# Learning Adversarial Interactions in Stackelberg Security Games with Limited Data

**Boli Fang, Miao Jiang**
Department of Computer Science
Indiana University
`bfang, miajiang@iu.edu`

**Alexey Tregubov, James Blythe, Emilio Ferrara**
Information Sciences Institute
University of Southern California
`tregubov, blythe, ferrarae@isi.edu`

## Abstract

Data driven security games have been instrumental in modeling the attacker-defender interaction within multi-agent green security games, with applications ranging from illegal fishing to drug interdiction. Despite the high level of success in modeling behavior of different parties, training security games require a non-trivial amount of hard-to-obtain real-time security game data. To that end, we develop an adaptive data augmentation (DA) strategy based on PAGANDA, a recently developed DA framework making use of generative adversarial networks for maximal information extraction from limited data. We further demonstrate by experiments that such data augmentation method indeed improves the performance of adversary interaction learning models driven by scarce Stackelberg Security Game data.

## 1 Introduction

Many real-world scenarios require for intelligent decision making against attacking adversaries in complex security problems. Towards these goals, Stackelberg Security Game (SSG) is a highly useful model in maximizing defender utilities from target attacks by intelligent adversaries, particularly in the cases where the defender has limited resources in protecting the designated set of targets and where the attackers possess only partial information about the targets and defenders. Much previous work in SSG assumes that the adversary utility function is not given, and that one could learn the adversary behavioral patterns through history. Prior work on such direction includes 2-stage ML approaches [11] and decision-focused learning approaches[12].

However, one notable drawback for these learning algorithms is the need for non-trivial amounts of reliable real-time data, which is often hard to acquire from game settings. Converting raw data (e.g. in GIS) into usable representations of attacker-defender games, as suggested by recent endeavors [11], often requires high manual labor. Motivated by these obstacles, we present in this paper a game-theoretic data augmentation (DA) strategy based on PAGANDA, a Generative Adversarial Network recently developed in [6]. Our contributions can be summarized as follows:

- We propose a data augmentation augmentation strategy based on PAGANDA [6] under game theoretic settings. By our best knowledge, our DA method is the first of its kind.
- We illustrate how the key components in the generative model would improve the quality of games generated.
- We experiment on real illegal fisheries data to showcase the effectiveness of GAN-based data augmentation in data-driven game-theoretic contexts.

Abstract for: Smooth Games Optimization and Machine Learning Workshop (NeurIPS 2019), Vancouver, Canada.

## 2 Related work

Previous research on security games has resulted in a substantial number of literature. [9, 17] have focused on games where payoffs are fully observable and all player behavior can be modeled with bounded rationality. For Stackelberg Security games in particular, *quantal response*(QR) and its model variations [19, 11] have been demonstrated to improve attack predictions. [11, 12] seek to improve upon previous results by building ML-based models to learn adversarial behavior. On the other hand, traditional and ML-based DA strategies [10, 14, 4, 15] have found their way in many deep image classification tasks. Recent work [1] makes use of GANs for data augmentation, but their choices of objective functions and structures still calls for further improvement.

## 3 Stackelberg Security Games and Data-driven decision learning

In game theory, a Stackelberg Security Game (SSG) is defined as a game with two players, a leader and a follower. A leader chooses a strategy, and a follower optimizes its reward from the strategy [16]. When a selected target is not protected by the defender during the game, the attacker receives a reward $u_a \geq 0$ and the defender receives a penalty $u_d \leq 0$. Each target has its associated $u_a$ and $u_d$. The goal of Stackelberg Security Games (SSG) is to maximize the defenders' expected utility (DEU) for the game, which is defined as follows [12]:

$$DEU(p; q) = \sum_{i \in T} (1 - p_i) q_i(u_a, p) u_d(i), \tag{1}$$

where $p$ is the probability of a defender defending targets(defender values) and $q$ is the attacker's attack probability over targets(attacker values). Following recent developments in SSG adversary modeling, we use in this paper the SUQR attacker function to model $q$ [11], where

$$q_i(p, y) \propto \exp\left(w p_i + \phi(y_i)\right). \tag{2}$$

Here $w$ is a constant, $y_i$ is a feature vector for target $i$, and $\phi$ is a target value function.

Notice it is possible to infer the adversarial behavior function $q$ from historical attacks. Previous work centers on two approaches: a two-stage approach, where attacker's behavior is learned through a machine learning model on historical attack data; and decision-focused approach, where attacker's behavior is adaptively learned using DEU values [12]. Figure 1 and 2 [12] illustrate the framework of the two approaches respectively.

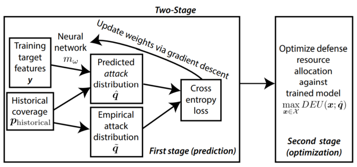
Two-Stage
Update weights via gradient descent
Training target features $\boldsymbol{y}$
Neural network $m_\omega$
Predicted *attack* distribution $\hat{\boldsymbol{q}}$
Historical coverage $\boldsymbol{p}_{\text{historical}}$
Empirical attack distribution $\tilde{\boldsymbol{q}}$
Cross entropy loss
First stage (prediction)
Optimize defense resource allocation against trained model $\max_{\boldsymbol{x} \in \mathcal{X}} DEU(\boldsymbol{x}; \hat{\boldsymbol{q}})$
Second stage (optimization)

Figure 1: two stage learning[12]

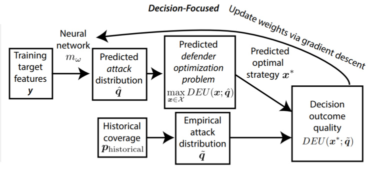
Decision-Focused
Update weights via gradient descent
Training target features $\boldsymbol{y}$
Neural network $m_\omega$
Predicted *attack* distribution $\hat{\boldsymbol{q}}$
Predicted *defender optimization problem* $\max_{\boldsymbol{x} \in \mathcal{X}} DEU(\boldsymbol{x}; \hat{\boldsymbol{q}})$
Predicted optimal strategy $\boldsymbol{x}^*$
Historical coverage $\boldsymbol{p}_{\text{historical}}$
Empirical attack distribution $\tilde{\boldsymbol{q}}$
Decision outcome quality $DEU(\boldsymbol{x}^*; \tilde{\boldsymbol{q}})$

Figure 2: Decision Focused learning[12]

However, the models performs poorly if there is insufficient real-time training data, which is generally difficult to obtain due to data access restrictions and labor-intensive pre-processing. Recent research on SSG [12] relies largely on synthetic datasets since the amount of real data is limited. The issue poses great challenges for real-time applications for these models.

## 4 PAGANDA: Data Augmentation with parallel adaptive GANs

Given the difficulty in obtaining sufficient data for decision-focused learning models, we turn our attention to data augmentation strategies. As demonstrated by theoretical justification [13, 2, 8] and experiments on various types of data [1, 5, 7], Generative Adversarial Network(GAN)s have the capacity to boost ML model performances. In this section we describe Parallel Adaptive Generative Adversarial Network Data Augmentation (PAGANDA) [6] in the context of SSGs. As in [6], our method consists of three interrelated components: generative data augmentation, parallel data addition with fold division, and adaptive weight adjustment for data augmentation, with modifications that take SSG properties into consideration.

### 4.1 Generative Data Augmentation

The first part of our method involves *generative data augmentation*, which constructs varied data points by repeatedly generating samples to be added to training set using GANs. We start off with a limited training set consisting of 3-tuples in the form of (features,defender_values,attacker_values), and consecutively. After running a fixed number $t$ of training epochs, we proceed to the augmentation epoch, during which we extract a fixed number of sample data points from the generator $G$ as produced by the interaction between generator and discriminator. Figure 3 is a flow-chart of our procedure viewed from one of the generative adversarial networks involved.

### 4.2 Parallel Data Generation with Fold Division

The second part of our method consists of a parallel data generation strategy, inspired by $K$-fold cross validation [3]. Dividing the training tuple data into $K$ folds at the beginning, we run in parallel $K$ independent generators $\{G_i\}_{i=1}^{K}$. Each generator $G_i$ is trained on one of data groups, and each data group $i$ consists of $K-1$ folds of the training set, except for the $i$-th fold.To allow for maximal usage of each generated data, we insert the data in a way such that the tuples generated by one generator $G_i$ are sent to the training data groups corresponding to all other $K-1$ generators except for that corresponding to $G_i$. This is to prevent overfitting and bolster the robustness of our strategy. Figure 4 demonstrates the process.

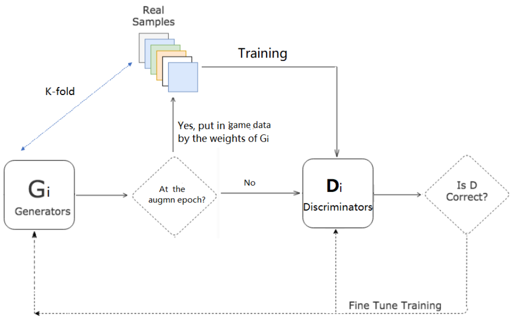

Figure 3: GAN flowchart

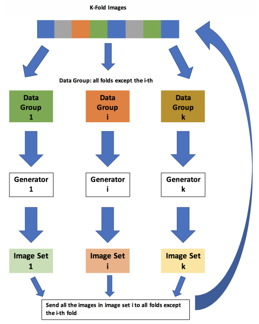
K-Fold Images
Data Group: all folds except the i-th
Data
Group
1
Data
Group
i
Data
Group
k
Generator
1
Generator
i
Generator
k
Image Set
1
Image Set
i
Image Set
k
Send all the images in image set i to all folds except the i-th fold

Figure 4: Parallel Data Generation

### 4.3 Adaptive Generator Weighting

Furthermore, to determine which generators are the most effective in generating authentic images, we introduce *adaptive generator weighting* at each augmentation epoch. At the initial stage, all the generators are treated equally. As training goes on, the generators that produce better images will contribute more to new generated batches at each augmentation epoch.

To evaluate the quality of such a batch of sample data, we compute the DEU values in the initial unaugmented game set and those in the generated game tuples to obtain the DEU vector $m$ of the initial dataset and the DEU vector $n$ of the augmented batch. We then consider the Jensen-Shannon divergence(JSD) between the two vectors $m, n$ for game similarity comparison:

$$JSD(m,n) = \frac{1}{2}KL(m\|\frac{m+n}{2}) + \frac{1}{2}KL(n\|\frac{m+n}{2}). \tag{3}$$

Notice that given a generator described in section 4.1, the lower the value of JSD, the closer the proximity between generated games and original games. Before the batch of sample tuples generated by one generator $G_i$ are sent to the data group corresponding to the other $K-1$ generators at each epoch, we collect the JSDs of each generator. From the pre-computed *JSD*'s, we define *the generator weight* $p_i$ of a generator $G_i$ as

$$p_i = \frac{\exp(-JSD_i)}{\sum_{j=1}^{K}\exp(-JSD_j)},$$

and determine how many tuples should be sampled from generator $G_i$ and sent to other data groups for subsequent training in the very next augmentation epoch.

# 5 Experiments

To demonstrate the effectiveness of our data augmentation method, we conduct experiments on real data obtained from records of illegal fishing. Our records are extracted from the AIS illegal fishing data from the Global Fishing Watch Project [18] and `www.marinetraffic.com`, consisting of traces of vessels which account for illegal fishing in different parts of the ocean.

We then use the decision-focused and two-stage models as described in [12] as our bases for learning adversarial behavior in Stackelberg security games. Overall, our method is able to substantially improve the defender expected utility (DEU) values of testing games in both the 2-stage learning and the decision-focused learning.

## 5.1 Model and Parameters

We follow the settings in [12]. To facilitate data augmentation, we reformat the features in each training game tuples into matrices, and concatenate attacker and defender vectors with the feature matrix to produce a game matrix for each 3-tuple. These game matrices are in turn fed into the PAGANDA framework, which in turn produces additional game matrices from the input.

The GANs used in our context are the state of the art WGAN-GP[8]. We start off with 128 game tuples, and randomly choose 32/48/64 'seed' tuples. We then augment this subset with PAGANDA to approximate data limitation, train SSG models with these games, and use testing games for DEU evaluation. For game-theoretic parameters, we refer to [11], with slight variations, as listed below.

- *Number of targets:* 20;
- *Number of features per target:* 8;
- *Number of training games:* 128;
- *Number of attacks per training game:* 5;
- *Number of Generators in PAGANDA:* 4;
- *Number of testing games:* 32.
- *Attacker weight $w$ on defender coverage for the SUQR model*: -4;
- *Historical Coverage Probability:* We assume the defender computes DEU thinking that all attacker values are equal.

## 5.2 Results

We evaluate the quality of generated games by computing the average DEU values of the 'seed' games and those of generated games. Higher DEU values on 2-stage and end-to-end settings after PAGANDA, as listed in table 1 and 2, suggest that PAGANDA significantly improves the performances of learning models under data scarcity.

| No. of Seeds | 32 | 48 | 64 | 128 |
|---|---|---|---|---|
| initial DEU | -2.8923 | -2.6396 | -2.4242 | -2.1404 |
| final DEU | -1.4842 | -1.4396 | -1.3672 | -2.1404 |

Table 1: 2-stage DEU with Augmentation

| No. of Seeds | 32 | 48 | 64 | 128 |
|---|---|---|---|---|
| initial DEU | -1.12 | -5.25e-14 | -1.74e-14 | -0.0015 |
| final DEU | -4.23e-15 | -3.77e-15 | -2.45e-15 | -0.0015 |

Table 2: end-to-end DEU with Augmentation

# 6 Conclusion and future work

In this paper we present a novel data augmentation strategy based on GANs to help game-theoretic learning models adapt to data scarcity. We consider two corresponding learning algorithms to learn optimal defense decisions from historical attacker behavior, and show by experiments on real-time fisheries data that data augmentation indeed helps with learning adversarial behavior under both 2-stage and end-to-end settings. As a further step, we plan to extend our data augmentation framework to other SSG models with more general assumptions such as non-convex objective functions.

## 7 Acknowledgment

This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Agreement No. HR00111990017.

## References

[1] Antreas Antoniou, Amos Storkey, and Harrison Edwards. Data augmentation generative adversarial networks. *arXiv preprint arXiv:1711.04340*, 2017.

[2] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, Proceedings of Machine Learning Research. PMLR, 2017.

[3] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[4] Terrance DeVries and Graham W Taylor. Dataset augmentation in feature space. *arXiv preprint arXiv:1702.05538*, 2017.

[5] Fabio Henrique Kiyoiti dos Santos Tanaka and Claus Aranha. Data augmentation using gans. 2019.

[6] Boli Fang, Miao Jiang, and Jerry Shen. Paganda: An adaptive task-independent automatic data augmentation. 2019.

[7] Emilio Ferrara. The history of digital spam. *Communications of the ACM*, 62(8):82–91, 2019.

[8] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems 30*. 2017.

[9] Jason S Hartford, James R Wright, and Kevin Leyton-Brown. Deep learning for predicting human strategic behavior. In *Advances in Neural Information Processing Systems 29*. 2016.

[10] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*. 2012.

[11] Thanh Hong Nguyen, Rong Yang, Amos Azaria, Sarit Kraus, and Milind Tambe. Analyzing the effectiveness of adversary modeling in security games. In *Twenty-Seventh AAAI Conference on Artificial Intelligence*, 2013.

[12] Andrew Perrault, Bryan Wilder, Eric Ewing, Aditya Mate, Bistra Dilkina, and Milind Tambe. Decision-focused learning of adversary behavior in security games. *arXiv preprint arXiv:1903.00958*, 2019.

[13] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *CoRR*, abs/1511.06434, 2015.

[14] Patrice Y. Simard, David Steinkraus, and John C. Platt. Best practices for convolutional neural networks applied to visual document analysis. 2003.

[15] Choon H Teo, Amir Globerson, Sam T Roweis, and Alex J Smola. Convex learning with invariances. In *Advances in neural information processing systems*, pages 1489–1496, 2008.

[16] Bernhard Von Stengel and Shmuel Zamir. Leadership with commitment to mixed strategies. Technical report, Citeseer, 2004.

[17] James R. Wright and Leyton-Brown Kevin. Predicting human behavior in unrepeated, simultaneous-move games. *Games and Economic Behavior*, 106, 2017.

[18] www.globalfishingwatch.org. *Global Fishing Watch Project*, 2016 (accessed 2019).

[19] Rong Yang, Christopher Kiekintveld, Fernando Ordonez, Milind Tambe, and Richard John. Improving resource allocation strategy against human adversaries in security games. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume One*, IJCAI'11, 2011.