal and Child Health. arXiv:2109.08075.

McKelvey, R. D.; and Palfrey, T. R. 1995. Quantal response equilibria for normal form games. In *Games and economic behavior*.

Minervini, P.; Demeester, T.; Rocktäschel, T.; and Riedel, S. 2017. Adversarial Sets for Regularising Neural Link Predictors.

Monti, F.; Boscaini, D.; Masci, J.; Rodola, E.; Svoboda, J.; and Bronstein, M. M. 2017. Geometric Deep Learning on Graphs and Manifolds Using Mixture Model CNNs. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 5425–5434. Los Alamitos, CA, USA: IEEE Computer Society.

Moosavi-Dezfooli, S.-M.; Fawzi, A.; and Frossard, P. 2015. DeepFool: a simple and accurate method to fool deep neural networks.

Mukhopadhyay, A.; and Vorobeychik, Y. 2017. Prioritized allocation of emergency responders based on a continuous-time incident prediction model. In *International Conference on Autonomous Agents and MultiAgent Systems*.

Muñoz-González, L.; Biggio, B.; Demontis, A.; Paudice, A.; Wongrassamee, V.; Lupu, E. C.; and Roli, F. 2017. Towards Poisoning of Deep Learning Algorithms with Back-gradient Optimization. *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*.

Muñoz-González, L.; Pfitzner, B.; Russo, M.; Carnerero-Cano, J.; and Lupu, E. C. 2019. Poisoning Attacks with Generative Adversarial Nets.

Nguyen, T. H.; Butler, A.; and Xu, H. 2020. Tackling Imitative Attacker Deception in Repeated Bayesian Stackelberg Security Games. In *European Conference on Artificial Intelligence*.

Nguyen, T. H.; and Sinha, A. 2022. The Art of Manipulation: Threat of Multi-Step Manipulative Attacks in Security Games.

Nguyen, T. H.; Sinha, A.; Gholami, S.; Plumptre, A.; Joppa, L.; Tambe, M.; Driciru, M.; Wanyama, F.; Rwetsiba, A.; Critchlow, R.; et al. 2016. Capture: A new predictive anti-poaching tool for wildlife protection. In *AAMAS '16*, 767–775.

Nguyen, T. H.; Sinha, A.; and He, H. 2020. Partial Adversarial Behavior Deception in Security Games. In *IJCAI*.

Nguyen, T. H.; Wang, Y.; Sinha, A.; and Wellman, M. P. 2019. Deception in Finitely Repeated Security Games. In *AAAI-19*.

Nguyen, T. H.; Yang, R.; Azaria, A.; Kraus, S.; and Tambe, M. 2013. Analyzing the Effectiveness of Adversary Modeling in Security Games. In *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence*, AAAI'13, 718–724. AAAI Press.

Perozzi, B.; Al-Rfou, R.; and Skiena, S. 2014. DeepWalk. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM.

Perrault, A.; Wilder, B.; Ewing, E.; Mate, A.; Dilkina, B.; and Tambe, M. 2019. Decision-Focused Learning of Adversary Behavior in Security Games. *CoRR*, abs/1903.00958.

Pita, J.; Jain, M.; Marecki, J.; Ordóñez, F.; Portway, C.; Tambe, M.; Western, C.; Paruchuri, P.; and Kraus, S. 2008. Deployed ARMOR protection: The application of a game theoretic model for security at the Los Angeles International Airport. 125–132.

Pita, J.; John, R.; Maheswaran, R.; Tambe, M.; and Kraus, S. 2012. A robust approach to addressing human adversaries in security games. In *ECAI 2012*, 660–665. IOS Press.

Rabinovich, Z.; Jiang, A. X.; Jain, M.; and Xu, H. 2015. Information disclosure as a means to security. In *AAMAS '15*, 645–653.

Rozsa, A.; Rudd, E. M.; and Boult, T. E. 2016. Adversarial Diversity and Hard Positive Generation.

Shafahi, A.; Huang, W. R.; Najibi, M.; Suciu, O.; Studer, C.; Dumitras, T.; and Goldstein, T. 2018. Poison frogs! targeted clean-label poisoning attacks on neural networks. *Advances in neural information processing systems*, 31.

Shieh, E.; An, B.; Yang, R.; Tambe, M.; Baldwin, C.; DiRenzo, J.; Maule, B.; and Meyer, G. 2012. PROTECT: A Deployed Game Theoretic System to Protect the Ports of the United States. In *AAMAS*.

Stewart, G. 1990. Perturbation theory for the singular value decomposition.

Su, J.; Vargas, D. V.; and Sakurai, K. 2019. One Pixel Attack for Fooling Deep Neural Networks. *IEEE Transactions on Evolutionary Computation*, 23(5): 828–841.

Su, Y.; Liu, Q.; Liu, Q.; Huang, Z.; Yin, Y.; Chen, E.; Ding, C.; Wei, S.; and Hu, G. 2018. Exercise-Enhanced Sequential Modeling for Student Performance Prediction. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

Sun, L.; Dou, Y.; Yang, C.; Wang, J.; Yu, P. S.; He, L.; and Li, B. 2020. Adversarial Attack and Defense on Graph Data: A Survey. arXiv:1812.10528.

Sun, M.; Tang, J.; Li, H.; Li, B.; Xiao, C.; Chen, Y.; and Song, D. 2018. Data Poisoning Attack against Unsupervised Node Embedding Methods.

Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; and Fergus, R. 2013. Intriguing properties of neural networks.

Tambe, M. 2011. *Security and game theory: algorithms, deployed systems, lessons learned*. Cambridge University Press.

Tang, J.; Qu, M.; Wang, M.; Zhang, M.; Yan, J.; and Mei, Q. 2015. LINE. In *Proceedings of the 24th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee.

Udeshi, S.; Peng, S.; Woo, G.; Loh, L.; Rawshan, L.; and Chattopadhyay, S. 2022. Model Agnostic Defence Against Backdoor Attacks in Machine Learning. *IEEE Transactions on Reliability*, 71(2): 880–895.

von Stengel, B. 2004. Leadership with Commitment to Mixed Strategies.

Wang, B.; Jia, J.; Cao, X.; and Gong, N. Z. 2021. Certified Robustness of Graph Neural Networks against Adversarial Structural Perturbation. arXiv:2008.10715.

Wang, H.; Xie, H.; Qiu, L.; Yang, Y. R.; Zhang, Y.; and Greenberg, A. 2006. COPE: Traffic engineering in dynamic networks. In *Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications*, 99–110.

Wang, K.; Wilder, B.; Perrault, A.; and Tambe, M. 2020. Automatically learning compact quality-aware surrogates for optimization problems. *Advances in Neural Information Processing Systems*, 33: 9586–9596.

Waniek, M.; Michalak, T. P.; Wooldridge, M. J.; and Rahwan, T. 2018. Hiding individuals and communities in a social network. *Nature Human Behaviour*, 2(2): 139–147.

Weber, M.; Xu, X.; Karlaš, B.; Zhang, C.; and Li, B. 2022. RAB: Provable Robustness Against Backdoor Attacks. arXiv:2003.08904.

Weng, C.-H.; Lee, Y.-T.; and Wu, S.-H. B. 2020. On the Trade-off between Adversarial and Backdoor Robustness. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 11973–11983. Curran Associates, Inc.

Wilder, B.; Dilkina, B.; and Tambe, M. 2018. Melding the Data-Decisions Pipeline: Decision-Focused Learning for Combinatorial Optimization. arXiv:1809.05504.

Wilder, B.; Ewing, E.; Dilkina, B.; and Tambe, M. 2020. End to end learning and optimization on graphs. arXiv:1905.13732.

Wilder, B.; Yadav, A.; Immorlica, N.; Rice, E.; and Tambe, M. 2017. Uncharted but not Uninfluenced: Influence Maximization with an uncertain network. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, 1305–1313.

Xi, Z.; Pang, R.; Ji, S.; and Wang, T. 2020. Graph Backdoor.

Xiao, Y.; Li, J.; and Su, W. 2021. A Lightweight Metric Defence Strategy for Graph Neural Networks Against Poisoning Attacks. In Gao, D.; Li, Q.; Guan, X.; and Liao, X., eds., *Information and Communications Security*, 55–72. Cham: Springer International Publishing. ISBN 978-3-030-88052-1.

Xu, L.; Gholami, S.; McCarthy, S.; Dilkina, B.; Plumptre, A.; Tambe, M.; Singh, R.; Nsubuga, M.; Mabonga, J.; Driciru, M.; et al. 2020. Stay ahead of Poachers: Illegal wildlife poaching prediction and patrol planning under uncertainty with field test evaluations (Short Version). In *2020 IEEE 36th International Conference on Data Engineering (ICDE)*, 1898–1901. IEEE.

Xue, Y.; Davies, I.; Fink, D.; Wood, C.; and Gomes, C. P. 2016. Avicaching: A two stage game for bias reduction in citizen science. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*, 776–785.

Yadav, A.; Chan, H.; Jiang, A. X.; Xu, H.; Rice, E.; and Tambe, M. 2016. Using Social Networks to Aid Homeless Shelters: Dynamic Influence Maximization under Uncertainty. In *AAMAS*, volume 16, 740–748.

Yadav, A.; Wilder, B.; Rice, E.; Petering, R.; Craddock, J.; Yoshioka-Maxwell, A.; Hemler, M.; Onasch-Vera, L.; Tambe, M.; and Woo, D. 2018. Bridging the gap between theory and practice in influence maximization: Raising awareness about HIV among homeless youth. In *IJCAI*, 5399–5403.

Yan, B.; and Gregory, S. 2012. Detecting community structure in networks using edge prediction methods. *Journal of Statistical Mechanics: Theory and Experiment*, 2012(09): P09008.

Yang, C.; Wu, Q.; Li, H.; and Chen, Y. 2017. Generative Poisoning Attack Method Against Neural Networks.

Yang, R.; Kiekintveld, C.; Ordonez, F.; Tambe, M.; and John, R. 2011. Improving resource allocation strategy against human adversaries in security games. In *IJCAI*.

Yuan, X.; He, P.; Zhu, Q.; and Li, X. 2017. Adversarial Examples: Attacks and Defenses for Deep Learning.

Zeng, Y.; Chen, S.; Park, W.; Mao, Z. M.; Jin, M.; and Jia, R. 2022. Adversarial Unlearning of Backdoors via Implicit Hypergradient. arXiv:2110.03735.

Zhang, J.; Wang, Y.; and Zhuang, J. 2021. Modeling multi-target defender-attacker games with quantal response attack strategies. *Reliability Engineering & System Safety*, 205.

Zhang, M.; Hu, L.; Shi, C.; and Wang, X. 2020a. Adversarial Label-Flipping Attack and Defense for Graph Neural Networks. 791–800.

Zhang, W.; Guo, H.; Ranganathan, P.; Patel, J.; Rajasekharan, S.; Danayak, N.; Gupta, M.; and Yadav, A. 2023. A Continual Pre-training Approach to Tele-Triaging Pregnant Women in Kenya.

Zhang, Z.; Jia, J.; Wang, B.; and Gong, N. Z. 2020b. Backdoor Attacks to Graph Neural Networks.

Zheng, H.; Xiong, H.; Chen, J.; Ma, H.; and Huang, G. 2022. Motif-Backdoor: Rethinking the Backdoor Attack on Graph Neural Networks via Motifs.

Zhuang, J.; Bier, V. M.; and Alagoz, O. 2010. Modeling secrecy and Deception in a multi-period attacker-defender signaling game. *European Journal of Operational Research*, 203: 409–418.

Zügner, D.; Akbarnejad, A.; and Günnemann, S. 2018. Adversarial Attacks on Neural Networks for Graph Data. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '18, 2847–2856. New York, NY, USA: Association for Computing Machinery. ISBN 9781450355520.

Zügner, D.; and Günnemann, S. 2019. Adversarial Attacks on Graph Neural Networks via Meta Learning. In *International Conference on Learning Representations (ICLR)*.