RESONANT: Reinforcement Learning-based Moving Target Defense for Credit Card Fraud Detection

George Abdel Messih
Department of Computer Science, Virginia Tech
Blacksburg, VA, USA
National Security Institute, Virginia Tech
Arlington, VA, USA
george98@vt.edu

Tyler Cody
Grado Department of Industrial and Systems Engineering, Virginia Tech
Blacksburg, VA, USA
National Security Institute, Virginia Tech
Arlington, VA, USA
tcody@vt.edu

Peter Beling
Grado Department of Industrial and Systems Engineering, Virginia Tech
Blacksburg, VA, USA
National Security Institute, Virginia Tech
Arlington, VA, USA
beling@vt.edu

Jin-Hee Cho
Department of Computer Science, Virginia Tech
Falls Church, VA, USA
National Security Institute, Virginia Tech
Arlington, VA, USA
jicho@vt.edu

Abstract

According to *security.org*, as of 2023, 65% of credit card (CC) users in the US have been subjected to fraud at some point in their lives, which equates to about 151 million Americans. The proliferation of advanced machine learning (ML) algorithms has contributed to detecting credit card fraud (CCF). However, using a single or static ML-based defense model against a constantly evolving adversary takes its structural advantage, which enables the adversary to reverse engineer the defense's strategy over the rounds of an iterated game. This paper proposes an adaptive *moving target defense* (MTD) approach based on *deep reinforcement learning* (DRL), termed RESONANT, to identify the optimal switching points to another ML classifier for credit card fraud detection. It identifies optimal moments to strategically switch between different ML-based defense models (i.e., classifiers) to invalidate any adversarial progress and always take a step ahead of the adversary. We take this approach in an iterated game theoretic manner where the adversary and defender take action in turns in the CCF detection contexts. Via extensive simulation experiments, we investigate the performance of our proposed RESONANT against that of the existing state-of-the-art counterparts in terms of the mean and variance of detection accuracy and attack success ratio to measure the defensive performance. Our results demonstrate the superiority of RESONANT over other counterparts, including static and naïve ML and MTD selecting a defense model at random (i.e., Random-MTD). Via extensive simulation experiments, our results show that our proposed RESONANT can outperform the existing counterparts up to two times better performance in detection accuracy using AUC (i.e., Area Under the Curve of the Receiver Operating Characteristic (ROC) curve) and system security against attacks using attack success ratio (ASR).

CCS Concepts

• **Applied computing** → **Secure online transactions**; **Online banking**; **Electronic data interchange**; **Digital cash**; **E-commerce infrastructure**; **Online shopping**; **Multi-criterion optimization and decision-making**; **Business intelligence**; *Business process monitoring*; **Business process modeling**; **Business process management systems**; IT governance; *IT architectures*; *Enterprise modeling*; **Enterprise data management**; **Service-oriented architectures**; • **Security and privacy** → *Database activity monitoring*.

Keywords

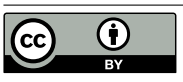
Iterated games, adversarial learning, moving target defense, fraud detection, reinforcement learning.

ACM Reference Format:
George Abdel Messih, Tyler Cody, Peter Beling, and Jin-Hee Cho. 2024. RESONANT: Reinforcement Learning-based Moving Target Defense for Credit Card Fraud Detection . In *Proceedings of the 11th ACM Workshop on Adaptive and Autonomous Cyber Defense (AACD '24), October 14–18, 2024, Salt Lake City, UT, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3689935.3690395

1 Introduction

1.1 Motivation

Advances in modern technologies have introduced rapid growth of e-commerce and digital marketplaces, which significantly increase the convenience of online shopping with fast and secure digital transactions. However, these digital transactions are still susceptible to fraud. Credit card fraud (CCF) is a widespread business problem worth billions of dollars across the world [32, 33]. In practice, many artificial intelligence and machine learning (AI/ML) models have been used to defend against fraudulent transactions [32, 33, 35].



AACD '24, October 14–18, 2024, Salt Lake City, UT, USA

ACM ISBN 979-8-4007-1231-9/24/10
https://doi.org/10.1145/3689935.3690395

Nevertheless, fraudsters have continuously developed new techniques to infiltrate these online-based systems with malicious intent, such as evasion, poisoning, and exploratory attacks [32, 35]. Most existing approaches aim to develop stronger yet static defensive ML models that often fail because they cannot deal with advanced adversaries capable of reverse-engineering the defender's model over time [12, 26]. Thus, the need for novel, effective, dynamic ML defenses adaptable to the ever-evolving adversarial attack strategies has been recognized [19].

There are **drawbacks to existing defense strategies**, such as adversarial training (AT) and generative adversarial networks (GANs). For instance, AT assumes that the fraudulent transactions the defense trains on will be very similar to the fraud faced in the real world [19]. In addition, the min-max nature of GANs makes similar assumptions regarding the knowledge of the adversary's pay-off structure [1, 21]. Since the real-world adversary is unknown and evolving, these assumptions, even if accurate initially, often lead to performance deterioration over time when the defense is deployed in real-world settings. Adversaries may not be rational, may vary their goals and attack strategies, and may directly exploit such assumptions. Even when the assumptions are appropriate, Arora et al. [1] showed how imposing realistic system conditions on the discriminator, such as finite capacity and the limited number of training steps, can disprove the GANs' ability to reach an equilibrium.

1.2 Research Goal & Key Ideas

We aim to avoid such assumptions by introducing an adaptable ML/AI-based defense. We adopt the principles of *moving target defense* (MTD), a proactive defense technique to increase attackers' uncertainty by changing attack surfaces. This approach will challenge the attacker to predict the defender's actions and adapt to their defenses [7]. Nonetheless, the MTD's effectiveness is significantly influenced by the defender's ability to predict potential attacks or how much knowledge the adversary has about the defense model [7, 8]. To address this limitation, we leverage the *reinforcement learning* (RL) technique to intelligently determine what ML classifier to ensure the robustness of the fraud detection process against CCF attacks. The RL agent chooses an ML classifier to maximize its reward per round. The RL agent will allow learning via trial and error and thus does not need the adversary's knowledge due to the nature of RL's autonomous decision-making process. We name our proposed approach RESONANT, representing 'Reinforcement lEarning-baSed mOviNg tArget defeNse deTection' for CCF.

1.3 Key Contributions

Our proposed RESONANT has the **key contributions**:

(1) RESONANT is a novel MTD approach that continuously changes an ML classifier to detect CCF. The underlying principle is to eliminate the nature of a static ML approach, whose knowledge and security vulnerabilities can be easily exploited by adversaries.
(2) RESONANT leverages RL to create a model-free defense without the strong assumptions towards adversarial behaviors, often generating modeling errors and performance deterioration. In addition, for more realistic modeling of attack-defense interactions, we consider their iterative interactions in the context of CCF.
(3) Via extensive simulation experiments, we demonstrate the outperformance of the proposed RESONANT over the considered counterparts in terms of the detection effectiveness with minimum variance across the iterated interactions, attack success ratio, and algorithmic efficiency.

1.4 Paper Structure

This paper is structured as follows. Section 2 discusses the overview of the related work, mainly in terms of reinforcement learning-based moving target defense and ML-based CCF detection. Section 3 describes the network and threat models considered in this work. Section 4 describes the details of the proposed RESONANT in realizing MTD using RL. Section 5 describes datasets, comparing defense mechanisms and metrics to conduct valid experimental environments. Section 6 demonstrates the experimental results and analyzes their overall trends along with the underlying reasons. Section 7 concludes the paper and suggests future work directions.

2 Related Work

This section overviews the state-of-the-art reinforcement learning (RL)-based moving target defense (MTD) for cybersecurity applications and ML-based credit card fraud detection mechanisms.

2.1 RL-based MTD for Cybesecurity

Eghtesad et al. [16] proposed an RL-based MTD strategy applied in the cyber-security domain, where the RL agent's action set is either to do nothing or to select a server from the network and re-image it to reset the adversarial infiltration of that server. Eghtesad et al. [15] further expanded on [16] by proposing and assessing various extensions to their previously used RL-based defense, such as dueling and double Q-Network agents. Chowdhary et al. [8] proposed a similar RL-MTD framework in the cybersecurity domain where the RL agent has three possible actions [no action, network shuffle, IP mutation]. This work showed that utilizing an RL-based MTD against an adaptive attacker produces better results than deploying a random selection-based MTD. Zhu et al. [37] also proposed an iterative RL-based MTD for the cybersecurity domain. The authors mathematically proved that their RL algorithm can perfectly reach convergence against another RL algorithm simulating the attacker's actions. Yet, the authors designed their framework in a min-max nature between the two RL agents and evaluated the performance of their proposed defense accordingly. Their findings resulted in the same vulnerabilities of other min-max frameworks, such as GANs. Yoon et al. [34] proposed a multi-agent deep RL (DRL)-based MTD approach to enhance an in-vehicle network's performance by making two decisions related to link bandwidth allocation to meet quality-of-service (QoS) requirements and the frequency of triggering IP shuffling as an MTD technique. Chai et al. [4] proposed a DRL algorithm to decide the optimal duration between subsequent network shuffles for the security of cyber-physical systems to minimize defense resource consumption while maintaining a secure integration of wearable devices in people's everyday lives. Kim et al. [24] proposed a DRL-based automated

cybersecurity framework to run scalable and effective network traffic inspection for threat detection and network address shuffling as an MTD operation to proactively handle attacks. Bera et al. [3] proposed a MTD approach for defending against RL-based penetration testing the uses domain adaptation theory to guide network re-configurations.

Limitations: However, unlike the above works discussed, our work is the first to propose a model-free RL-based MTD in the ML robustness domain against attackers in the context of CCF detection. No prior work has applied the concept of MTD to increase the diversity of ML classifiers for CCF, leading to effective changes in the attack surface.

2.2 ML-based Credit Card Fraud Detection

Both Vimal et al. [33] and Zhinin-Vera et al. [36] proposed RL agents that mimic a typical ML classifier but with more advanced training metrics and learning models. Each RL agent described in these efforts interacts with a single transaction at a time and formulates its state from that transaction's features. The agent then takes a binary classification action of 0 for non-fraud or 1 for fraud. Dornadula and Geetha [14] proposed an ML-based CCF detection mechanism where customers are clustered into groups based on their historical transactions and behavioral patterns. The authors demonstrated the conventional ML-based CCF detection pipeline, where the imbalanced data was addressed using SMOTE. They used the conventional ML classifiers, such as decision trees, for CCF detection. Awoyemi et al. [2], Khatri et al. [23] and Dhankhad et al. [13] demonstrated comparative analysis frameworks on the effectiveness of the most common supervised learning ML models when applied in the CCF detection domain, such as k-nearest neighbors (KNN), Logistic Regression and Random Forest. The authors also provided a basic framework to address the class imbalance in CCF datasets, using either over-sampling, under-sampling, or both.

Limitations: However, both RL models in [33, 36] are only capable of making binary classification decisions, such as fraud or non-fraud. On the other hand, our proposed RESONANT is designed for an RL agent to make decisions at the meta-level of a game between the defender and the adversary to determine the optimal classifier to utilize at each step of the game. This achieves effective, reliable MTD strategies to maximize the system performance in terms of high detection accuracy and low attack success ratio. Further, unlike the static ML-based CCF detection methods in [2, 13, 14, 23], our proposed RESONANT incorporates RL to develop an effective MTD strategy to realize a dynamic, proactive defense capable of counteracting sophisticated attacks in CCF detection.

3 System Model

This section describes the considered system model in terms of network and threat models.

3.1 Network Model

As shown in Figure 1, we consider an enterprise network providing credit card (CC) services. In this network, a cloud server exists to process all online CC transactions and customers' deposits or withdrawals. Simultaneously, fraudsters send fraudulent transactions to that cloud server, aiming to make illegal CC transactions.

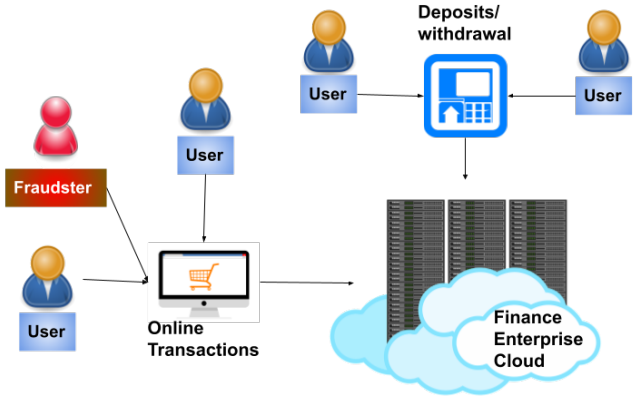


Figure 1: Example of a credit card enterprise network model.

3.2 Threat Model

We concern CCF adversaries aiming to increase their fraud infiltration rate by passing multiple fraud transactions. The adversaries observe what transactions are accepted and learn the infiltration patterns by repeating this process. By learning the system's vulnerabilities, the adversaries will make more intelligent fraudulent transactions to achieve the highest fraud acceptance rate. The most common adversarial attack type in CCF is *evasion attacks* by injecting fraudulent or adversarial transactions into the defender's system to maximize misclassification. The adversary designs and manipulates the malicious transactions to evade the defense system during the testing phase [5].

4 Proposed Approach: RESONANT

This section describes the overview of the proposed RESONANT in terms of attack-defense interactions, resolving class imbalance, and the RL agent's MTD decision-making.

4.1 Attack-Defense Interactions

Inspired by the game-theoretic framework in [9], we create a 50-round iterative game where both the adversary and the defender aim to maximize their respective utilities, simulating the attack-defense iterative interactions in the CCF domain. The attacker aims to maximize the number of accepted fraud transactions by selecting which transactions to over-sample and inject into the defender's system in each round. On the other hand, the defender aims to detect fraud transactions based on detection accuracy metrics (e.g., ROC-AUC score). The defender can achieve this in the conventional setting through two potential actions: retraining its ML classifier between rounds or implementing a random-MTD strategy. Our work introduces a defender agent that can autonomously make an optimal MTD decision based on RL.

Furthermore, we implement a label delay window (LDW) between undetected fraudulent transactions and receiving their truth labels, representing the delay window between undetected fraud occurrences and customers reporting these frauds.

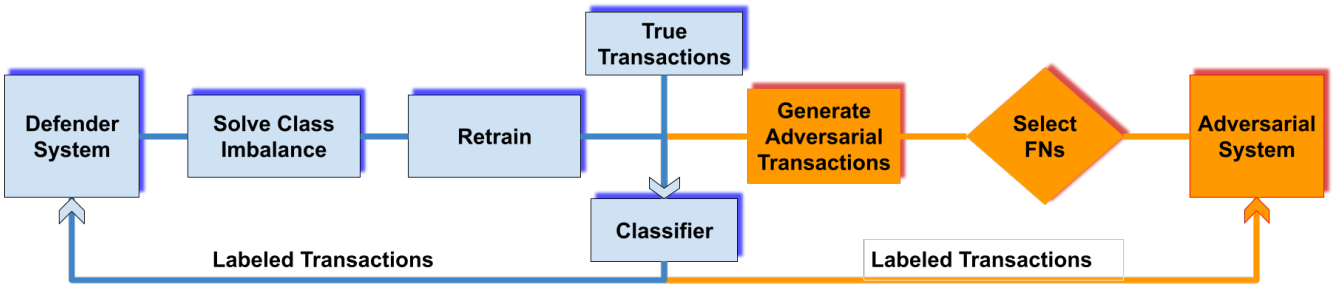


Figure 2: The overview of the interactions between the adversary and the defender system.

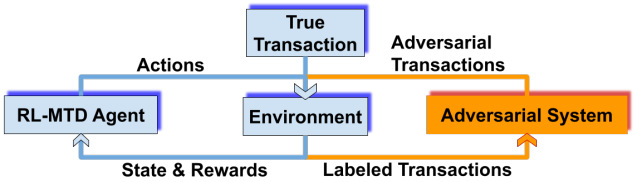


Figure 3: The procedure of RL-based MTD defense against CCF adversaries.

4.2 Attacker's and Defender's Goals and Actions

4.2.1 **Defender's Goal and Action.** The RL-MTD defense agent aims to maximize its reward function. It achieves its goal by choosing an optimal action, an ML classification model to deploy against CCF adversaries based on the environment's current state to maximize its reward.

4.2.2 **Attacker's Goal and Action.** We model adversarial behavior by creating an artificial fraud generation method performing evasion attacks under varying defense strategies controlled by our RESONANT in the iterative games modeled in this work. The fraud generation method was created using *the Synthetic Minority Oversampling Technique (SMOTE)-based Offensive Policy* [6], a semi-black-box targeted evasion attack. We choose SMOTE because of its merit in efficiency [25]. In this method, the adversary collects the accepted frauds from each round and applies the SMOTE to generate new fraudulent samples, which fall within the same decision space previously accepted by the defender's classifier. Next, the adversary injects the newly generated fraudulent transactions into the next round's defender test set. This process is repeated each round to generate new fraudulent samples. The adversary's goal in each round is to maximize the number of accepted fraudulent transactions, i.e., maximizing false negatives. Accordingly, in each round, the adversary must select which credit card transactions to over-sample using SMOTE and inject it into the next round's test data.

In addition, we demonstrate that the proposed testing method can highlight defense vulnerabilities that may not have shown in the single-round test typically used by other research efforts [2, 13, 23]. For instance, the static ML defense experiment shows how the defense performs well in the first round while the adversary is able to infiltrate the defender's system as more interactions occur in the iterative game. Accordingly, this proves that the snapshot type of experiments typically run in most other research efforts in the CCF detection domain is insufficient to evaluate the defense model's effectiveness.

4.3 Resolving Class Imbalance

One of the well-known challenges in most CCF detection models is a class imbalance issue. There are many more cases of legitimate transactions than fraudulent transactions. For instance, our data has only ~0.1% fraud. Accordingly, if the ML model training is performed on the unbalanced data, the model will learn to classify everything as non-fraud and have a 99.9% accuracy. However, none of the fraud will be detected. Therefore, data distribution between the fraudulent minority class and the non-fraudulent majority class should be balanced as the first step. There are multiple approaches to deal with imbalanced data, such as oversampling the minority class or under-sampling the majority class [29]. Over-sampling possibly causes over-fitting, while under-sampling leads to losing valuable information on the majority classes [27].

To avoid losing any useful information from under-sampling as well as provide more data for training a DRL model (e.g., DQN) of the RL-MTD agent, we use SMOTE to generate additional fraudulent transactions, the minority class [17]. While there are more advanced options for data augmentation, such as GANs, SMOTE was a better option in this framework to avoid the additional computational requirements associated with the alternatives [25]. After resolving class imbalance, various ML classifiers are trained on each round's labeled data and tested on the next round's new incoming data. We create and test a mixture of defensive strategies in the diverse simulated scenarios, detailed in Section 5.2.

4.4 RL-based MTD for Credit Card Fraud Detection

Inspired by [16, 31] using RL to identify optimal MTD strategies in the cybersecurity domain, we formulate the RL agent's action, reward, and state as below.

4.4.1 **Action Space.** The RL's action space consists of $\mathcal{A} = \{a_0, a_1, a_2, a_3, a_4\}$ where action i, denoted by a_i, refers to classifier i to be used in each round. We set each action to the following classifiers: (1) a_1: Logistic Regression classifier (LR); (2) a_1: Random Forest classifier (RF); (3) a_2: Decision Tree classifier (DT); (4) a_3:

AdaBoost classifier; and (5) a_4: XGBoost classifier (XGB). We select these well-known ML classifiers. One may choose other ML classifiers to further enhance the performance of CCF detection.

4.4.2 **Reward Function.** The RL agent aims to maximize its accumulate reward, $\mathcal{R}_t$ by taking the best action where $\mathcal{R}_t = \sum_{t=0}^{T} \gamma^t r_t$. Here r_t is an immediate reward at time t, γ is a discount factor, and T is the period of an episode. The agent's immediate reward, r_t, is formulated by:

$$r_t = \alpha * \mathcal{R}^X + \beta * \mathcal{R}^{TD}, \tag{1}$$

where $\mathcal{R}^X$ refers to the reward depending on the outcome of CCF detection (i.e., true positives or true negatives), and $\mathcal{R}^{TD}$ is the reward based on the transfer distance (TD) between the decision spaces of defensive classifiers [10, 11]. Specifically, we define $\mathcal{R}^X$ by:

$$\mathcal{R}^X = \begin{cases} +\frac{X_{11}}{X_{total}} & \text{if } t = 1 \text{ and } p = 1, \\ -\frac{X_{10}}{X_{total}} & \text{if } t = 1 \text{ and } p = 0, \\ +\frac{\lambda X_{00}}{X_{total}} & \text{if } t = 0 \text{ and } p = 0, \\ -\frac{\lambda X_{01}}{X_{total}} & \text{if } t = 0 \text{ and } p = 1, \end{cases} \tag{2}$$

$$\mathcal{R}^{TD} = T_{LC,NC}, \tag{3}$$

where R^{TD} is ranged as a real number in $[0, 1]$, representing the TD between the classifier used in the last round of the iterated game and the new classifier selected by the RL agent. For instance, $T_{LC,NC}$ stands for the TD between the classifier used in the last round, named the *last classifier* (LC) and the newly selected classifier, named the *new classifier* (NC). For example, $T_{1,1}$ refers to $(LC, NC) = (1, 1)$, which means LC is RF and NC is RF, resulting in $T_{1,1} = 0$. The TD value is calculated based on the hamming distance between the predicted fraud/non-fraud arrays of each classifier, and the classifier used last round [30].

4.4.3 **State Space.** The state space, $\mathcal{S}$, is defined based on (1) each round's Confusion Matrix data, $\mathbf{C}_M$ after applying λ hyper-parameter in Eq. (2); (2) each round's TD values: $\mathbf{T}_D = (T_{LC,0}, T_{LC,1}, T_{LC,2}, T_{LC,3}, T_{LC,4})$; (3) The current classification model, $\mathcal{M}$; and (4) Number of rounds the current classifier has been consecutively deployed, denoted by $\mathcal{N}_C$. Therefore, we denote $\mathcal{S} = \{\mathbf{C}_M, \mathbf{T}_D, \mathcal{M}, \mathcal{N}_C\}$.

4.5 Training of RL-MTD Agent

The RL-MTD agent is trained based on a Deep Q-learning Network (DQN), which uses a value-based RL agent to train a Deep Neural Network (DNN) and estimate its rewards for each round using its state-action values. We choose DQN, which is known as the most suitable option to learn the optimal MTD policy in this effort [4, 22]. Our RL agent is trained in an iterative framework, with each training episode composed of a single round of data. Given the DNN nature of the agent, a larger number of datasets is needed for the agent to be trained. Accordingly, we created 100 smaller datasets based on random sampling. Then, the RL agent is trained for 200 episodes, using the 100 smaller datasets twice. Since our agent's reward is normalized based on the total number of transactions in each round, the ranges of the RL-MTD agent's reward and state space values are not affected by the size of the datasets.

Table 1: Key Design Parameters, Their Meanings, and Default Values

Notation	Meaning	Default
LDW	Number of rounds between undetected frauds and receiving their delayed label	2
F_R	Percentage of adversarial fraud transactions in each round's data	3 %
ε	Number of RL training epochs	200
R	Number of test rounds in each game	50
λ	RL agent's $\mathcal{R}^X$ reward weight for non-fraud classifications - $[0, 1]$	0.5
α	Reward multiplier for $\mathcal{R}^X$ - real number	6
β	Reward multiplier for $\mathcal{R}^{\mathcal{TD}}$ - real number	8
γ	RL agent's learning discount factor - [0,1]	0.99

During the training phase of the RL-MTD agent, we tested the effect of training two different levels of attack involvement on defense performance where the agent is (1) trained and tested on the same adversary and (2) trained on one adversary and tested on another adversary. Under both cases, our proposed RESONANT does not require prior knowledge of adversaries to produce high effectiveness of its defense because our model uses the model-free design with RL.

5 Experimental Setup

This section describes the datasets, comparing defense schemes and metrics used for the conducted experiments.

5.1 Datasets

The utilized data [25] was provided by a financial institution engaged in the retail banking industry. The dataset was comprised of ≈ 80 million anonymized credit card transactions spanning eight months. The dataset contained 70 features, and the transactions were already pre-labeled as fraud or non-fraud. However, to mitigate the curse of high dimensionality in data, we reduced the data dimensionality to 11 features per transaction: (1) Fraud Indicator; (2) Approved Authorization Count; (3) Average Daily Authorization Amount; (4) Merchant Category ID; (5) Point of Service Entry Method ID; (6) Recurring Authorization indicator; (7) Distance From Home; (8) Current Account Balance; (9) Authorization Amount; (10) Authorization Outstanding Amount; and (11) Plastic Issuance Duration.

The data is split into separate subsets to implement the iterated adversarial game framework. A separate dataset is used for each round. The first game is designed to model the interaction between the ML defense and the adversary (see Figure 2). The second game is tailored to model the interaction between the RL-MTD defense and the adversary (see Figure 3).

All the data subsets used in the various experiments contain anonymized real-world fraudulent transactions. These transactions serve as real-world seeds for our adversarial fraud generation methods. The dataset can provide realistic fraud transactions in all rounds. Specifically, by the final round, we do not oversample the artificial fraud data created by the previous rounds' oversampling steps.

Table 2: Big-O Complexity Analyses of the Four Comparing Schemes

Defense Method	Static ML	Naïve ML	Random MTD	RESONANT
Big-O	$O\Big(K \times n^2 \times \log(n)\Big)$	$O\Big(R \times K \times n^2 \times \log(n)\Big)$	$O\Big(R \times K \times n^2 \times \log(n)\Big)$	$O\Big(\mathcal{E} \times O(\sum_{l=1}^{2} F_{l-1} \times N_l^2 \times F_l \times \mu_l^2)\Big)$

(Notations: R is the number of rounds in the games. K is the number of variables. n is the number of samples randomly drawn for each tree. ε is the number of DQN training epochs. l is the index of the deep-NN layer. F_{l-1} stands for the number of input channels of the l-th layer. F_l is the number of filters in the l-th layer. N is the size of the filter. μ_l is the size of the output feature map of the l-th layer.)

5.2 Comparison Defense Mechanisms

We will compare the performance of the following defenses:

- **Static ML**: This uses a single ML classifier with Random Forest (RF), proven the best-performing classifier among the ones considered. We train the RF classifier on the first round's balanced data and then use it to detect fraud across the iterated games without retraining.
- **Naïve ML**: This uses a single ML classifier (i.e., RF) with retraining at the start of each round on the last round's data to keep updated with the changing offensive fraud data trends. Hence, it is an advanced version of the Static ML but incurs extra costs for retraining.
- **Random MTD**: This selects a random classifier each round among the five ML classifiers (i.e., RF, DT, LR, XGB, and AdaBoost), and trains it on the last round's data.
- **RESONANT**: This is the proposed RL-MTD approach using DQN for the model training based on 100 smaller datasets where this scheme is detailed in Section 4.

For a fair comparison, we use the same set of hyperparameters in all defense mechanisms.

5.3 Metrics

We use the following metrics for our experiments:

- **AUC (Area under the ROC Curve)**: The AUC score is a great way to measure a classification task's performance with respect to true positive and false positive rates. In addition, it is not biased by class imbalances, which is imperative in the CCF detection domain.
- **Attack success ratio (ASR)**: This captures the number of successful fraud transactions over the total number of fraud attempts.
- **Asymptotic complexity in Big-O**: This captures the algorithmic running time in Big-O.

6 Numerical Results & Analysis

This section demonstrates our experimental results and discusses the underlying reasons for the overall trends in terms of algorithmic complexity analyses, comparative performance analyses, and sensitivity analyses under varying ratios of CCF transactions. For our experimental results in Sections 5.2 and 6.3, we use the default values for the key design parameters as described in Table 1.

6.1 Algorithmic Complexity Analyses

6.1.1 **Static ML.** This defense builds an RF model once and then repeatedly deploys it across the iterated rounds of the game without retraining. Hence, we can define its worst-case time complexity as the Big-O complexity of RF. Based on [28], it is given by:

$$O(\mathcal{K} \times n^2 \times \log(n)), \tag{4}$$

where n is the number of samples, and K is the number of variables randomly drawn at each node.

6.1.2 **Naïve ML & Random MTD.** Naïve ML and Random MTD defenses have the following complexity:

$$O((\mathcal{R} \times K \times n^2 \times \log(n)), \tag{5}$$

where R is the number of rounds in the iterated game, as both defenses fit the ML model in each round. In addition, both defenses have the same complexity because RF has a time complexity that is no less than that of the other models in the MTD framework. Therefore, the MTD will deploy the RF every round in the worst-case scenario.

6.1.3 **RESONANT.** The most time consuming step in RESONANT is training the DQN agent. Based on [18, 20], its worst-case time complexity given a fixed sample size n is given by[1]:

$$O\Big(\mathcal{E} \times \Big(\sum_{l=1}^{2} F_{l-1} \times N_l^2 \times F_l \times \mu_l^2\Big)\Big), \tag{6}$$

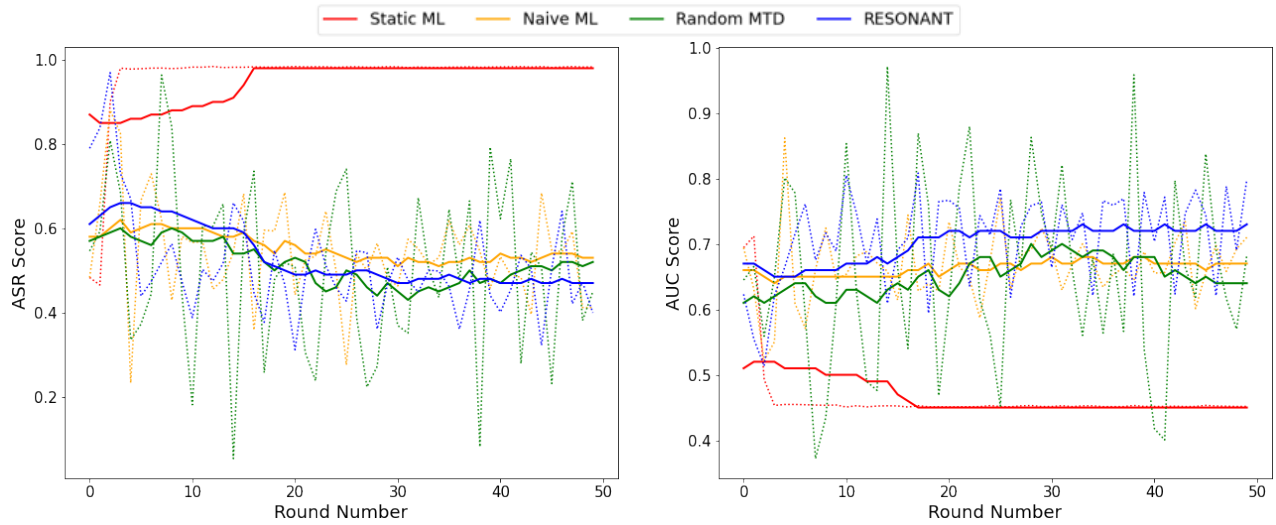
where ε is the number of DQN training epochs, l is the index of the deep-NN layer, F_{l-1} stands for the number of input channels of the l-th layer, F_l is the number of filters in the l-th layer, N is the size of the filter, and μ_l is the size of the output feature map of the l-th layer. After the agent is trained, each step of the game could involve re-training with the complexity given in Eq. (5). Therefore, the longer the agent is used, the lower its relative cost.

From our experiments, the DQN agent, RESONANT, takes significantly longer to train than the alternative defense approaches. Yet, the agent does not require retraining between the rounds of the iterative games, so it has a high initial computation cost but then has a complexity matching Eq. (5). Table 2 summarizes the Big-O complexities of all four schemes compared in this work.

6.2 Comparative Performance Analyses

Figure 4 compares the performance of all the considered defenses (i.e., classifiers with or without MTD) in terms of the AUC and ASR curves. To obtain clearer patterns of the results, we used a moving average with a window size of 15. The dotted lines are the actual scores, while the solid lines represent the moving averages. Figure 4a shows a similar ASR moving average for the Naïve ML, Random MTD, and RESONANT defenses. Yet, it is worth to note

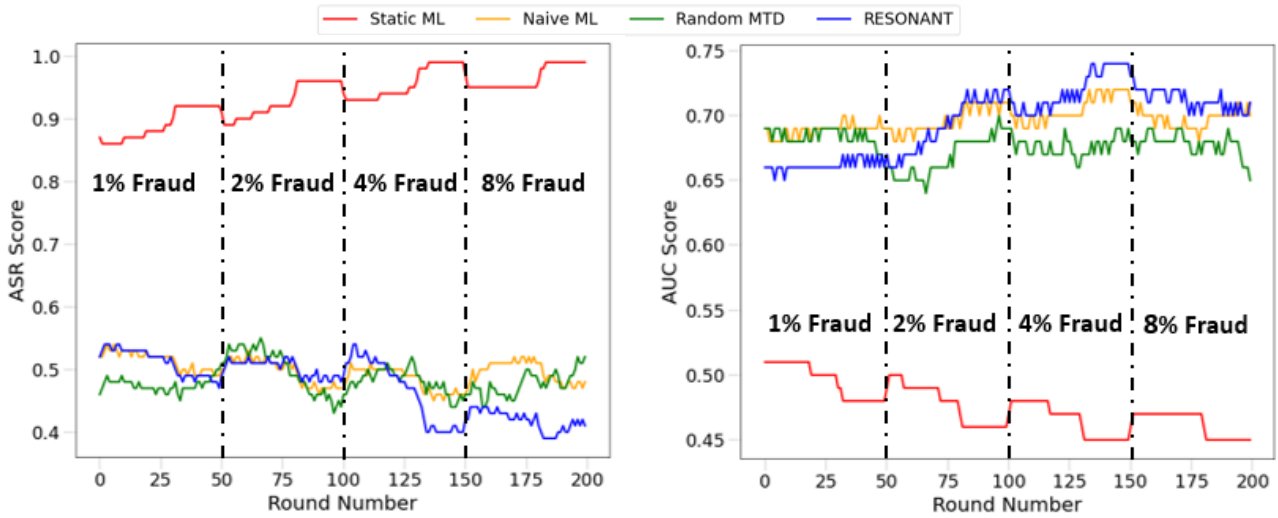
[1]Here given for DQN agent's that use convolutional neural networks (CNN) even though we use fully connected layers.



(a) ASR Under 3% CCF Transactions and 2 Round LDW

(b) AUC Under 3% CCF Transactions and 2 Round LDW

Figure 4: Comparative Performance Analyses of Static ML, Naïve ML, Random MTD, and RESONANT in ASR and AUC Under 3% CCF Transactions and 2 Round Label Delay Window (LDW).



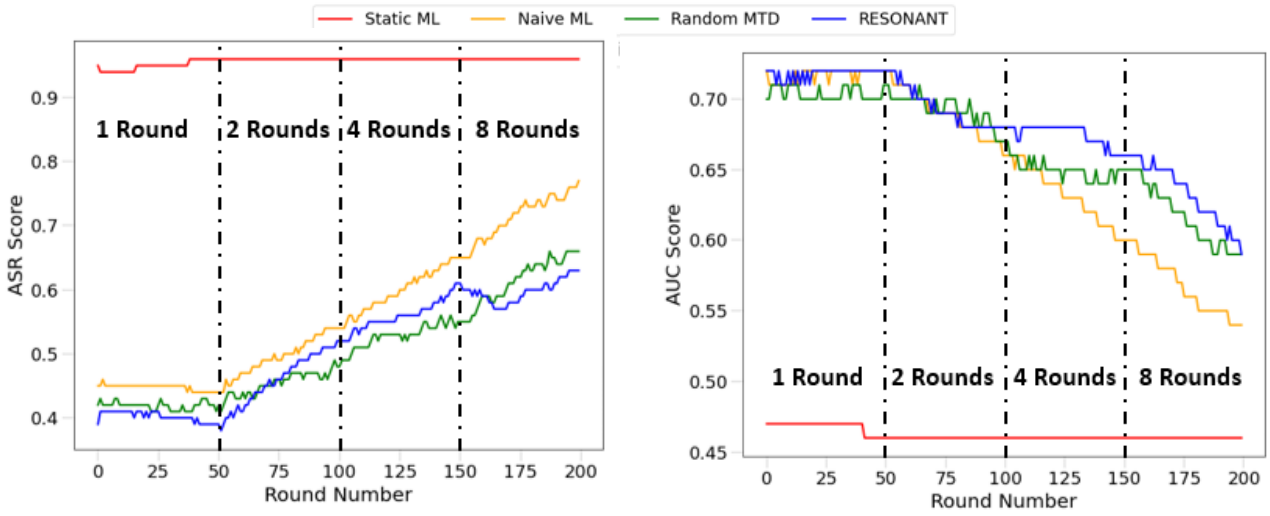
(a) ASR Under Varying the Percentage of CCF Transactions

(b) ASR Under Varying the Percentage of CCF Transactions

Figure 5: Comparative Performance Analyses of Static ML, Naïve ML, Random MTD, and RESONANT in ASR and AUC Under Varying the Percentage of CCF Transactions.

the dotted line of the Random MTD shows high fluctuations, representing high unpredictability (i.e., uncertainty) in performance, compared to the RESONANT defense. On the other hand, Figure 4b shows a clear outperformance of the RESONANT defense in AUC.

We also observe the average and variance of the AUC-ROC and ASR scores in the games in Figure 4 in Table 3. Based on this analysis, we can observe that RESONANT achieves the highest average AUC score while maintaining a relatively low variance in AUC. Specifically, RESONANT results in an AUC variance four times less than that of the Random MTD. On the other hand, Random MTD achieves the lowest ASR average score yet also results in the highest ASR variance. Accordingly, by examining both the average and variance in ASR, RESONANT produces the second lowest ASR average while having a three times less variance than that of Random MTD.



(a) ASR Under Varying LDW

(b) ASR Under Varying LDW

Figure 6: Comparative Performance Analyses of Static ML, Naïve ML, Random MTD, and RESONANT in ASR and AUC Under Varying Label Delay Windows (LDW).

Table 3: Performance Summary of the Iterated Adversarial Games

Defense Strategy	Average AUC	AUC Variance	Average ASR	ASR Variance
Static ML	0.463	**0.002**	0.96	**0.010**
Naïve ML	0.666	0.003	0.543	0.014
Random MTD	0.656	0.02	**0.504**	0.041
RESONANT	**0.708**	0.005	0.512	0.016

6.3 Effect of Varying the CCF Rate

Figure 5 analyzes the effect of varying the degree of attempted CCF transactions in percentage on ASR and AUC. Higher CCF rate represents higher severity in CCF attacks. Figure 5a shows that RESONANT performs comparably as Naïve ML and Random MTD when the CCF rate is relatively low (i.e., $< 4\%$) while it outperforms under higher CCF rates (i.e., ≥ 4) in ASR. Figure 5b demonstrates the clear out-performance of RESONANT overall (i.e., ≥ 2 %). A moving average with a window size of 30 was used to clearly obtain the overall patterns of the results.

6.4 Effect of Varying the Size of Label Delay Windows

In Figure 6, we also investigate the performance of different defenses against CCF attacks while varying the size of the label delay window. The label delay window refers to the number of rounds between an occurrence of undetected fraud and receiving its truth label, representing the window between fraud occurrences and receiving complaints from the CC owners. A larger label delay window represents a harsher condition to the defender because having more delayed access to the truth labels makes it harder for the defender to make accurate fraud detection decisions. Moreover, we utilized a moving average with a window size of 50 to clearly obtain the overall patterns of the results. Figure 6a shows a comparable performance of the Random MTD and RESONANT defenses in ASR. RESONANT outperforms the other defenses at the 1-round and 8-round delay windows, while the Random MTD outperforms at the 2-round and 4-round delay windows. On the other hand, Figure 6b illustrates a tied performance between the Naïve ML and RESONANT defenses at the lower fraud rates. Yet, there is a clear outperformance of the RESONANT defense under the harsher adversarial conditions, starting at a 4-round label delay window. Test sets utilized in the games include more than 10 million transactions. Thus, improvements in AUC scores can represent significantly large monetary gains to the defender.

7 Conclusions & Future Work

While prior defense mechanisms against adversarial examples aim to optimize a fixed target defense, this effort aims to develop a novel model-free MTD framework based on the autonomous nature of RL in identifying optimal solution(s).

We recapitulate the **key contributions** of the proposed credit card fraud (CCF) detection as follows:

- We proposed a novel MTD strategy to bypass vulnerabilities and risks introduced by using the static nature of most current defenses.
- We designed a deep-RL agent named RESONANT to create a model-free defense for an RL agent to autonomously identify the MTD's optimal triggering conditions.
- We modeled attack-defense interactions in a game-theoretic sense to reflect realistic scenarios. This was realized by employing the iterative interaction between the fraudsters and the defender (finance companies) in the real world, representing more realistic representations of real-world adversarial capabilities in the CCF contexts.

From this study, we obtained the following **key findings**:

- The proposed defense method results in a more effective and consistent defensive performance over time, exhibiting a higher average performance with a lower variance.
- Using RESONANT is more beneficial in harsher conditions than friendly conditions, such as higher rates of CCF or longer delays to obtain the truth information (i.e., longer label delay windows).

For **future research**, we plan to (1) experiment with other deep-RL agents in search of the optimal MTD decision-making agent; (2) expand the action space of the agent to include manipulating both which ML model to use and what hyper-parameters to use for a particular model; and (3) utilize a DNN-based artificial fraud generation method to explore the performance of RESONANT against a DNN-based adversarial model and compare it to that of other defense techniques.

8 Acknowledgements

Project sponsored by the National Security Agency under Grant Agreement Number H98230-21-1-0322 and the US Army Research Office with a grant W911NF-20-2-0140. The United States Government is authorized to reproduce and distribute reprints notwithstanding any copyright notation herein.

References

[1] Sanjeev Arora, Rong Ge, Yingyu Liang, Tengyu Ma, and Yi Zhang. 2017. Generalization and equilibrium in generative adversarial nets (gans). In *International Conference on Machine Learning*. 224–232.

[2] John O Awoyemi, Adebayo O Adetunmbi, and Samuel A Oluwadare. 2017. Credit card fraud detection using machine learning techniques: A comparative analysis. In *2017 International Conference on Computing Networking and Informatics (ICCNI)*. 1–9.

[3] Shreyas Bera, Liam Glenn, Abhay Raghavan, Emma Meno, Tyler Cody, and Peter A Beling. 2023. Deterring adversarial learning in penetration testing by exploiting domain adaptation theory. In *2023 Systems and Information Engineering Design Symposium (SIEDS)*. IEEE, 314–318.

[4] Xinzhong Chai, Yasen Wang, Chuanxu Yan, Yuan Zhao, Wenlong Chen, and Xiaolei Wang. 2020. DQ-MOTAG: deep reinforcement learning-based moving target defense against DDoS attacks. In *2020 IEEE Fifth International Conference on Data Science in Cyberspace (DSC)*. IEEE, 375–379.

[5] Anirban Chakraborty, Manaar Alam, Vishal Dey, Anupam Chattopadhyay, and Debdeep Mukhopadhyay. 2018. Adversarial attacks and defences: A survey. *arXiv preprint arXiv:1810.00069* (Sep. 2018).

[6] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. 2002. SMOTE: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research* 16 (Jun. 2002), 321–357.

[7] Jin-Hee Cho, Dilli P Sharma, Hooman Alavizadeh, Seunghyun Yoon, Noam Ben-Asher, Terrence J Moore, Dong Seong Kim, Hyuk Lim, and Frederica F Nelson. 2020. Toward proactive, adaptive defense: A survey on moving target defense. *IEEE Communications Surveys & Tutorials* 22, 1 (Jan. 2020), 709–745.

[8] Ankur Chowdhary, Dijiang Huang, Abdulhakim Sabur, Neha Vadnere, Myong Kang, and Bruce Montrose. 2021. SDN-based Moving Target Defense using Multi-agent Reinforcement Learning. In *Proceedings of the first International Conference on Autonomous Intelligent Cyber defense Agents (AICA 2021), Paris, France*. 15–16.

[9] Tyler Cody, Stephen Adams, and Peter A Beling. 2018. A utilitarian approach to adversarial learning in credit card fraud detection. In *2018 Systems and Information Engineering Design Symposium (SIEDS)*. 237–242.

[10] Tyler Cody, Stephen Adams, and Peter A Beling. 2022. Empirically measuring transfer distance for system design and operation. *IEEE Systems Journal* 16, 3 (Feb. 2022), 4962–4973.

[11] Tyler Cody and Peter A Beling. 2023. A systems theory of transfer learning. *IEEE Systems Journal* 17, 1 (2023), 26–37.

[12] Richard Colbaugh and Kristin Glass. 2012. *Predictive Moving Target Defense*. Technical Report. Sandia National Lab.(SNL-NM), Albuquerque, NM (United States).

[13] Sahil Dhankhad, Emad Mohammed, and Behrouz Far. 2018. Supervised machine learning algorithms for credit card fraudulent transaction detection: a comparative study. In *2018 IEEE International Conference on Information Reuse and Integration (IRI)*. 122–125.

[14] Vaishnavi Nath Dornadula and Sa Geetha. 2019. Credit card fraud detection using machine learning algorithms. *Procedia Computer Science* 165 (Jan. 2019), 631–641.

[15] Taha Eghtesad, Yevgeniy Vorobeychik, and Aron Laszka. 2019. Deep reinforcement learning based adaptive moving target defense. *arXiv preprint arXiv:1911.11972* (Nov. 2019).

[16] Taha Eghtesad, Yevgeniy Vorobeychik, and Aron Laszka. 2020. Adversarial deep reinforcement learning based adaptive moving target defense. In *Decision and Game Theory for Security: 11th International Conference, GameSec 2020, College Park, MD, USA, October 28–30, 2020, Proceedings 11*. 58–79.

[17] Jianqing Fan, Zhaoran Wang, Yuchen Xie, and Zhuoran Yang. 2020. A theoretical analysis of deep Q-learning. In *Learning for Dynamics and Control*. 486–489.

[18] Ning Gao, Zhijin Qin, Xiaojun Jing, Qiang Ni, and Shi Jin. 2019. Anti-intelligent UAV jamming strategy via deep Q-networks. *IEEE Transactions on Communications* 68, 1 (Oct. 2019), 569–581.

[19] Ian Goodfellow. 2019. A research agenda: Dynamic models to defend against correlated attacks. *arXiv preprint arXiv:1903.06293* (Mar. 2019).

[20] Kaiming He and Jian Sun. 2015. Convolutional neural networks at constrained time cost. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5353–5360.

[21] Yongjun Hong, Uiwon Hwang, Jaeyoon Yoo, and Sungroh Yoon. 2019. How generative adversarial networks and their variants work: An overview. *ACM Computing Surveys (CSUR)* 52, 1 (Feb. 2019), 1–43.

[22] Anoop Jeerige, Doina Bein, and Abhishek Verma. 2019. Comparison of deep reinforcement learning approaches for intelligent game playing. In *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*. 0366–0371.

[23] Samidha Khatri, Aishwarya Arora, and Arun Prakash Agrawal. 2020. Supervised machine learning algorithms for credit card fraud detection: a comparison. In *2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*. 680–683.

[24] Sunghwan Kim, Seunghyun Yoon, Jin-Hee Cho, Dong Seong Kim, Terrence J Moore, Frederica Free-Nelson, and Hyuk Lim. 2022. DIVERGENCE: deep reinforcement learning-based adaptive traffic inspection and moving target defense countermeasure framework. *IEEE Transactions on Network and Service Management* 19, 4 (Jan. 2022), 4834–4846.

[25] Alex Langevin, Tyler Cody, Stephen Adams, and Peter Beling. 2022. Generative adversarial networks for data augmentation and transfer in credit card fraud detection. *Journal of the Operational Research Society* 73, 1 (Jan. 2022), 153–180.

[26] Cheng Lei, Hong-Qi Zhang, Jing-Lei Tan, Yu-Chen Zhang, and Xiao-Hu Liu. 2018. Moving target defense techniques: A survey. *Security and Communication Networks* 2018 (Jul. 2018).

[27] Enlu Lin, Qiong Chen, and Xiaoming Qi. 2020. Deep reinforcement learning for imbalanced classification. *Applied Intelligence* 50 (Aug. 2020), 2488–2502.

[28] Gilles Louppe. 2014. Understanding random forests: From theory to practice. *arXiv preprint arXiv:1407.7502* (Jul. 2014).

[29] Roweida Mohammed, Jumanah Rawashdeh, and Malak Abdullah. 2020. Machine learning with oversampling and undersampling techniques: overview study and experimental results. In *2020 11th International Conference on Information and Communication Systems (ICICS)*. 243–248.

[30] Mohammad Norouzi, David J Fleet, and Russ R Salakhutdinov. 2012. Hamming distance metric learning. *Advances in Neural Information Processing Systems* 25 (2012).

[31] Sailik Sengupta and Subbarao Kambhampati. 2020. Multi-agent reinforcement learning in bayesian stackelberg markov games for adaptive moving target defense. *arXiv preprint arXiv:2007.10457* (Jul. 2020).

[32] Aihua Shen, Rencheng Tong, and Yaochen Deng. 2007. Application of classification models on credit card fraud detection. In *2007 International Conference on Service Systems and Service Management*. 1–4.

[33] Siddharth Vimal, Kanishka Kayathwal, Hardik Wadhwa, and Gaurav Dhama. 2021. Application of deep reinforcement learning to payment fraud. *arXiv preprint arXiv:2112.04236*.

[34] Seunghyun Yoon, Jin-Hee Cho, Dong Seong Kim, Terrence J Moore, Frederica Free-Nelson, and Hyuk Lim. 2021. DESOLATER: Deep reinforcement learning-based resource allocation and moving target defense deployment framework. *IEEE Access* 9 (Apr. 2021), 70700–70714.

[35] Mary Frances Zeager, Aksheetha Sridhar, Nathan Fogal, Stephen Adams, Donald E Brown, and Peter A Beling. 2017. Adversarial learning in credit card fraud detection. In *2017 Systems and Information Engineering Design Symposium (SIEDS)*. 112–116.

[36] Luis Zhinin-Vera, Oscar Chang, Rafael Valencia-Ramos, Ronny Velastegui, Gissela E Pilliza, and Francisco Quinga-Socasi. 2020. Q-Credit Card Fraud Detector for Imbalanced Classification using Reinforcement Learning.. In *ICAART (1)*. 279–286.

[37] Minghui Zhu, Zhisheng Hu, and Peng Liu. 2014. Reinforcement learning algorithms for adaptive cyber defense against heartbleed. In *Proceedings of the first ACM Workshop on Moving Target Defense*. 51–58.