Detecting Fraud in Adversarial Environments: A Reinforcement Learning Approach

Adrian Mead, Tyler Lewris, Sai Prasanth, Stephen Adams, Peter Alonzi, Peter Beling
University of Virginia, atm4rf, tal3fj, lbs7aa, sca2c, lpa2a, pb3a@virginia.edu

Abstract - **Credit card fraud is a costly problem for banks and a major frustration for consumers. As such, static models to detect fraud that rely on supervised training are exposed to the risk of being learned and circumvented. Previous adversarial learning work in fraud prevention showed increased effectiveness over static models that did not account for changing fraudster behavior. We extend this work by utilizing Reinforcement Learning and framing the fraudster and card issuer interaction as a Markov Decision Process (MDP) and performing prediction and control. Our MDP takes on the perspective of an agent (in this case the fraudster with a stolen credit card) who interacts with an environment (merchants and a fraud classifier), by taking actions (transactions), and receiving rewards (relating to whether the transactions were successful/declined). This approach allows us to simulate fraudulent episodes in such a way that techniques like model-free policy iteration can identify an optimal policy for the fraudster. The episode ends when the card is terminated by the credit card company for fraud. We found that, compared to a static classifier, making small changes to our fraud classifier on a regular basis led to a significant decrease in the ability of a fraud agent to learn an optimal policy.**

Index Terms – Adversarial Learning, Consumer Credit Fraud Detection, Markov Decision Process, Monte Carlo Policy Control, Reinforcement Learning

INTRODUCTION

In 2016, credit card fraud resulted in losses of $22 billion globally [1]. To prevent such losses, present-day fraud models can be simplified to a classification problem where the outcome is a probability that any given transaction is classified as fraud. Credit card companies must walk a fine line between minimizing the number of instances where a customer's card gets declined (false positive) and allowing fraudulent transactions (false negatives). Hence, credit card firms need to strike a balance between the volume of flagged transactions and losses due to fraud, especially as consumers expect safety and surety from their credit card company.

The normal fraud detection paradigm is that of a static model that acts as a filter for these fraud and non-fraud transactions. However, given the costly nature of credit card fraud and the ease with which stolen credit card data can be obtained from the dark web, reality paints a picture of a highly contentious interaction between fraudsters and the financial institutions. Fraudsters attempt to defeat the fraud detection models and are constantly adapting their strategies to maximize stolen dollars. As fraudsters learn the classifier and chip away at its effectiveness, data scientists at credit card companies spend large amounts of time and money to combat their efforts.

We advocate moving away from the static model paradigm into a different perspective whereby modeling will take into account the behavior of the adversary. The competitive advantage of modeling adversary behavior lies with the financial institution being able to learn and adapt to changing fraud strategies and react accordingly.

Thus far, adversarial approaches have been primarily applied to game theory, a popular modeling method commonly used in economics and psychology focusing on groups of entities interacting with one another. We present a novel framework for studying credit card fraud that includes an MDP model of fraudster decisions and an active defense by the lender. This approach simulates the manner in which a fraudster learns how to beat a classifier by implementing states, actions, and rewards that resemble the risks and payoffs that fraudsters actually experience. We also disrupt the adversary's learning process, altering classifier probability classification thresholds at regular intervals to effectively limit the amount of successful fraud.

RELATED WORKS

Adversarial learning is a subfield of machine learning that focuses on the ability of an actor to generate high-volume, low-cost observations to prod the capabilities of a classifier. Given the abundance of data and its industry applications, a decent amount of the work in adversarial learning has been in the area of spam detection [2][3]. Nelson et al demonstrated that an adversary with access to even 1% of the training set could adapt their fraud creation and effectively render a working classifier ineffective [2]. Dalvi et al implemented a game theoretic framework that had a classifier take the adversary's best actions into account, greatly increasing their model's predictive power [3].

Credit card fraud detection has a number of particularly important similarities to spam detection including large class imbalances, a similar amount of high-volume observations that can be generated at low cost, and the burden that comes in the domain of computer security:

malicious actors that want to fool the classifier. Recent work from Zeager et al. has applied the game theoretic approach to credit card fraud detection and found that adversary-trained classifiers with retrainings performed better than static classifiers [4]. Much of their work builds upon literature relating to spam filters, in particular work by Liu and Chawla that rigorously modeled the adversarial problem as a non-cooperative Stackelberg game [5].

The literature is sparser when it comes to the application of reinforcement learning for problems in adversarial learning. Much of the work today is focused on modelling an adversary that works to prevent a reinforcement learning agent from successfully training to an optimal policy (training-focused approach). This is slightly different from our work, in which the reinforcement learning agent itself is going to be an adversary to a classifier and the training process is unmolested (classifier-focused approach). Pinto et al use the training-focused approach to generate a reinforcement learning agent that performs better in testing than agents that had been trained without the aid of an adversary [6]. More in line with our work, Shen et al framed the problem of adversaries attacking a cyber security network into a MDP [7]. They found that the MDP improved their modeling of adversary actions.

DATA

The data set used in this paper was provided by a major financial institution. The data set contains nearly 86 million observations which are anonymized credit card transactions. There are 69 attributes within the data set, one of which is the response variable, a fraud indicator labeled as either "Y" or "N". The other attributes range from transaction level data such as transaction amount, merchant category code, distance between the transaction and home, and account level data such as account balance, account opening date, days since phone number change on account, and other account related features. The dataset was large enough that we built a set of 25 randomly-sampled sets of the transaction history of 3,000 credit card accounts. On average, each of these sets had 100,000 transactions and all of these sets contained some fraud.

One of the immediate issues with our data was a significant class imbalance. In our dataset, fraud makes up ~0.1% of all credit card transactions. This means that we must sample intelligently so as to train our models in such a way that fraud is not completely swamped. To this end, we downsampled the non-fraud transactions to achieve an ~15% fraud representation during our model-building step. We performed downsampling as opposed to upsampling because of the relative softening of demand on computing power and also because of the perceived homogeneity in the accounts that did not experience fraud.

METHODS

Our analysis is rooted primarily in reinforcement learning. In contrast to supervised and unsupervised methods, the reinforcement learning approach has no labels or clusters, but rather simulates episodes that produce strings of rewards which are used as signal for updating the model. The overall goal of the agent is to maximize their cumulative expected reward. Figure 1 demonstrates the general algorithm that is being followed. An agent (the fraudster), is in a state. It interacts with the environment (the fraud classifier), by taking an action (entering a transaction). The environment processes the action and returns some feedback to the agent in the form of a new state (the new status of the stolen credit card), and a reward (good or bad depending on if the transaction was accepted or declined). Policy control is a process that can be implemented to teach the fraudster the best transactions to take in any given state such that they maximize their cumulative expected reward.

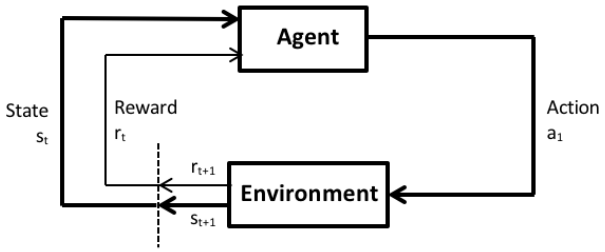

FIGURE I
REINFORCEMENT LEARNING WITH MDPS

Credit card fraud lends itself well to the reinforcement learning paradigm. In our experiments, the fraudster (agent) is trying to determine the best set of transactions (actions) to steal as much money as possible by beating the bank's fraud classifier (environment). This novel approach allows us to simulate the learning process for an adversary in an environment meant to mimic the real world and closely aligned with actual fraudster incentives.

The above scenario can be expressed more compactly as an MDP. MDPs are composed of a set of states, a set of actions, a set of rewards, a probability transition function, and a discount factor. In our formulation, these are as follows:

- States: a 2-tuple representing the [number of low $-amount transactions, number of high $-amount transactions] taken on a card. This is limited to a max of 5 total transactions.
- Actions: either to take a low $-amount transaction or a high $-amount transaction
- Rewards:
  - -50 for the first transaction taken on the card (representing the cost of stealing the card)
  - +5 for a successful low $-amount transaction
  - +50 for a successful high $-amount transaction
  - 0 when the transaction is declined and the card is shut off by the credit card company
- Discount Factor: 1
- Probability Transition Function: not explicitly defined; implicitly represented with the fraud classifier

We used +5 as the reward for a successful low $-amount transaction as it represents the 25th percentile of transaction amount in our fraud dataset. The +50 reward comes from the 75th percentile. The discount factor of 1 is common when dealing with a finite horizon problem. We limit the total number of transactions on the card to 5 before being declined in order to limit the size of the state space. This means that if the number of low $-amount transactions + the number of high $-amount transactions = 5, the next transaction will be declined regardless of the classifier. This is another way of implicitly saying that all fraud will be caught after 5 transactions (which is true for 84% of fraud-compromised accounts in our dataset).

We built a logistic regression model to act as the fraud classifier environment designed to interpret state and action information coming from the agent. This fraud classifier is fairly simple with just 3 predictors: # of low $-amount transactions, # of high $-amount transactions, and whether the action is a low or high $-amount. We used $15 as the cutoff for a low/high $-amount transaction based on median transaction amount across all fraudulent transactions in the dataset. We trained our model using data from one of the 25 random samples across accounts by randomly selecting an account, and then randomly selecting a sequence of five contiguous transactions from that account. We achieved the undersampling of non-fraud by only sampling from accounts that contained fraud.

An episode of our reinforcement learning algorithm represents the complete case starting from when an adversary purchases the information of a compromised credit card (with the state [0, 0]), followed by the adversary making their first transaction and receiving their first reward corresponding to an accept/decline from the classifier. This process repeats from the new state until the credit card reaches the decline state.

Finding the best action to take from any given state is a process called policy control. Our work implemented the Monte Carlo policy control algorithm in Python. Broadly speaking, Monte Carlo is used to describe any method of estimation with significant randomness. In policy control, the basic idea is that the model learns from experience, using sequences of states, actions, and rewards generated by simulating your MDP across many episodes. As specific state-action pairs are observed over the course of many episodes, a running average of their total reward is recorded and updates are made each time that state and action is visited. Equations (1) and (2) correspond to this process. Once enough episodes have been compiled, the actual policy is extracted by following the action from each state that corresponds to the largest Q state-action value. This is shown in (3). It can be shown that as the number of sampled episodes approaches infinity, the policy converges on optimality. In our experiments, we ran 200 episodes of the credit card which was sufficient for finding the optimal policy.

$$N(S_t, A_t) = N(S_t, A_t) + 1 \quad (1)$$

$$Q(S_t, A_t) = Q(S_t, A_t) + \frac{1}{N(S_t, A_t)} * [G_t - Q(S_t, A_t)] \quad (2)$$

$$\pi(S_t) = \arg\max_{a \in A_t} Q(S_t, a) \quad (3)$$

One of the principle concerns in our work was with how quickly the agent finds the optimal policy. That is, how many episodes (stolen credit cards) does it take for an adversary to converge on the best policy. To test this we took our logistic regression fraud classifier and varied the classification threshold (the value at which the model declares a transaction either fraud or not fraud), between 0 and 1 and calculated the total value associated with the optimal policy (how much money the fraudster could steal). Armed with this knowledge we found the region of classification thresholds that were sensitive to precision and recall (that is, not all false positives or all false negatives). We then went back and trained the agent again in ten separate trials, this time varying the classification thresholds randomly at regular intervals during the training process and evaluating the performance of the fraudster after each episode. This allowed us to see the relative effectiveness between different rates of classification threshold change. We tried a variety of different rates, changing the classification threshold every 1, 2, 4, 8, 16, 32, 64, and 200 episodes.

RESULTS

Our experimental classifier was trained on the credit card dataset and tested on a held-out dataset randomly sampled from the original dataset. Figure II shows the effectiveness of our classifier in a Receiver Operating Characteristics (ROC) curve. Despite only using three predictive features when the original space had 68, the model still has reasonable predictive power with an AUC of 0.64.

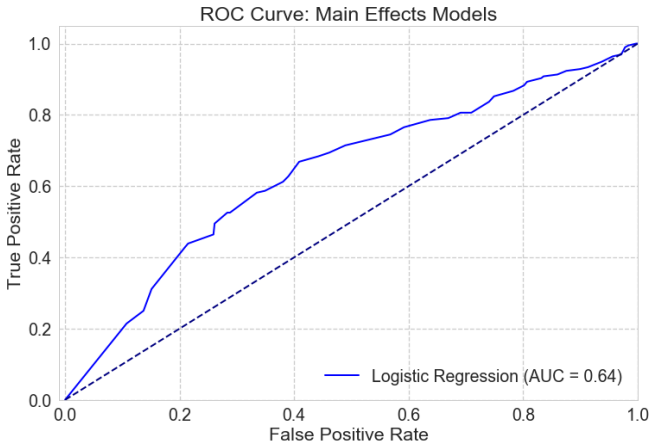

FIGURE II
AN ROC CURVE DEMONSTRATING THE EFFECTIVENESS OF OUR FRAUD CLASSIFIER

Then our classifier was used to train the fraud agent through interaction with the environment at different classification thresholds. Once the optimal policy was reached, the total value of the policy was recorded, as was

the number of episodes it took to converge on the optimal policy. Figure III shows the total value in dollars accrued by following the optimal policy at different classification thresholds. When the classification threshold is sufficiently low, then all transactions are counted as fraud and any transactions the fraudster puts on the card are promptly declined (as are all real transactions, it should be noted). When the classification threshold is sufficiently high, all transactions are successful and it is in the fraudster's best interest to make as many large transactions as possible. In between there is a small region where the behavior is slightly more nuanced. It is in this small region that the actual fraud classifier would operate, and these are the classification thresholds we used when varying our classifier at regular intervals to confuse the fraudster.

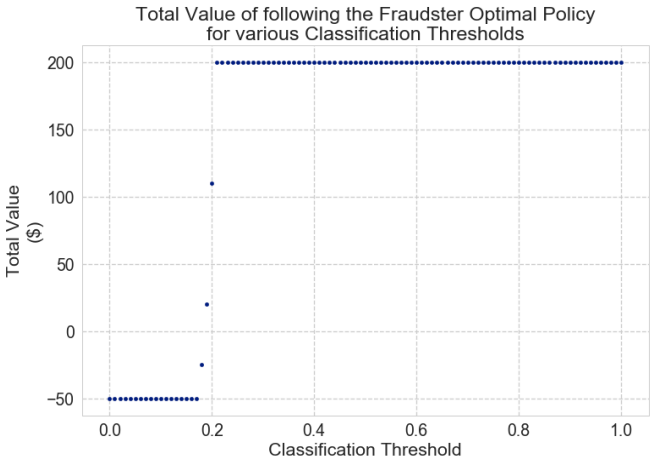

FIGURE III

THE DOLLARS ACCRUED BY FOLLOWING THE OPTIMAL POLICY LEARNED AT DIFFERENT MODEL CLASSIFICATION THRESHOLDS

Figure IV shows our efforts at trying to fool the fraudster by regularly varying the classification threshold. The curve shows the amount of money that the fraudster would be able to steal following their best understanding of the classifier at that time. The dashed vertical lines represent the episodes in which the classification threshold was changed. The horizontal dashed lines represent the maximum possible dollar amount that the fraudster could have successfully stolen by following the optimal policy for a given classification threshold. So the difference between the curve and the max dollar value at that classification threshold is the opportunity that was missed by the fraudster. We can see that in this case, the fraudster struggles to reach the max possible value for any specific classification threshold. It's briefly able to achieve the optimal behavior at around episode 100 for classification threshold 0.2, but then the classification threshold changes and it loses its edge. Then at the end with episodes 180 onwards, we see that the fraudster has found the best policy for classification threshold 0.19.

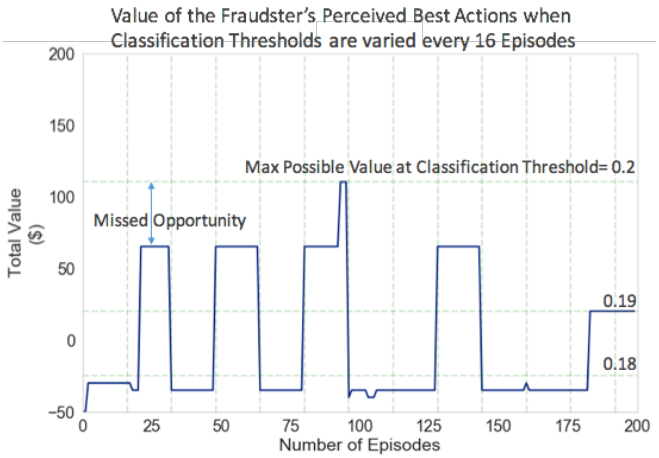

FIGURE IV

AN EXAMPLE PLOT OF THE FRAUDSTER ATTEMPTING TO LEARN TO BEAT THE CLASSIFIER ACROSS 200 EPISODES

We gathered our results at a high level in Figure V, which shows the performance across ten trials of the 200-episode fraudster training process at a variety of different classification threshold rates of change. This graph displays the percent of opportunity that the fraudster was able to capture at the end of each episode where -$50 is a baseline as that is the worst that the fraudster could do. For example, if in a single episode the fraudster could have possibly made $125 following the best series of actions for a particular classification threshold but only made $50, then that would result in a captured value of 0.57. Starting from the right, it appears that when the classification threshold is never changed (the static 200 episode case), the fraudster learns the best policy fairly quickly and then never needs to deviate from it. However, once we vary the classification threshold every 64 episodes, the median moves down and the fraudster captures less of the possible value. This trend continues into the 32 and 16 episodes cases. This is in line with our hypothesis that models that are varied less often are much more at risk of being exploited and learned by a dedicated adversary. Particularly interesting though is the general plateauing of effectiveness once we reach classification threshold changes every 16 episodes or less. That is, it seems that a fraudster which is dealing with the classifier being changed every episode seems to be about as successful as a fraudster dealing with the classifier being changed every 2, 4, 8 or 16 episodes, implying a diminishing return for changing the classifier too often.

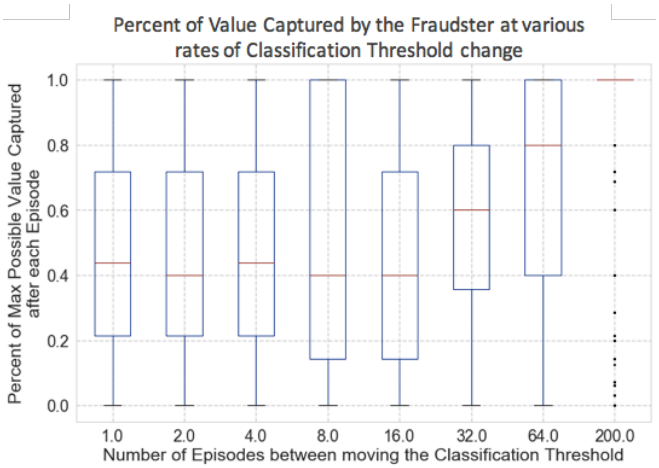

FIGURE V

ACROSS TEN TRIALS, THE PERFORMANCE OF THE ADVERSARY ON A PER-EPISODE BASIS AS A PROPORTION OF THEIR ABILITY TO CAPTURE THE MAX DOLLAR VALUE POSSIBLE

CONCLUSIONS

We successfully translated the adversarial interaction between a fraudster and credit card company into an MDP. Although the state, action, and reward spaces are fairly simple, we were able to achieve predictive power in our classifier and train a fraud agent to exploit the weaknesses in the classifier. This represents a significant change from the typical assumptions-heavy game-theoretic implementations towards a more flexible framework that can simulate over many more scenarios.

Our research also found a strong relationship between making small, regular changes to a classifier and an associated decrease in the ability of a fraud agent to learn the best policy for maximizing successful fraudulent transactions. We found that the effectiveness of this technique is somewhat limited and that changing the classifier too often does not significantly improve our method's performance. These important findings have serious implications for credit cards and the consumer lending industry at large as fraud techniques become more powerful and nuanced, requiring data scientists to leverage more powerful methods that can adequately hobble a fraudster's learning process.

Possible future work is considerable. Rather than focusing on a specific classification threshold, the modeler could replace the logistic regression classifier with other models such as Random Forests, KNNs, Support Vector Machines, etc. Different models might be stronger against certain fraud profiles and swapping models regularly could be confusing to the adversary. Similarly, hyperparameters could be shifted instead of classification thresholds to observe the effect on the fraudster's effectiveness. In our analysis, model classification thresholds were swapped at random and at regular intervals. This swapping could be done in a more methodical manner or at irregular intervals. Currently our model makes particularly simplistic assumptions about the state and action space available to the fraudster. In reality a fraud adversary has control over things like their distance from home when they make a transaction, the merchant category code of their purchase, and a more granular control of the dollar amount of the transaction to name only a few. Including these features would certainly contribute to an increase in the performance of the classifiers. Finally, one last consideration is the choice of algorithm used to accomplish policy control. Our work used Monte Carlo given its easy interpretation and implementation, but SARSA and Q-learning are methods that tend to learn more quickly and with greater accuracy that could be implemented and may more closely represent the speed at which a human fraudster would actually learn.

ACKNOWLEDGMENT

Thank you to our project advisors at the University of Virginia for teaching, guiding, and helping us throughout our project. Additionally, we would like to thank Capital One for insight and guidance on the use case for this paper.

REFERENCES

[1] The Nilson Report, October 2016, Vol. 1096.
[2] Nelson, B., Barreno, M., Chi, F. J., Joseph, A. D., Rubinstein, B. I. P., et al (2008). Exploiting machine learning to subvert your spam filter. In Proceedings of the workshop on large-scale exploits and emerging threats (LEET).
[3] Nilesh Dalvi , Pedro Domingos , Mausam , Sumit Sanghai , Deepak Verma, Adversarial classification, Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, August 22-25, 2004, Seattle, WA, USA [doi>10.1145/1014052.1014066]
[4] M. F. Zeager, A. Sridhar, N. Fogal, S. Adams, D. E. Brown, and P. A. Beling, "Adversarial learning in credit card fraud detection," in Systems and Information Engineering Design Symposium (SIEDS), 2017. IEEE, 2017, pp. 112–116.
[5] Wei Liu , Sanjay Chawla, A Game Theoretical Model for Adversarial Learning, Proceedings of the 2009 IEEE International Conference on Data Mining Workshops, p.25-30, December 06-06, 2009 [doi>10.1109/ICDMW.2009.9]
[6] Lerrel Pinto, James Davidson, Rahul Sukthankar, and Abhinav Gupta. Robust adversarial reinforcement learning. arXiv preprint arXiv:1703.02702, 2017.
[7] D. Shen, G. Chen, E. Blasch, G. Tadda, "Adaptive Markov game theoretic data fusion approach for cyber network defense", Military Communications Conference 2007. MILCOM 2007. IEEE, pp. 1-7, 2007.

AUTHOR INFORMATION

Adrian Mead, M.S. Student, Data Science Institute, University of Virginia.
Tyler Lewris, M.S. Student, Data Science Institute, University of Virginia.
Sai Prasanth, M.S. Student, Data Science Institute, University of Virginia.
Stephen Adams, Senior Scientist, Systems and Information Engineering, University of Virginia.
Peter Alonzi, Senior Research Data Scientist/Computation, Research Data Services, University of Virginia.
Peter Beling, Associate Professor, Department of Systems and Information Engineering, University of Virginia.