Robust Fraud Transaction Detection: A Two-Player Game Approach

Qi Tan*, Yi Zhao†, Laizhong Cui*,✉, Qi Li‡,✉, Ming Zhu§, Xing Fu¶, Weiqiang Wang¶, Xiaotong Lin¶, Ke Xu§,✉

*College of Computer Science and Software Engineering, Shenzhen University, Email: {tanqi, cuilz}@szu.edu.cn
†School of Cyberspace Science and Technology, Beijing Institute of Technology, Email: zhaoyi@bit.edu.cn
‡Institute for Network Science and Cyberspace, Tsinghua University, Email: qli01@tsinghua.edu.cn
§Department of Computer Science and Technology, Tsinghua University, Email: {minzhu, xuke}@tsinghua.edu.cn
¶Ant Group, Email: {zicai.fx, weiqiang.wwq, lxt203095}@antgroup.com

***Abstract*—Machine learning (ML)-based fraud detection systems are widely employed by enterprises to reduce economic losses from fraudulent activities. However, fraudsters are intelligent and evolve rapidly, employing advanced techniques to falsify the features of transactions to evade the detection system. Worse still, since these falsification processes are not restricted to small intervals, existing robustness enhancement methods based on small-scale perturbations are ineffective. Detecting unrestrictedly perturbed fraudulent activities, which significantly increases uncertainties in fraud detection, is still an open problem.**

To resolve this issue, we propose *GAMER*, a robust fraud detection system based on two-player game, achieving both high accuracy and strong robustness in detecting fraudulent activities. Specifically, *GAMER* leverages feature selection to proactively combat intelligent fraudsters in fraud detection (i.e., selecting fewer features to reduce the combinations of feature falsification), and innovatively formulates the detecting process as a two-player game. By solving the equilibrium of the two-player game, *GAMER* calculates the optimal probability for feature selection, which takes into account all possible falsification strategies of the fraudsters. The equilibrium-based selection probability not only minimizes the profits obtained by fraudsters, demotivating them to launch falsification; but also enables the system to select robust features (i.e., the features that are less likely to be falsified) in detecting fraudulent activities, enhancing the robustness of the system in fraud detection. Our theoretical and experimental results validate the properties of deterrence and robustness enhancement. Moreover, experiments on real-world attacks suffered by the world's leading online payment enterprise demonstrate that *GAMER* outperforms traditional techniques of robustness enhancement, which increases the F1 score by 67.5% on average for two-month fraud detection.

I. Introduction

Fraudulent activities are social issues that have far-reaching effects in industry and our daily life. According to the statistical study from *Juniper Research*, global payment fraud losses will exceed $343 billion before 2027 [46]. Combine it with *ABA Banking Journal*'s report[1], the corporate cost caused by fraudulent activities will exceed $1.495 trillion, which is 1.36% of the global GDP. As a result, enterprises are forced to enhance their fraud detection measures to protect themselves and their customers from financial damage [46]. To cope with highly complex fraud patterns, industry institutions have incorporated machine learning (ML)-based methods as the crucial part of fraud detection [74], [29], [71], [9]. Over 93% of the financial institutions have invested in the ML-based fraud detection system [48], and the market size is projected to reach $57.147 billion by 2033 [31].

However, traditional fraud detection systems cannot keep up with the fast evolving fraud activities. Specifically, fraudsters are constantly developing new tactics to evade ML-based detection systems [18], [21], leading to more sophisticated and stealthy fraud activities. These tactics are rooted in the deliberately falsified features in fraud transactions, which cause detection systems to misclassify them into benign transactions. Worse still, fraudsters can employ advanced techniques to detect vulnerabilities in the detection system, leading to more effective methods to falsify features.

The act of falsifying transaction features to bypass detection mechanisms can be formalized as executing adversarial attacks against targeted detection systems. Yet, there are three distinct characteristics in falsifying the features of fraud activities: (i) *the perturbations are unrestricted.* Transactions are constituted with monetary features (e.g., *Transfer Amount*, *Register Capital*) or temporal features (e.g., *Days After Certified*), these features are not restricted in small intervals; (ii) *the falsification process is resource consuming.* Unlike adversarial examples in the image or language field, which only changes pixels or semantics, falsifying features in fraud detection consumes resources (e.g., falsifying the feature of *Days After Certified* takes substantial time to maintain accounts); (iii) *the fraudsters are profit-driven.* Since fraudsters aim to get illegal profits from fraud activities as much as possible, the variation of profits exhibits a profound influence on their fraud behaviors (e.g., over 60% of the fraudsters gave up to purchase new accounts

[1]Every $1 lost to fraud costs $4.36 in related expenses (e.g., legal fees) [34].

Network and Distributed System Security (NDSS) Symposium 2026
23-27 February 2026, San Diego, CA, USA
ISBN 979-8-9919276-8-0
https://dx.doi.org/10.14722/ndss.2026.241611
www.ndss-symposium.org

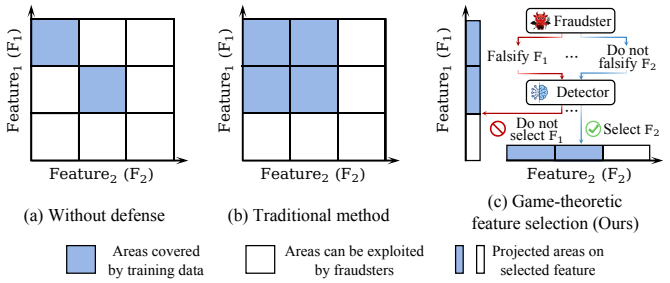


Fig. 1. Different strategies for robustness enhancement in fraud-transaction detection: (a) When the falsification process is unrestricted, fraudsters can manipulate a wide range of features to evade detection; (b) Traditional approaches restrict only a subset of manipulable features and therefore provide merely local robustness, leaving the detector vulnerable once falsification extends beyond those constrained dimensions; (c) Our proposed method formulates feature selection as a two-player game, enabling the detector to select a compact set of robust features that both narrows the exploitable attack surface and favors features that are inherently difficult for fraudsters to falsify through equilibrium-based selection.

to launch falsification when the costs of these accounts had increased)[2].

The aforementioned characteristics make the detection of falsified fraud transactions significantly different from traditional defense methods. On the one hand, classical techniques such as adversarial training, regularization, or data augmentation are local robustness enhancement methods, which are ineffective in detecting unrestrictedly perturbed fraud transactions. On the other hand, the latter two characteristics allow us to incorporate economical methods to design fraud detection systems, which is fundamental to solving security issues [26], [3].

According to these intuitions, the effective detection of falsified fraud transactions depends on reducing the impact of unrestricted perturbations and taking advantage of fraudsters' economical properties, which motivate us to design the system with game-theoretic feature selection. To be more precise, selecting fewer features in fraud detection reduces the combinations of perturbed features, raising the difficulties of falsification. Moreover, employing game theory in feature selection maximizes the effectiveness of selected features [81], minimizing the profits of fraudsters.

To this end, we develop *Game Selection*, a game-theoretic robust detecting method that leverages the equilibrium of a two-player game in feature selection. As displayed in Fig. 1, despite fraudsters can falsify features to evade the detection system, *Game Selection* can select not to use falsified features in detection, which fails the falsification process, resulting in economic loss for fraudsters. The selection probability depends on the equilibrium of a two-player game, which takes into account the cost-profit of falsifying different features and the fraudsters' response to any particular feature selection strategy. In particular, equilibrium-based feature selection enables the method to select robust features (i.e., the features that are less likely to be falsified) in detecting fraudulent activities, which enhances the robustness of the detection method. Furthermore, it also minimizes the profits of attacks, exhibiting deterrence for fraudsters[3].

Based on *Game Selection*, we propose *GAMER* (**G**ame-theoretic r**A**ndo**M**ized robust fraud d**E**tecto**R**), a robust fraud detection system that achieves high accuracy and strong robustness with hypothesis testing[4]. Specifically, *GAMER* utilizes the consistency of two models, one is trained by the classical method (e.g., SGD) to accurately detect unfalsified transactions and the other is equipped with *Game Selection* to robustly detect falsified transactions, to identify falsification in the transactions, thereby employing different models to detect the transactions they expertise in. Through hypothesis testing, *GAMER* leverages more statistical information of transactions, which is collected from two different models, to accurately identify the falsification behaviors of fraudsters, allowing the system to achieve high accuracy and strong robustness simultaneously.

Moreover, we develop a theoretical model to analyze the falsification process in fraud detection. Based on causal analysis [57], [27], we construct a causal diagram to model the entire life cycle of fraud detection, which systematically reveals the reasons for misclassification, indicating that selection bias, i.e., insufficient training data, is the cause of misclassification. Combining it with data coverage analysis[5] [7], [32], we prove that selection bias can be exponentially alleviated by selecting fewer features (Thm. 1), theoretically demonstrating that *GAMER* enhances the robustness of the detection system. Our experiments on irrational fraudsters, which do not take cost-profit information into consideration, validate this property, indicating that *GAMER* is more effective in detecting fraud transactions even if the transactions are irrationally falsified.

We establish a partnership with one of the world's foremost online payment enterprise and evaluate *GAMER* on real-world transactions provided by the enterprise. In particular, the enterprise experienced large-scale attacks in January, 2023, which doubled the asset loss rate of the enterprise in that month. Meanwhile, these attacks provide numerous data of falsified transactions, which can be used to evaluate *GAMER* over real-world transactions. To the best of our knowledge, this is the first time in fraud detection that real-world falsified transactions are used for evaluation. We discover that *GAMER* outperforms traditional adversarial training techniques and improves the F1 score by 67.5% on average in detecting real-world transactions.

The contributions of this paper are four-fold:

- We formulate the issue of detecting falsified fraud transactions as a two-player game between the detection system and the fraudsters, and incorporate the equilibrium to design the optimal detection strategy *Game Selection*.
- We propose a game-theoretic fraud detection system, namely *GAMER*, based on *Game Selection* and hypothesis

[2]Originate from the statistics of the platform we cooperate with.

[3]fraudsters will take into account that feature falsification consumes resources [4], [42] but can be useless once this feature is not selected.

[4]Hypothesis testing is a classic statistic method to decide whether the data sufficiently supports a particular hypothesis [73].

[5]Data coverage analysis aims to estimate if there are enough samples in the dataset for each category [6].

testing, which achieves both high accuracy and strong robustness.
- We develop a theoretical model to analyze falsification in fraud detection, which demonstrates that *GAMER* not only exhibits deterrence to fraudsters, but also enhances detection robustness by utilizing fewer features.
- We validate *GAMER* on real-world transactions collected by the world's leading online payment enterprise. The system improves the F1 score by 67.5% on average for two-month fraud detection.

II. Problem Formulation and Preliminary

A. Problem Formulation

In order to detect real-world frauds, we first formalize the unrestricted falsification process to clarify the target issue in fraud detection. Then we formulate the issue of detecting falsified transactions as a two-player game and get the solution based on solving the equilibrium. Finally, we employ causal analysis and data coverage analysis to prove the effectiveness of the detection model.

Regarding the falsification process in fraud detection, there is a target classifier $F(\boldsymbol{W};\cdot)$ for fraudsters, where $\boldsymbol{W}$ is the model parameter of the classifier. The goal of the classifier is to identify a specific transaction $\boldsymbol{x}$ as fraud or not, i.e., $F(\boldsymbol{W};\boldsymbol{x}) \in \{1,0\}$. Moreover, $\boldsymbol{x}$ is also associated with a real label $y_{\boldsymbol{x}}$, therefore the target classifier naturally divides the set of all possible $\boldsymbol{x}$, i.e., $\boldsymbol{X}$, into two categories: $\boldsymbol{X}_{\text{clean}} = \{\boldsymbol{x} \mid F(\boldsymbol{W};\boldsymbol{x}) = y_{\boldsymbol{x}}\}$ and $\boldsymbol{X}_{\text{adv}} = \{\boldsymbol{x} \mid F(\boldsymbol{W};\boldsymbol{x}) \neq y_{\boldsymbol{x}}\}$, where the latter is the set of adversarial examples. For clarity purposes, since falsified transactions are specific adversarial examples in fraud detection, we will use them without distinction in the following parts.

Based on these definitions, the target of a falsification process is to find an adversarial example that minimizes the costs of falsification. Hence, it can be formalized as

$$\arg\min_{\boldsymbol{x}^*} \; Cost(\boldsymbol{x},\boldsymbol{x}^*); \quad \text{s.t. } y_{\boldsymbol{x}} = y_{\boldsymbol{x}^*}, \boldsymbol{x}^* \in \boldsymbol{X}_{adv} \; . \tag{1}$$

The first constraint indicates that the fraudsters' goal is maintained, i.e., fraud or not. The second constraint requires the falsification to cause misclassification, which implies that the falsification is successful. Finally, the target of cost minimization reveals that fraudsters aim to maximize their profits. In particular, falsified examples are not required to be in the ϵ-neighborhood of the input transactions, thus Eq. (1) provides a more generic definition of adversarial examples [63], [8]. For example, traditional adversarial examples are generated by using *Lp*-distance, i.e., $\|\boldsymbol{x}-\boldsymbol{x}^*\|_p$, as the cost function [84].

B. Preliminary

Two-player Game. In this paper, we formalize the issue of detecting falsified fraud transactions as a two-player game between the detection system and the fraudster, and incorporate the equilibrium to design the detection strategy.

In particular, the two-player game aims to model the interactions between the detection system and the fraudster based on security game [55], [81], [66], which provides mathematical approaches for allocating security resources to maximize their effectiveness. These games have been widely used in solving social issues, e.g., airport security [59], wildlife security [23], etc. In the game, both the system and the fraudster decide their strategies to optimize their own profits. Specifically, there is a payoff matrix in the game to quantitatively depict the profit of different strategies. The payoffs in the matrix can be obtained by the knowledge acquisition from domain experts, e.g., the answers to a set of questions about the impact of attacks, which are created by domain experts [66]. With the payoff matrix, the equilibrium of this game can be solved, which consists of the best reactions of detection systems and fraudsters to their opponents.

In section V-A, we propose *Game Selection*, a game-theoretic robust strategy that achieves the optimal feature selection. In practice, despite fraudsters can falsify features to evade the fraud detection system, the system can select not to use falsified features for detection. These are strategies in the two-player game, and *Game Selection* calculates the optimal selection probabilities for each feature in equilibrium, which takes into account the cost-profit of falsifying different features and the fraudsters' response to any particular feature selection strategy.

Causal Analysis. Causal analysis [56], [58], [27], which has been widely used in security analysis [47], [67], [44], is an innovative methodology that can be used to identify causality in correlations. It overcomes the limitations of traditional statistics by constructing the causal diagram. As indicated in Fig. 2, causal analysis divides the correlations into two categories: causality and spurious correlations. The causality (i.e., the black arrows in Fig. 2) represents that the result variable Z will be changed accordingly when the cause variable X is changed. In particular, the chain structure in Fig. 2 is one of the three basic structures in the causal diagram. In this structure, X is the cause of Z and Z is the cause of Y, which means that X causes changes of Y through the mediation variable Z. When the data in the analysis is conditioned on the mediation variable Z (e.g., $Z \in \{0,1\}$ but we only use the data of $Z=1$ in the analysis), X is independent of Y [57].

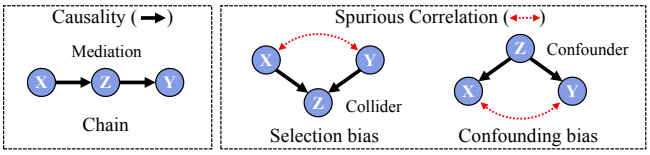


Fig. 2. Causal analysis identify causality from numerous correlations based on different properties of these structures.

The other two basic structures are the reasons for spurious correlations: the *selection bias* and the *confounding bias* in Fig. 2. In selection bias, X and Y are independent variables, but both are the causes of Z. When the data used in the analysis are conditioned on the collider variable Z, X becomes spuriously correlated to Y. Finally, in confounding bias, X and Y are actually independent but have a common cause Z, hence they are spuriously correlated. If the data used in the analysis are

conditioned on the confounder variable Z, X and Y become independent [57]. All casual diagrams are made up of these structures, and the main difference between them is the role of Z (i.e., mediation, collider, or confounder). Causal analysis uses these properties to identify causalities from a vast of correlations.

In section VI-A, we construct a causal diagram to model the entire life cycle of fraud detection, which systematically reveals the details of falsification process in fraud detection. Based on causal analysis, we discover that the spurious correlation between the real label and the predicted label caused by selection bias, i.e., insufficient training data, causes misclassification in fraud detection. Moreover, according to this result, we further demonstrate that our proposed methods can enhance the robustness of the detection system by using fewer features.

III. Threat Model and Problem statement

This paper studies the unrestricted falsification process in fraud detection, in which both the detection system and the fraudster are smart enough. Specifically, the detection system seeks to enhance the detection method to accurately detect fraud activities and to be robust to various falsified transactions. The fraudsters aim to take advantage of the deficiency of the fraud detection system to evade it by falsifying transactions. Moreover, the detection system and fraudster mutually impact each other (e.g., the fraudster will not falsify the feature once it is not selected by the detection system), trying to maximize their profits. This forms a two-player game [66] between them.

Threat Model. This paper considers strong fraudsters that can employ various techniques (e.g., gradient ascent [45], semantic composition [33], [62], [28], generative models [63], [72], or even LLMs [79], [49]) to discover vulnerabilities of the fraud detection system, which can seek out possible adversarial examples (i.e., $\forall\, \boldsymbol{x}^* \in \boldsymbol{X}_{adv}$) to evade the detection system. As indicated in Fig. 1, attackers can evade the system by falsifying input transactions. To combat smart detection systems, they can select to falsify different combinations of features to craft transactions (e.g., feature falsification via gradient ascent). Moreover, feature falsification consumes resources [42] (e.g., money or time) and the costs are distinct for falsifying different features (e.g., falsifying transfer amount costs less than falsifying the registered capital). The fraudsters are intelligent and smart enough, hence they try to minimize the cost of feature falsification.

Problem Statement. The goal of this paper is to tackle the unknown and sufficiently intelligent falsification process, which means that fraudsters can always discover adversarial examples of a specific detection system. In particular, the falsification process is unrestricted and intelligent, hence it cannot be throttled by existing methods, which utilize definite approaches in the detection (i.e., the detection model and the feature selection process are definite after deployment). Specifically, to combat intelligent falsification, the definite model must be perfect that is a model without any adversarial examples (i.e., $\boldsymbol{X}_{adv} = \varnothing$). This idealized target raises the asymmetric issue between detection systems and fraudsters, and the local robustness enhancement [52] is ineffective in this case. Moreover, perfectly training the detection model is unpractical when falsification processes are unrestricted since the detection system cannot completely collect the training data [6], [32], especially in the case of high-dimensional feature space [60].

To reverse the asymmetry between the detection system and the fraudster, accurately detect unrestricted falsified transactions, we develop *GAMER* to achieve the following goals.

(1) **Black-box Robustness Enhancement.** *GAMER* should not suppose the attack process, including constraints of falsification, ML algorithms, and background knowledge. This requirement is reasonable since the falsification process is black-box and unpredictable to the detection system, any unrealistic supposition will have the opposite effect once it is violated [22], [69].

(2) **Provable Deterrence Against Intelligent Fraudsters.** Even for unknown and intelligent falsification processes, *GAMER* can guarantee that there is no better profit for the fraudsters. Moreover, the strategy of *GAMER* should enable detection systems to select robust features in fraud detection.

(3) **High Accuracy and Strong Robustness.** *GAMER* should be as accurate as the non-robust model on clean examples, and also be robust to adversarial examples.

IV. Key Observations and Overview

Key Observations. In real-world fraud detection, fraudsters can deliberately craft transactions (i.e., falsify the features of a transaction in Fig. 1) after system deployment, which employs feature falsification to evade the detection system. Moreover, the falsification approach consumes resources [42], fraudsters will not falsify the feature if the profits of the falsification cannot cover the costs. As shown in Fig. 1, the fraud detection system can select features for fraud detection. The profit of falsifying a feature is bound to be 0 if it is not selected by the detection system, and smart enough fraudsters will not consume resources to falsify this non-profit feature (i.e., the unselected feature). These strategies interact with each other. Therefore, we can formulate the detection of falsified transactions as a two-player game and solve the equilibrium to get the optimal strategy for feature selection.

Data collected from a real-world enterprise provides concrete evidence regarding the cost of feature falsification. In particular, falsifying the Days After Certified attribute requires considerable time investment and ongoing maintenance to keep an account active. An alternative approach for fraudsters is to directly purchase aged accounts: empirical market data shows that accounts approximately six months old cost around $42, nearly twice the price of newly registered accounts (roughly $21). These observations demonstrate that falsifying this feature entails a non-trivial and economically measurable cost. As a result, designing detection methods that take this information into account can increase fraudsters' operational costs and serve as an effective deterrent against fraudulent behaviors.

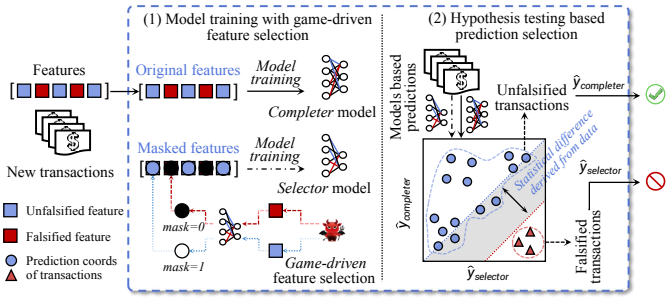


Fig. 3. Overview of *GAMER*.

Overview of *GAMER*. In this paper, we propose a novel fraud detection system, namely *GAMER*, to accurately and robustly detect real-world fraud transactions. Specifically, *GAMER* employs the equilibrium of a two-player game in feature selection, which enhances the system's ability to select unfalsified features in detection, improving the robustness of the system. Moreover, *GAMER* leverages hypothesis testing to identify falsification transactions, utilizing more information (the consistency of two models) in deciding the final prediction.

To be more precise, *GAMER* is composed of two models: *Completer* and *Selector* (as in Fig. 3). *Completer* is a model trained by classical methods (e.g., SGD) with all features, which aims to make full use of all features to accurately detect fraudulent transactions when transactions are unfalsified. *Selector* is a model equipped with game-driven feature selection, i.e., *Game Selection*, which employs the equilibrium of a two-player game in feature selection to robustly detect fraudulent transactions when transactions are falsified. In the *Game Selection*, the cost-profit knowledge of feature falsification is obtained from security experts [66] and the detection of falsified transactions is formalized as a two-player game. By solving for equilibrium of the game, *Game Selection* calculates the optimal probability for feature selection, which motivates the system to select unfalsified features in detection, thereby enhancing the robustness of the detection system. Moreover, equilibrium-based feature selection also minimizes profits for fraudsters, demotivating them to launch feature falsification.

After deployment, *GAMER* uses both models to detect fraudulent activities. On the one hand, all features are sent to *Completer* to get *Completer* prediction. On the other hand, features are masked with equilibrium-based Bernoulli variables to select input features, then the masked features are sent to *Selector* to get *Selector* prediction. Finally, the absolute distance between these predictions is used for hypothesis testing, which uses statistical difference to identify falsified transactions. In particular, with hypothesis testing, *GAMER* uses more information, which is obtained from two different models, to make final decisions, leading to greater precision in detecting fraudulent activities regardless of whether they are falsified or not. Deployment results reveal that implementing *GAMER* raised the estimated cost borne by fraudsters by roughly 20%, which corresponded to a 10–20% reduction in daily attack attempts. Regulatory data further confirm that attackers shifted their activities to other payment platforms once the defended platform became more costly to compromise.

V. Robust Fraud Detection with Game-Theoretic Feature Selection

In this section, we detail the design of *GAMER*. We first propose *Game Selection* to obtain the optimal probability of feature selection by solving the equilibrium of a two-player game. Then we present how we can achieve both high accuracy and strong robustness in our proposed *GAMER*.

A. Using Two-Player Game in Feature Selection

Unrestricted falsified transactions aggravate the asymmetries between the detection system and the fraudsters, reducing the effectiveness of local robustness enhancement methods [15]. To counteract the asymmetry in fraud detection, we adopt a proactive two-player-game based strategy to guide the design of our detection method.

Note that, as shown in Fig. 1, incorporating selected features enhances model robustness by increasing resistance to a range of feature falsification strategies in fraud detection. Moreover, if a detection model relies on all available features and adjusts their weights following failed detections, adaptive adversaries may observe these updates and refine their attack strategies accordingly. This feedback loop can drive both sides toward a game-theoretic equilibrium in which neither party has an incentive to further modify its strategy. At this equilibrium, the detection model naturally converges to using a randomly selected subset of features.

1) Game Selection: As illustrated in Fig. 1, the strategies of feature selection and feature falsification interact with each other. This fact motivates us to formulate the detection process as a two-player game and solve for the optimal strategy of feature selection according to the equilibrium of the game, enabling the system to proactively combat intelligent fraudsters.

Specifically, the input $\boldsymbol{X}$ is composed of multiple features, i.e., X_i, $i \in \{1, \cdots, d\}$, hence fraudsters can choose to falsify various combinations of features. Meanwhile, the detection system can select distinct features to accurately detect fraudulent activities. To simplify the analysis, we assume that the feature falsification process and the feature selection process are independent on different features, which makes the sub-games on X_i independent of each other. Therefore, the zero-sum two-player game on X_i is constructed as follows.

TABLE I
Two-player game between detection systems and fraudsters

	Falsify X_i	Do not falsify X_i
Do not select X_i	$U_i^{(ns,a)} - C_i$	$U_i^{(ns,na)}$
Select X_i	$U_i^{(s,a)} - C_i$	$U_i^{(s,na)}$

In Table I, C_i is the cost of falsifying the feature X_i, $U_i^{(ns,a)}$ and $U_i^{(ns,na)}$ are the profits of *Do not select & Falsify* and *Do not select & Do not falsify* on X_i, respectively. In particular, $U_i^{(ns,a)} = U_i^{(ns,na)}$ since if X_i is not selected, falsifying it will not affect the model prediction, which does not change

Algorithm 1 SGD with *Game Selection*

Input: Initial model $\boldsymbol{W}^{(0)}$, the training data $(\boldsymbol{X}, y)$, learning rate η, the profit $\{\Delta U_i\}_{i=1}^d$, the cost $\{C_i\}_{i=1}^d$
Output: Final model $\boldsymbol{W}^{(T)}$

Get optimal probability of feature selection:
$p_i = \min\left(1, \frac{C_i}{\Delta U_i}\right)$, $i \in \{1, \cdots, d\}$
Get multivariate Bernoulli distribution:
$\boldsymbol{m} = (m_1, \cdots, m_d)$, where $m_i \sim \text{Bernoulli}(p_i)$
for $t = 0$ **to** $T-1$ **do**
  Sample $(\boldsymbol{X}^{(t)}, y^{(t)})$ from $(\boldsymbol{X}, y)$
  Sample $\boldsymbol{m}^{(t)}$ from $\boldsymbol{m}$
  $\boldsymbol{W}^{(t+1)} = \boldsymbol{W}^{(t)} - \eta \cdot \nabla_{\boldsymbol{W}} \mathcal{L}(F(\boldsymbol{W}^{(t)}; \boldsymbol{X}^{(t)} \odot \boldsymbol{m}^{(t)}); y^{(t)})$
end for
return $\boldsymbol{W}^{(T)}$

the profits. Additionally, $U_i^{(s,a)}$ and $U_i^{(s,na)}$ are the profits of *Select & Falsify* and *Select & Do not falsify* in regard to X_i, respectively. These variables can be obtained via knowledge acquisition from domain experts [66]. In this work, we assume these variables can be specified with precision.

According to the game in Table I, if the mixed strategy of the detection system is defined as $(1-p_i, p_i)$, then the equilibrium can be obtained by solving the following equation.

$$\begin{aligned}&(1-p_i)(U_i^{(ns,a)} - C_i) + p_i(U_i^{(s,a)} - C_i)\\ &= (1-p_i)U_i^{(ns,na)} + p_i U_i^{(s,na)}.\end{aligned} \tag{2}$$

If X_i is not selected, the falsify X_i will not affect the fraudster's profits. Therefore, by solving Eq. (2), the optimal probability of feature selection is

$$p_i = \min\left[1, \frac{C_i}{U_i^{(s,a)} - U_i^{(s,na)}}\right]. \tag{3}$$

According to Eq. (3), the optimal probability of feature selection is decided by two factors: the cost to falsify X_i, i.e., C_i, and the profit of falsifying X_i when the detection system selects X_i for prediction, i.e., $\Delta U_i = U_i^{(s,a)} - U_i^{(s,na)}$. As displayed in Fig. 4, if $\Delta U_i \leq C_i$, i.e., falsifying X_i is unproductive, the fraudster will not falsify X_i, which implies that the feature X_i is trustworthy. Therefore, the optimal selection probability is 100%. Moreover, when falsifying X_i is productive, i.e., $\Delta U_i > C_i$, the selection rate of X_i is an increasing function of C_i and a decreasing function of ΔU_i.

After getting the optimal probability of feature selection, a Bernoulli random variable m_i is employed to mask the feature X_i, reducing the dependence on X_i if X_i is easily

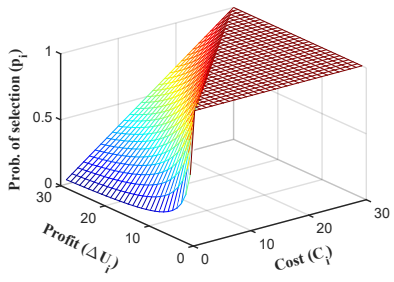


Fig. 4. The optimal probability (i.e., p_i) of selecting X_i is an increasing function of the cost C_i and a decreasing function of the profit ΔU_i.

to be falsified. Specifically, $m_i \sim \text{Bernoulli}(p_i)$, where p_i is calculated by Eq. (3).

In particular, to ensure that data distributions are consistent in the training and testing phases, we employ $\{m_i\}_{i=1}^d$ to change traditional target function in the model training to

$$\min_{\boldsymbol{W}} \mathbb{E}_{\boldsymbol{X},\boldsymbol{m}} \left[\mathcal{L}(F(\boldsymbol{W}; \boldsymbol{X} \odot \boldsymbol{m}); y)\right], \tag{4}$$

where $\boldsymbol{m} = (m_1, \cdots, m_d)$ is the mask vector and $\odot$ represents the Hadamard product. The details of training process are displayed in Algorithm 1 and Fig. 5. In particular, this method can also be applied to the training of gradient boosting decision tree (GBDT), we put these analyses in the appendix due to the space limitation.

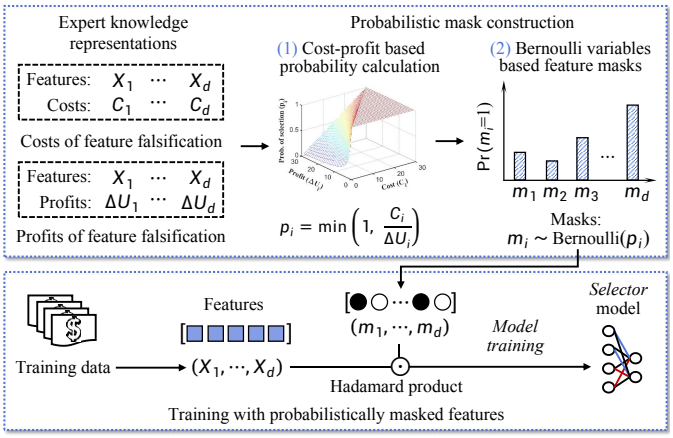
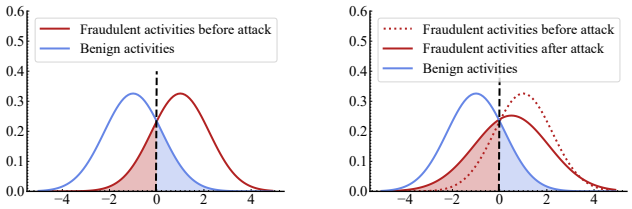


Fig. 5. The training of models equipped with *Game Selection* (i.e., *Selector*).

2) The Advantages of Game Selection: To intuitively illustrate the advantages of *Game Selection*, we provide a toy example of the two-player game based on one-dimensional Gaussian data.

Basic Settings of the Game. The benign and the fraudulent activities follow Gaussian distributions of $\mathcal{N}(-1, 1.5)$ and $\mathcal{N}(1, 1.5)$ respectively. In this case, the optimal classifier is $x = 0$ (i.e., the dashed line in Fig. 6(a)), which achieves the optimal error rate [24].



(a) Before attack (b) After attack

Fig. 6. A toy example to illustrate the advantages of *Game Selection*. Specifically, the benign activities follow $\mathcal{N}(-1, 1.5)$ and fraudsters can falsify the feature to change fraudulent activities from $\mathcal{N}(1, 1.5)$ to $\mathcal{N}(0.5, 2.5)$.

Under these settings, the detection system have two distinct strategies: first, classifying without the training data (e.g., building the classifier with other features), which provides the classifier with error rates 0.21 and 0.25 for benign and

fraudulent activities respectively; second, getting the optimal classifier with the training data, i.e., $x = 0$. With the optimal classifier, the error rates for both benign and fraudulent activities are 0.2071 (i.e., the shaded area in Fig. 6(a)).

Fraudsters also have two strategies: first, falsifying the feature, changing it from $\mathcal{N}(1, 1.5)$ to $\mathcal{N}(0.5, 2.5)$ (as displayed in Fig. 6(b)). In this case, regarding the optimal classifier before the attack, i.e., $x = 0$, the error rate for benign data remains the same but the error rate for fraudulent data increases from 0.2071 to 0.3759. Second, do not falsify the feature, which maintains the error rates for both benign and fraudulent data. In particular, fraudsters consume \$4 to falsify the feature and can obtain \$50 from successfully evading fraud detection, i.e., misclassifying fraudulent data into benign data. In this case, the game can be constructed as Table II.

TABLE II
THE TWO-PLAYER GAME IN THE TOY EXAMPLE

	Falsify	Do not falsify
Do not select	8.5	12.5
Select	14.795	10.355

Specifically, the payoff matrix is calculated as follows.

- NS & F: $\underbrace{50}_{\text{Profit}} * \underbrace{0.25}_{\text{Error rate}} - \underbrace{4}_{\text{Cost}} = 8.5$
- NS & NF: $\underbrace{50}_{\text{Profit}} * \underbrace{0.25}_{\text{Error rate}} = 12.5$
- S & F: $\underbrace{50}_{\text{Profit}} * \underbrace{0.3759}_{\text{Error rate}} - \underbrace{4}_{\text{Cost}} = 14.795$
- NS & NF: $\underbrace{50}_{\text{Profit}} * \underbrace{0.2071}_{\text{Error rate}} = 10.355$

With the payoff matrix, we can get the mixed strategy equilibrium {*Detection system: (0.5485, 0.4515), Fraudster: (0.2556, 0.7444)*}, which means the probability 0.4515 is the optimal probability for the detection system to select the feature. In this case, the system and the fraudster have no incentives to change their strategies since the mixed strategy maximizes their profits. The optimal profit for the fraudster is \$11.342 (i.e., the optimal profit for the system is $-$\$11.342).

In addition to *Game Selection*, the detection system can also employ adversarial training to enhance the model's robustness. In this scenario, the system can get the optimal classifier after the falsification (i.e., $x = -0.0324$ in Fig. 6(b)) by utilizing the data after being falsified. In this case, the error rates for benign and fraudulent data are 0.2148 and 0.3682, respectively. Moreover, the fraudsters' profit is \$14.405.

As illustrated in Table III, compared to other methods, *Game Selection* reduces the fraudsters' profit from \$14.795 to \$11.342, thereby reducing the incentive of fraudsters to falsify the features. Moreover, *Game Selection* reduces the error rate by 33.47% in fraud detection and causes a 74.44% reduction in committing falsification, exhibiting deterrence to fraudsters.

B. Identifying falsification behaviors

As indicated in Table III, the error rate of *Game Selection* is slightly higher than the non-robust model in detecting the clean examples. To overcome this limitation, we employ hypothesis testing to identify falsification behaviors in fraud detection, thereby using different models, e.g., model with *Game Selection* or non-robust model, to detect the transactions they expert in. Combining it with *Game Selection*, we design *GAMER*, a fraud detection system that achieves both high accuracy on unfalsified transactions and strong robustness on falsified transactions.

TABLE III
COMPARISONS IN THE TOY EXAMPLE[1]

	Profit	Benign error rate	Fraudulent error rate	Prob. of being falsified by fraudsters
Non-robust Training	$14.795_{(-)}$	$20.71\%_{(-)}$	$37.59\%_{(-)}$	$100\%_{(-)}$
Adversarial Training	$14.405_{(\downarrow)}$	$21.48\%_{(\uparrow)}$	$36.82\%_{(\downarrow)}$	$100\%_{(-)}$
Game Selection	$11.342_{(\Downarrow)}$	$20.87\%_{(\uparrow)}$	$25.01\%_{(\Downarrow)}$	$25.56\%_{(\Downarrow)}$

[1] The double arrow means the decrease is larger than 10%.

As the predictions of the robust model and the non-robust model are significantly different on falsified transactions.[6], we can use this difference to identify falsification behaviors. Moreover, we employ the *Game Selection* strategy in the robust model, which maintains the deterrent effect on fraudsters.

Hypothesis Testing. As illustrated in Fig. 3, two models (i.e., *Completer* and *Selector*) are trained with the classic non-robust method (e.g., naive SGD) and *Game Selection*, respectively. Then *GAMER* employs hypothesis testing to decide the output prediction. In hypothesis testing, *GAMER* utilizes the training data to construct the statistic of $\kappa = \|y\_score_{Selector} - y\_score_{Completer}\|_1$, i.e., the difference in predicted probability between *Selector* and *Completer*, and sets the null hypothesis for each transaction as H_0 : *it is an unfalsified transaction*. Finally, *GAMER* chooses a significance level α (e.g., the frequently used 5%) for hypothesis testing and calculates the quantile (i.e., Δ_α) of the rejection region [73].

After preparations, *GAMER* decides the output of the system by hypothesis testing. If H_0 is rejected (i.e., $\kappa > \Delta_\alpha$), indicating that the transaction is falsified, *GAMER* outputs *Selector* prediction. Otherwise, *GAMER* outputs *Completer* prediction. Moreover, if *Completer* prediction is $\hat{y}_{Completer} = 1$, *GAMER* outputs $\hat{y}_{Completer}$ since intelligent fraudsters will certainly not falsify features to cause them to be detected.

VI. Theoretical Analysis

In this section, we employ causal analysis to explain the cause of misclassification (i.e., the reasons why the predicted label is different from the real label) in fraud detection. Based on this cause, we demonstrate that *GAMER* enhances the robustness of the detection model.

A. Causal model of Fraud Detection

To systematically reveal the reasons for the misclassification in fraud detection, a causal diagram is constructed to model the

[6]The difference is the reason why the robust model is more accurate than the non-robust model on falsified transactions.

entire life cycle of ML-based fraud detection system. Based on the causal diagram, we use causal analysis to explain the reasons for fraud detection evasion.

1) Causal Diagram Construction: As illustrated in Fig. 7, the workflow of an ML-based fraud detection system is four-fold, including *training data collection*, *model training*, *model prediction after deployment*, and *prediction based decision making*. The former two steps aim at preparing the detection model (i.e., working during the training phase), and the latter two steps work after model deployment (i.e., working during the testing phase).

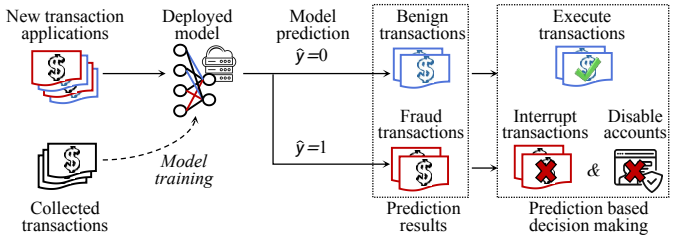


Fig. 7. The workflow of an ML-based fraud detection system.

In particular, *prediction based decision making process* is the motivation of fraudsters to falsify the transactions since fraudsters are intelligent and profit-sensitive. Specifically, the falsification process consumes resources [42], e.g., falsifying the feature of *Registered Capital* costs money in reality. If the profits cannot cover the falsification cost, fraudsters will not falsify features because it causes economic loss. The *prediction based decision making process* results in differences in the profits of the fraudsters: fraudulent activities are successful when the model identifies them as benign, and fraudsters can get illegal profits from the fraudulent activities; otherwise, fraudsters obtain nothing since the transaction is interrupted. As a result, the *prediction based decision making process* leads the model prediction to be the cause of feature falsification (i.e., to obtain illegal profits from fraudulent activities).

As a result, the entire life cycle of fraud detection can be explained as the process in Fig. 8. The fraud detection system aims to distinguish fraudulent activities from benign activities, hence collecting the training data and training the detection model. Moreover, the fraudsters aim to evade the detection system by falsifying input features to obtain illegal profits from fraudulent activities.

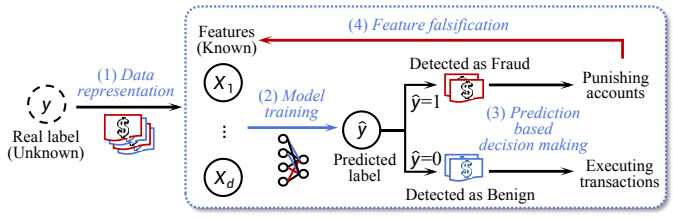


Fig. 8. The entire life cycle of fraud detection.

There are three connections in Fig. 8: first, the *data representation process* (i.e., the black arrow on the left of Fig. 8). In practice, the real label of a transaction (i.e., y) is unknown, we can only collect data with certain features $\boldsymbol{X} = (X_1, \cdots, X_d)$ to get information of various transactions, e.g., we collect the features of account balance, registered capital, etc., to describe transactions. Moreover, y results in the variation of features, since it decides the behaviors in the transaction, e.g., if transactions are collected from fraudulent activities, their features must reveal the harmfulness of fraudulence.

Second, the *model training process* (i.e., the blue arrow in Fig. 8). With the collected data, the fraud detection system trains a detection model to model the correlations between the features and the label. The model can be a neural network, a decision tree, etc. After training, the model is deployed to detect real-world fraudulent activities. This process indicates that the training process establishes the relationship between the features of $\boldsymbol{X}$ and the predicted label $\hat{y}$.

Third, *feature falsification process* (i.e., the red arrow in Fig. 8). If the transaction is recognized as fraudulent, it will be interrupted and the account will be disabled through *prediction based decision making process*. These punishments motivate fraudsters to falsify features to evade the detection system, which escapes from the punishments and obtains profits from fraudulent activities. As a result, model prediction (i.e., $\hat{y}$) affects input data $\boldsymbol{X}$, i.e., fraudsters will consume resources to falsify features if their transactions are identified as fraudulent.

To formalize the relationship in Fig. 8, we construct a causal diagram in Fig. 9. In this causal diagram, we separate the data $\boldsymbol{X}^T$ into $\boldsymbol{X}^t$ and $\boldsymbol{X}^{t+1}$ to distinguish the falsified data $\boldsymbol{X}^{t+1}$ after deployment from the training data $\boldsymbol{X}^t$. This separation removes the cyclic connections [17], [1].

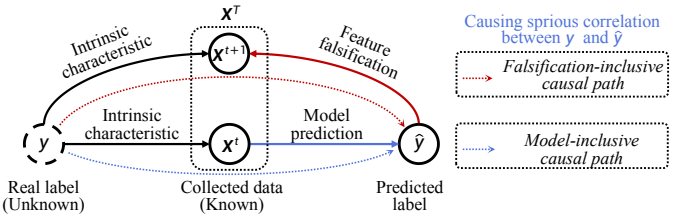


Fig. 9. The causal diagram of the fraud detection, revealing the cause of misclassification.

As indicated in the causal diagram, y is the cause of the training data $\boldsymbol{X}^t$, since it decides the intrinsic characteristic of the data. For example, the fraud transaction (i.e., $y = 1$) must reveal the harmfulness of fraudulence, and this harmfulness is characterized by features. Meanwhile, y is the cause of falsified data $\boldsymbol{X}^{t+1}$. That is, the falsification process is restricted if y is decided, which means that some features cannot be falsified. Otherwise, y will be changed (e.g., fraud activity changes to benign activity due to the regulated behavior).

Moreover, training data $\boldsymbol{X}^t$ is the cause of model prediction, i.e., $\hat{y}$. Specifically, in ML-based fraud detection system, a detection model is trained with the data (i.e., $\boldsymbol{X}^t$) to establish the relationship between input features and their labels.

Finally, the prediction $\hat{y}$ is the cause of falsified data $\boldsymbol{X}^{t+1}$ after model deployment. The reason is that fraudsters have

incentives to falsify the data to get more profits when the data is identified as fraudulent (i.e., $\hat{y} = 1$). If the data is identified as benign (i.e., $\hat{y} = 0$), fraudsters will not consume resources to falsify features due to the optimality of the result, i.e., fraudsters can obtain profits from fraud activities. Table VII in the Appendix provides detailed case analyses.

2) Causal Analysis of the Misclassification in Fraud Detection: As indicated in Fig 9, the causal diagram consists of two basic structures: the collider structure (i.e., the falsification-inclusive causal path) and the chain structure (i.e., the model-inclusive causal path). These structures reveal different characteristics in modeling the relationship between y and $\hat{y}$.

According to causal analysis, the cases are two-fold: $\boldsymbol{X}^{t+1}=\boldsymbol{X}^{t}$[7] or $\boldsymbol{X}^{t+1}\neq\boldsymbol{X}^{t}$. First, if $\boldsymbol{X}^{t+1}=\boldsymbol{X}^{t}$, either $\boldsymbol{X}^{t+1}$ or $\boldsymbol{X}^{t}$ represents the complete data distribution. That is, the data in the training phase or after falsification is not conditioned on the timestamp, i.e., $\boldsymbol{X}^{T}=\boldsymbol{X}^{t+1}=\boldsymbol{X}^{t}$. Based on properties explained in section II, the unconditioned data blocks the collider structure (i.e., the falsification-inclusive causal path), hence the falsification process cannot affect the model inference. Moreover, the unconditioned data maintains the chain structure (i.e., the model-inclusive causal path), which means that the relationship between y and $\hat{y}$ is a causal relationship.

Second, if $\boldsymbol{X}^{t+1}\neq\boldsymbol{X}^{t}$, which means $\boldsymbol{X}^{t+1}$ and $\boldsymbol{X}^{t}$ follow different distributions. In other words, fraudsters falsify the input data to make it fall into the uncovered area of the training data [7]. In this case, both of the falsified data $\boldsymbol{X}^{t+1}$ and the training data $\boldsymbol{X}^{t}$ are conditioned on the timestamp, i.e., $\boldsymbol{X}^{t+1} = \boldsymbol{X}^{T} \mid T = t+1$ and $\boldsymbol{X}^{t} = \boldsymbol{X}^{T} \mid T = t$. According to the properties of the basic structures in causal diagrams, the conditioned data blocks the chain structure and opens up the collider structure, which means only the falsification-inclusive causal path is connected. The connected falsification-inclusive causal path makes y and $\hat{y}$ spuriously correlated with each other. This spurious correlation originates from selection bias (i.e., insufficient training data), allowing fraudsters to craft spurious correlation $(y = 1) \rightarrow (\hat{y} = 0)$ through feature falsification.

B. Robustness Enhancement of Game Selection

As selection bias causes misclassification, the reason for selection bias becomes vitally important. Selection bias arises from the procedure by which data are selected in model training [27], which is related to the idea of data coverage [7], [43]. For instance, if the falsified data is not selected in the training process (i.e., the falsification process causes distribution shifts), it can cause misclassification more easily.

For data coverage, we make the following definition.

Definition 1 (Data coverage of a data point)**.** *For a specific classifier $F(\boldsymbol{W};\cdot)$, the data coverage of a data point $\boldsymbol{x}$ is defined as a d-dimensional cube with side length as follows.*

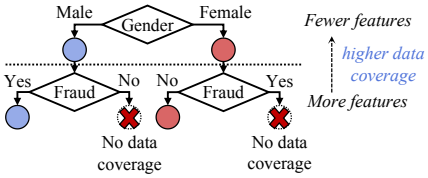
$$\epsilon = \min_{F(\boldsymbol{W};\boldsymbol{x}^*)\neq F(\boldsymbol{W};\boldsymbol{x}) \text{ or } y_{\boldsymbol{x}^*}\neq y_{\boldsymbol{x}}} \|\boldsymbol{x}^* - \boldsymbol{x}\|_{-\infty}, \tag{5}$$

where $y_{\boldsymbol{x}}$ represents the real label of $\boldsymbol{x}$.

[7] $\boldsymbol{X}^{t+1}=\boldsymbol{X}^{t}$ means the falsified data and the training data follow the identical data distribution.

ID	Gender	Fraud
1	Male	Yes
2	Female	No

(a) A sample dataset



(b) The corresponding feature representations

Fig. 10. Using fewer features increases data coverage.

In particular, $F(\boldsymbol{W};\boldsymbol{x}^*) \neq F(\boldsymbol{W};\boldsymbol{x})$ is related to the model smoothness, and $y_{\boldsymbol{x}^*} \neq y_{\boldsymbol{x}}$ is related to the ground truth, both of which affect feature falsification.

According to Definition 1, we have the following theorem.

Theorem 1 (Maximum data coverage)**.** *Denote the volume of the input space as Vol, the data coverage (i.e., γ) of a dataset with N data points is upper-bouned by*

$$\gamma \leq N \cdot \left[\frac{\epsilon}{\sqrt[d]{Vol}}\right]^{d}, \tag{6}$$

where d is the number of selected features.

Theorem 1 is the consequence of the summation of N cubes, the inequality depends on the possible overlap of cubes (i.e., the data coverage of a data point).

If falsification processes are unrestricted, the volume of the input space (i.e., Vol) becomes extremely large, which means $\epsilon \ll \sqrt[d]{Vol}$. According to Theorem 1, the data coverage exponentially decreases with the increase of feature dimension (i.e., d), indicating that using fewer features increases the data coverage (as shown in Fig. 10). In particular, as d is the exponent, using fewer features is more efficient in dealing with unrestricted perturbed fraud transactions, which is consistent with the fact that feature selection is the dominant method of dealing with selection bias [38].

As *Game Selection* reduces the expected number of features used in model prediction, it enhances the model robustness by enlarging the data coverage. Specifically, as indicated in Fig. 5, *Game Selection* masks each feature X_i with an independent variable $m_i \sim \text{Bernoulli}(p_i)$, which means that X_i is selected to use with probability p_i. Therefore, the expected number of features used in *Game Selection* is

$$\mathbb{E}(\sum_{i=1}^{d} m_i) = \sum_{i=1}^{d} p_i \ll d, \tag{7}$$

demonstrating that *GAMER* uses fewer features for detection, thereby enhancing the robustness of the detection system.

VII. Experimental Evaluation

In this section, we aim to validate our design and demonstrate that the proposed system is effective at detecting real-world fraud activities regardless of fraudsters commit falsifications or not. All experiments are performed upon a Supermicro SYS-420GP-TNR server with two Intel(R) Xeon(R) Gold 6348 CPUs (2 × 28 cores), Ubuntu 18.04.1, 10GB memory, and four NVIDIA A100 PCIe 80GB GPUs.

Datasets. We first use three real-world fraud detection datasets to perform the simulated experiments, including TabFormer [51], CreditCard [35], and IEEE-CIS [36]. Moreover, we validate our design with the real-world data collected from the world's leading online payment enterprise, which provides real-world transactions for evaluation. We send the details of these datasets to the Appendix due to the space limitation. Moreover, we employ the techniques of SMOTE [13] and random under-sampling to overcome the limitations of highly unbalanced data.

Models. We evaluate our design with three ML models commonly applied to tabular data. First, *MLP*, a model with two hidden layers of 64 and 32 neurons respectively. The activation function of MLP is Relu. Second, *TabNet* [5], a transformer-based neural network specialized in tabular data. Third, *LightGBM* [37], an efficient tree-based model used to process tabular data.

A. Simulated Experiments

In this section, we design experiments to validate the roles of feature selection and compare *GAMER* with traditional techniques used in robustness enhancement.

The Roles of Feature Selection. As indicated in Fig. 1, feature selection enables detection systems to affect fraudsters. To validate this interaction, we employ a probabilistic fraudster to attack the MLP model (i.e., falsifying features according to the model) on the CreditCard dataset. The fraudster utilizes a probabilistic strategy to attack the model with probability $\boldsymbol{p}_{at.}$[8]. If the fraudster falsify the feature, the Gaussian noise $\mathcal{N}(4,1)$ is added to the feature, otherwise the feature remains unchanged. We utilize interpolation to investigate the impact of feature selection. Specifically, the selection rate in Algorithm 1 is set to $\boldsymbol{p} = 1-[\lambda\cdot\boldsymbol{p}_{at.}+(1-\lambda)\cdot\mathbf{0}]$ ($\mathbf{0}$ means training without feature selection). The results displayed in Fig. 11 indicate that the AUC and F1 score increase consistently as λ increases, which means that the system detects more accurately if the selection probability is closer to the optimum (i.e., $\lambda = 1$).

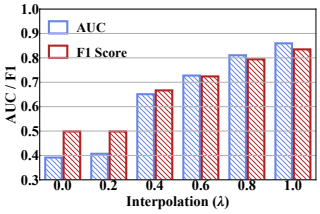


Fig. 11. Selection probabilities affects the model performance.

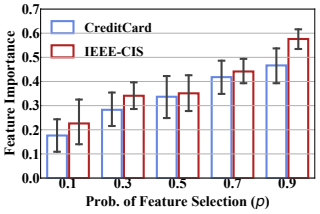


Fig. 12. Selection probabilities affects the feature importance.

Moreover, we use TabNet to investigate the rationale of feature selection since it can estimate the feature importance in model prediction. For this target, we train the TabNet with different selection probability (i.e., $p \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$ for a feature set that consists of the first two-thirds of all features) and calculate the feature importance of this feature set (i.e., the summation of the importance over all features in the feature set). The results in Fig. 12 indicate that increasing the selection probability increases the feature importance in the prediction process. This is the reason that feature selection can prevent feature falsification because it reduces the importance of the feature if it is more easily to be falsified.

Compared with Existing Methods. We utilize a cost-aware fraudster (as illustrated in Algorithm 3 in the Appendix) to simulate real-world fraudulent activities [42] and compare *GAMER* with the classical methods. We employ an MLP in these experiments to ensure a fair and consistent comparison across all methods. The data were normalized using standard normalization, a widely adopted preprocessing technique for tabular data. This procedure stabilizes model training and scales feature magnitudes to a comparable range, typically within (0, 1). Thus, perturbations applied to tabular inputs become semantically meaningful when evaluated under classical vision-style attacks. Moreover, we randomly set C_i, which follows a discrete uniform distribution (i.e., $C_i \sim \mathcal{U}(1, 31)$), to X_i and repeat these experiments 30 times to reduce the bias introduced by cost assignment.

For *GAMER*, we set the cost identical to fraudster's cost and set the profit as $\Delta U_i = \sqrt{At.Cost \cdot C_i}$, where $At.Cost$ represents the attack cost bound of the fraudster. The reasons for this profit setting are two-fold: first, intelligent fraudsters launch attacks if the total profit is larger than $At.Cost$ and they will enlarge $At.Cost$ if the profit is larger than $At.Cost$ to raise the successful probability, hence the total profit is identical to $At.Cost$; second, the profit of falsifying X_i can be formalized as $At.Cost \cdot \frac{C_i}{\Delta U_i} = \Delta U_i$, which indicates that the fraudster's expected profit equals the real profit. Notably, these experiments also reveal that taking cost-profit knowledge into account is beneficial in fraud detection.

First, we aim to validate that utilizing fewer features enhances detection robustness (as illustrated in Fig. 10). In these experiments, we use naive SGD to train non-robust MLP models and use the cost-aware falsification process with various cost bounds to attack the models. In particular, the normalization preprocess is not used and the sample sizes of all datasets are similar to each other. As illustrated in Fig. 13, the model trained on the dataset with fewer features (i.e., TabFormer) reveals stronger robustness, the AUC and F1 score decrease when the cost bound grows significantly large, which are consistent with our conclusion. As the model on the Tabformer dataset reveals strong robustness due to the small number of features, we utilize the other two datasets (i.e., CreditCard and IEEE-CIS) to investigate the performance of robustness enhancement.

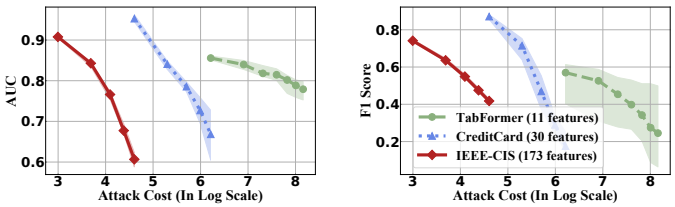


Fig. 13. Using fewer features enhances model robustness.

[8]We randomly assign the probability for falsifying features with $\mathcal{U}(0,1)$ to eliminate the bias incurred by probability assignation.

TABLE IV
COMPARISONS OVER THE COST-AWARE (I.E., RATIONAL) FRAUDSTERS[1]

Dataset	Method \ At. Cost	AUC					F1 Score				
		100	200	300	400	500	100	200	300	400	500
CreditCard	Non-robust	$0.9629_{(-)}$	$0.8547_{(-)}$	$0.7883_{(-)}$	$0.7199_{(-)}$	$0.6609_{(-)}$	$0.8729_{(-)}$	$0.7504_{(-)}$	$0.5483_{(-)}$	$0.3712_{(-)}$	$0.2827_{(-)}$
	CW [12]	$\mathbf{0.9970}_{(\uparrow)}$	$0.9566_{(\uparrow)}$	$0.8914_{(\uparrow)}$	$0.8269_{(\uparrow)}$	$0.7651_{(\uparrow)}$	$0.9107_{(\uparrow)}$	$0.8802_{(\uparrow)}$	$0.8476_{(\Uparrow)}$	$0.7937_{(\Uparrow)}$	$0.7342_{(\Uparrow)}$
	TARDES [85]	$0.9042_{(\downarrow)}$	$0.8748_{(\uparrow)}$	$0.8436_{(\uparrow)}$	$0.8178_{(\uparrow)}$	$0.8025_{(\Uparrow)}$	$0.7565_{(\downarrow)}$	$0.7323_{(\downarrow)}$	$0.7163_{(\uparrow)}$	$0.7077_{(\Uparrow)}$	$0.7041_{(\Uparrow)}$
	PGD [45]	$0.9265_{(\downarrow)}$	$0.8995_{(\uparrow)}$	$0.8746_{(\uparrow)}$	$0.8493_{(\uparrow)}$	$0.8306_{(\Uparrow)}$	$0.7467_{(\Downarrow)}$	$0.7286_{(\Downarrow)}$	$0.7150_{(\Downarrow)}$	$0.7032_{(\Downarrow)}$	$0.6955_{(\Downarrow)}$
	FGSM [25]	$0.9297_{(\downarrow)}$	$0.8940_{(\uparrow)}$	$0.8625_{(\uparrow)}$	$0.8482_{(\uparrow)}$	$0.8405_{(\Uparrow)}$	$0.8590_{(\downarrow)}$	$0.8452_{(\uparrow)}$	$0.8365_{(\Uparrow)}$	$0.8335_{(\Uparrow)}$	$0.8317_{(\Uparrow)}$
	GAMER	$0.9828_{(\uparrow)}$	$\mathbf{0.9584}_{(\uparrow)}$	$\mathbf{0.9402}_{(\Uparrow)}$	$\mathbf{0.9247}_{(\Uparrow)}$	$\mathbf{0.9164}_{(\Uparrow)}$	$\mathbf{0.9403}_{(\uparrow)}$	$\mathbf{0.9115}_{(\Uparrow)}$	$\mathbf{0.8954}_{(\Uparrow)}$	$\mathbf{0.8785}_{(\Uparrow)}$	$\mathbf{0.8687}_{(\Uparrow)}$
	Method \ At. Cost	20	40	60	80	100	20	40	60	80	100
IEEE-CIS	Non-robust	$0.9077_{(-)}$	$0.8411_{(-)}$	$0.7632_{(-)}$	$0.6818_{(-)}$	$0.6055_{(-)}$	$0.7414_{(-)}$	$0.6375_{(-)}$	$0.5485_{(-)}$	$0.4767_{(-)}$	$0.4187_{(-)}$
	CW [12]	$0.8379_{(\downarrow)}$	$0.8247_{(\downarrow)}$	$0.8103_{(\uparrow)}$	$0.7945_{(\uparrow)}$	$0.7808_{(\Uparrow)}$	$0.5945_{(\downarrow)}$	$0.5808_{(\downarrow)}$	$0.5662_{(\uparrow)}$	$0.5530_{(\uparrow)}$	$0.5416_{(\Uparrow)}$
	TARDES [85]	$0.7221_{(\Downarrow)}$	$0.7128_{(\downarrow)}$	$0.7038_{(\downarrow)}$	$0.6939_{(\uparrow)}$	$0.6839_{(\uparrow)}$	$0.5865_{(\downarrow)}$	$0.5806_{(\downarrow)}$	$0.5748_{(\uparrow)}$	$0.5685_{(\uparrow)}$	$0.5616_{(\Uparrow)}$
	PGD [45]	$0.7610_{(\downarrow)}$	$0.7519_{(\downarrow)}$	$0.7428_{(\downarrow)}$	$0.7338_{(\uparrow)}$	$0.7251_{(\uparrow)}$	$0.5933_{(\downarrow)}$	$0.5855_{(\downarrow)}$	$0.5784_{(\uparrow)}$	$0.5712_{(\uparrow)}$	$0.5651_{(\Uparrow)}$
	FGSM [25]	$0.6019_{(\Downarrow)}$	$0.5879_{(\Downarrow)}$	$0.5744_{(\Downarrow)}$	$0.5623_{(\downarrow)}$	$0.5493_{(\downarrow)}$	$0.5103_{(\Downarrow)}$	$0.4984_{(\Downarrow)}$	$0.4868_{(\downarrow)}$	$0.4759_{(\downarrow)}$	$0.4640_{(\uparrow)}$
	GAMER	$\mathbf{0.9151}_{(\uparrow)}$	$\mathbf{0.8842}_{(\uparrow)}$	$\mathbf{0.8585}_{(\uparrow)}$	$\mathbf{0.8393}_{(\Uparrow)}$	$\mathbf{0.8210}_{(\Uparrow)}$	$\mathbf{0.7889}_{(\uparrow)}$	$\mathbf{0.7500}_{(\uparrow)}$	$\mathbf{0.7183}_{(\Uparrow)}$	$\mathbf{0.6942}_{(\Uparrow)}$	$\mathbf{0.6717}_{(\Uparrow)}$

[1] The double arrow means the improvement (or the decrease) is larger than 20%.

TABLE V
COMPARISONS OVER THE IRRATIONAL FRAUDSTERS

Attack method	Without attack	AutoAttack [19]			Square attack [2] (Black-box attack)		
Def. method \ Metric	F1	F1	ASR	$\frac{At.\ Cost}{ASR}$	F1	ASR	$\frac{At.\ Cost}{ASR}$
Non-robust	$\mathbf{0.8298}_{(-)}$	$0.0323_{(-)}$	$0.9812_{(-)}$	$11.55_{(-)}$	$0.0446_{(-)}$	$0.9738_{(-)}$	$11.58_{(-)}$
CW [12]	$0.6090_{(\downarrow 26.6\%)}$	$0.3743_{(\uparrow)}$	$0.7210_{(\downarrow 26.5\%)}$	$4.75_{(\downarrow 58.9\%)}$	$0.4142_{(\uparrow)}$	$0.6964_{(\downarrow 28.5\%)}$	$3.79_{(\downarrow 67.3\%)}$
TARDES [85]	$0.5922_{(\downarrow 28.6\%)}$	$0.4749_{(\uparrow)}$	$0.4891_{(\downarrow 50.1\%)}$	$4.34_{(\downarrow 62.4\%)}$	$0.4796_{(\uparrow)}$	$0.4751_{(\downarrow 51.2\%)}$	$4.41_{(\downarrow 61.9\%)}$
PGD [45]	$0.6006_{(\downarrow 27.6\%)}$	$0.4678_{(\uparrow)}$	$0.5442_{(\downarrow 44.5\%)}$	$4.50_{(\downarrow 61.0\%)}$	$0.4677_{(\uparrow)}$	$0.5381_{(\downarrow 44.8\%)}$	$4.76_{(\downarrow 58.8\%)}$
FGSM [25]	$0.5234_{(\downarrow 36.9\%)}$	$0.0323_{(-)}$	$0.9670_{(\downarrow 1.44\%)}$	$9.77_{(\downarrow 15.4\%)}$	$0.0367_{(\downarrow)}$	$0.9609_{(\downarrow 1.32\%)}$	$10.20_{(\downarrow 11.9\%)}$
GAMER	$0.8012_{(\downarrow 3.45\%)}$	$\mathbf{0.6356}_{(\uparrow)}$	$\mathbf{0.4145}_{(\downarrow 57.8\%)}$	$\mathbf{23.26}_{(\uparrow 101.4\%)}$	$\mathbf{0.7308}_{(\uparrow)}$	$\mathbf{0.2826}_{(\downarrow 70.9\%)}$	$\mathbf{34.67}_{(\uparrow 199.4\%)}$

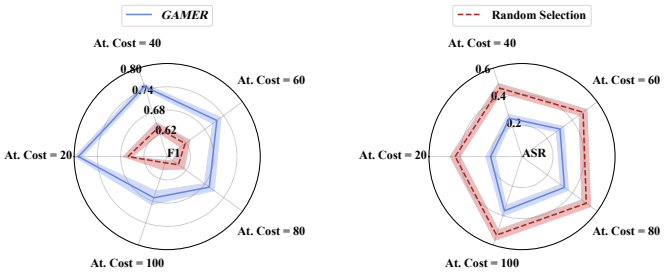


Fig. 14. The detection performance improvement achieved by *GAMER*.

To illustrate the advantages of incorporating cost-profit knowledge in detail, we compare *GAMER* with random selection, which randomly selects each feature with the probability of 50% (i.e., random guess without expert knowledge). The median results of the 30 times experiments shown in Fig. 14 indicate that the use of cost-profit knowledge in *GAMER* increases the F1 score by up to 16.43% and reduces the attack success rate (ASR) by up to 30.89%. These results demonstrate that calculating equilibrium-based selection probabilities based on cost-profit knowledge is beneficial for robustness enhancement.

Evaluate the Robustness Enhancement over Rational Fraudsters. With the cost-aware attack, we compare *GAMER* with classical adversarial training techniques, including CW [12], TRADES [85], PGD [45], and FGSM [25]. We use torchattacks 3.5.1 [39] to generate these adversarial examples. For all of these techniques, we utilize the radius of the neighborhood as $L_\infty = 50/255$, which achieves the best performance in our experiments, and set the number of optimization steps as 100 (which is identical to the attack steps). In the experimental results displayed in Table IV, we observe that the performance of *GAMER* is higher than adversarial training, implying that *GAMER* is effective in extracting robust features to improve detecting performance. Additionally, as the attack cost increases, which means that the attack ability is enhanced, *GAMER* exhibits stronger robustness. In particular, when the attack cost is 500, *GAMER* at least increases the AUC by 5.15% and the F1 score by 19.07% on IEEE-CIS. These results validate that *GAMER* is an effective method to enhance the robustness of the model in fraud detection.

Evaluate Robustness Enhancement over Irrational Fraudsters. We also utilize AutoAttack [19] and Square Attack [2] to evaluate the performance of *GAMER* against irrational fraudsters (on the IEEE-CIS dataset) because these fraudsters aim at raising ASR without considering cost-profit information. The radius of AutoAttack is set to $L_\infty = 50/255$, which is identical to the defense methods. Additionally, the number of queries in the Square attack is set to 1000. As displayed

in Table V, although the performance of *GAMER* is slightly weaker than the non-robust model when there is no falsification in transactions, it achieves a much higher F1 score than traditional methods in robustness enhancement . This result is reasonable since *GAMER* slightly loses useful information (i.e., uses fewer features in detection) when there is no attack in transactions. Furthermore, *GAMER* outperforms traditional methods against irrational fraudsters, increasing the F1 score and reducing ASR under both ensemble attacks and black-box attacks. In particular, as $\frac{At.\ Cost}{ASR}$ indicates the average cost of increasing 1% ASR, the experimental results demonstrate that *GAMER* significantly increases the cost of successful attacks by up to 199.4%, resulting in more economic losses for irrational fraudsters.

Sensitivity Analysis. To investigate the sensitivity of *GAMER* to cost misspecification, we introduce randomized perturbations to the true costs during training on the IEEE-CIS dataset, thereby generating a wide range of misspecification scenarios. Specifically, the model is trained using perturbed costs of the form: $\widetilde{C}_i = C_i + \mathcal{U}(-C_i, (\sqrt{12}\rho - 1) \cdot C_i)$, where $\rho = \frac{\sqrt{Var(\widetilde{C}_i - C_i)}}{C_i}$ controls the expected scale of perturbation. To reduce randomness-induced variance, each experimental setting is repeated 30 times. The results (see Fig. 15) demonstrate that *GAMER* exhibits strong robustness under cost misspecification. Even when the costs are perturbed by up to 300% of their true values, the average F1-score decreases by only 6.5%.

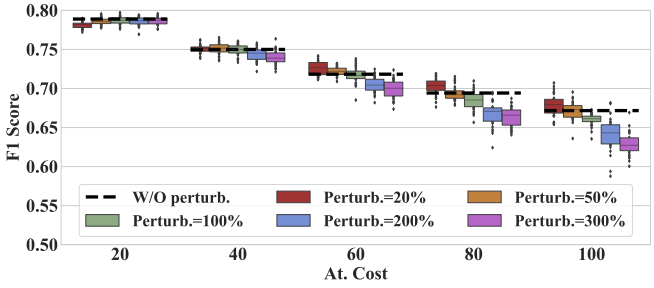


Fig. 15. Sensitivity analysis on cost misspecification.

Additionally, we apply similar perturbation settings to the profit values to assess whether $\Delta U_i = \sqrt{At.Cost \cdot C_i}$ serves as a reasonable profit approximation. As shown in Fig. 16, using the profit formulation $\sqrt{At.Cost \cdot C_i}$ achieves detection performance that is close to the optimal configuration across experiments. This result indicates that the proposed profit estimation provides a sufficiently accurate and practical approximation of the true profit.

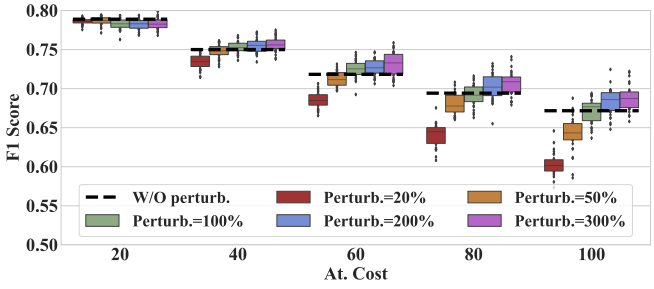


Fig. 16. Validation of the profit estimation.

TABLE VI
COMPARISONS OVER REAL-WORLD TRANSACTIONS

Method \ Metric	TabNet AUC	TabNet F1	LightGBM AUC	LightGBM F1
Non-robust	$0.751_{(-)}$	$0.550_{(-)}$	$0.780_{(-)}$	$0.534_{(-)}$
CW	$0.756_{\uparrow 0.7\%}$	$0.583_{\uparrow 6.0\%}$	—[1]	—
TARDES	$0.594_{\downarrow 20.9\%}$	$0.537_{\downarrow 2.4\%}$	—	—
PGD	$0.650_{\downarrow 13.4\%}$	$0.544_{\downarrow 1.1\%}$	—	—
FGSM	$0.554_{\downarrow 26.2\%}$	$0.516_{\downarrow 6.2\%}$	—	—
GAMER	$0.790_{\uparrow 5.2\%}$	$0.683_{\uparrow 24.2\%}$	$0.808_{\uparrow 3.6\%}$	$0.666_{\uparrow 24.7\%}$

[1] Gradient based techniques are unsuitable for tree based models [14].

B. Real-world Experiments

In addition to the simulated experiments, we also validate *GAMER* on real-world transactions. The data is collected from the world's leading online payment enterprise. Specifically, the training data are composed of the transactions collected from Oct. 7 to Nov. 26, 2022. Moreover, the test data consist of the transactions collected from Nov. 28, 2022 to Jan. 28, 2023. The labels of these data are collected from whether the transactions are complained about by users. It is worth noting that after Jan. 1, 2023, the fraud detection system deployed in the enterprise experienced precipitately exacerbated attacks, which doubled the asset loss rate of the enterprise in that month. The analysis of the risk control department in the enterprise indicates that fraudsters have discovered flaws in the detection system and then falsified their transactions accordingly to evade the detection system.

Before model training, we collect the cost-profit knowledge for falsifying each feature from a group of security experts in the risk control department of the enterprise. Specifically, the experts divide the features into three categories: high-cost features, medium-cost features, and low-cost features. The security experts fix the profit[9] of successfully evade the detection system as $\$1 \times 10^2$ and carefully set the relative costs of $\$81$, $\$72.25$, $\$42.25$ to these features, respectively (we send the details of the process to the Appendix). Then with these profits and costs, we calculate the optimal probability of feature selection according to Eq. (3) as 0.9, 0.85, 0.65 for the high-cost features, the medium-cost features, and the low-cost features, respectively.

With the optimal probability of feature selection, we train the models in *GAMER* with TabNet and LightGBM on the real-world dataset. In these experiments, we compare *GAMER* with adversarial training techniques, the step number and the radius are set as 10 and $50/255$. The results in Table VI indicate *GAMER* can be applied to various model training and achieves the best performance improvement, increasing the AUC and F1 score by 4.4% and 24.45% respectively on average.

Ablation Evaluation Using Real-world Data. We use LightGBM as the model and compare the performance of *GAMER*, *Completer*, and *Selector* on real-world dataset. As displayed in Fig. 17, the F1 score of *GAMER* is higher than both *Selector*

[9]The selection rate depends on the ratio between the cost and the profit, hence experts only need to decide the relative value of these variables.

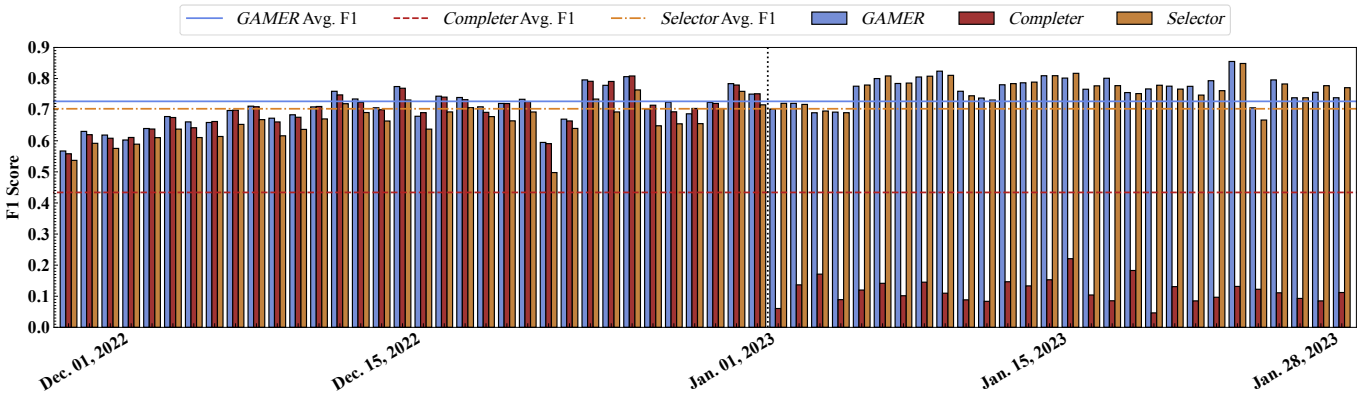


Fig. 17. Daily evaluation on real-world fraud detection (F1 Score).

and *Completer* in the first 34 days. The reason is that there are minor falsified transactions before Jan. 1, 2023. The detection mechanism in *GAMER* can accurately identify falsification behaviors and utilize *Selector* for fraud detection. Meanwhile, *GAMER* does not lose useful information by utilizing *Completer* for fraud detection if the transactions are unfalsified. Moreover, after Jan. 1, attacks are precipitately exacerbated, leading to a significant decrease in the F1 score of *Completer*. However, *Selector* and *GAMER* achieve higher F1 scores since the model equipped with *Game Selection* is robust to feature falsification. In particular, *GAMER* achieves higher F1 score than *Selector* since there still were unfalsified transactions after Jan. 1, and *GAMER* does not lose useful information on these transactions by employing *Completer* in fraud detection. As a result, *GAMER* increases the F1 score by 67.5% on average compared to *Completer* and can increase the F1 score by up to 19.47% compared to *Selector*, demonstrating that *GAMER* achieves higher accuracy on unfalsified transactions and strong robustness on falsified transactions.

The Rationale for Hypothesis Testing. To validate the hypothesis testing in *GAMER*, we display the CDF of κ on different transactions, including the transactions for training, the transactions before Jan. 1 (i.e., the transactions under minor attacks), and the transactions after Jan. 1 (i.e., the transactions under major attacks). The results in Fig. 18 indicate that the falsification process significantly increases κ because the model predictions in *GAMER* are distinct on falsified transactions. Hence, the hypothesis testing process can effectively decide the output of the detection system based on the difference between *Completer* prediction and *Selector* prediction.

The Efficiency of *Game Selection*. The results in Fig. 19 indicate that using *Game Selection* in model training is more efficient than adversarial training techniques. It slightly increases the training time by 10%, which is close to the optimal training time of the non-robust model.

Cross-Domain Generalizability. Operational statistics collected from a real-world enterprise environment highlight a key result: *GAMER* demonstrates strong generalization beyond its original fraud-detection context, effectively extending to other malicious-event scenarios such as online gambling and account-takeover activities. This empirical evidence underscores that attackers across these domains exhibit similar incentive-driven behaviors, and their falsification actions require comparable resources that align closely with our equilibrium-based detection framework. Consequently, the same game-theoretic formulation remains consistently effective across diverse categories of malicious activities, highlighting the practicality and robustness of *GAMER* in real-world applications.

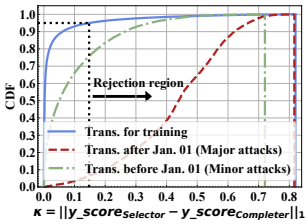


Fig. 18. CDF of prediction difference over different data.

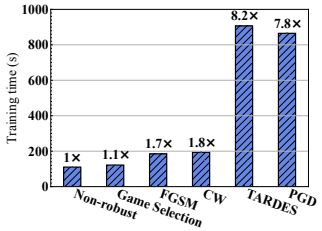


Fig. 19. Training time (s) for 35 epochs of TabNet.

VIII. Related Work

Adversarial Training. In machine learning, non-robust features [30], [40], [64] are demonstrated to cause the vulnerability of the model to adversarial examples [12], [89], [70], [10], [16], [76], [2], [61]. Then multiple researchers devote to enhancing the robustness of the model by utilizing adversarial examples in model training [75], [45], [53], [68], [77], [54], [83], [20], [65]. Zhang et al. [86] and Zhou et al. [88] focus on the problem of imperfect supervision and propose to implement adversarial training with complementary labels. Kireev et al. [42], [41] reveal the differences in detecting adversarial examples on tabular data and extend adversarial training techniques to this field. Specifically, adversarial training techniques improve the robustness by smoothing the model and leveraging data distribution shift [87]. However, these techniques cannot exhaust all possible adversarial examples, thus only locally enhancing the robustness of the ML model [60], [52], i.e., they will be ineffective when feature falsification are unrestricted [11]. Unlike adversarial training, our work leverages feature selection to reduce the combinations of features for fraudsters and

incorporates the equilibrium of a two-player game to maximize the effectiveness of feature selection.

Feature Selection. Xiao et al. [78] enhance the robustness of model training by LASSO, leading to a sparse feature selection process. Yoon et al. [82] establish a neural network-based feature selection method, which employs an extra neural network to calculate the importance of the features. Similarly, Yan et al. [80] aim to mitigate adversarial attacks by selecting features with an additional neural network. However, these techniques enhance the robustness of the model by utilizing fewer features, but the model is still definite after deployment, which means that adversarial examples can be discovered by intelligent fraudsters if the data coverage is insufficient (i.e., Theorem 1). Compared to these methods, *GAMER* calculates the optimal probability of feature selection and employs equilibrium-based strategies to deter attackers.

IX. Conclusion

In this work, we formulate fraud detection with falsified transactions as a two-player game between the detection system and the fraudster. Our proposed game-theoretic fraud detection system *GAMER* leverages equilibrium-based probability in feature selection, enabling the detection system to select robust features to detect fraudulent activities. Additionally, the equilibrium-based probability also reduces the attack profits of fraudsters, exhibiting deterrence to the fraudsters. Our theoretical analysis and extensive experiments validate these properties, demonstrating that *GAMER* achieves high accuracy on unfalsified transactions and strong robustness on falsified transactions.

X. Ethical Considerations

The real-world data is preprocessed as tabular data with no sensitive user information. Data are stored on the enterprise's devices and we access these data through an internship program. To mitigate any potential disruption to the production environment, we performed experiments in an isolated environment. The experimental procedures were reviewed and approved by the enterprise's ethics board.

Acknowledgement

We thank our shepherd and anonymous reviewers for their valuable comments. This work was supported in part by the National Science Foundation for Distinguished Young Scholars of China (Grant No. 62425201), the Science Fund for Creative Research Groups of the National Natural Science Foundation of China (Grant No. 62221003), the National Natural Science Foundation of China (Grant No. 62132011, No. 62472036, No. 62202258, No. U23B2026, and No. 62372305), Fundamental and Interdisciplinary Disciplines Breakthrough Plan of the Ministry of Education of China (Grant No. JYB2025XDXM114), Guangdong Basic and Applied Basic Research Foundation (Grant No. 2024B1515040012), Beijing-Tianjin-Hebei Natural Science Foundation Cooperation Project (Grant No. 25JJJC0003), and Beijing Nova Program. The authors from Ant Group are supported by the Leading Innovative and Entrepreneur Team Introduction Program of Hangzhou (Grant No.TD2022005). Laizhong Cui, Qi Li and Ke Xu are the corresponding authors.

References

[1] C. Amendola, P. Dettling, M. Drton, F. Onori, and J. Wu. Structure learning for cyclic linear causal models. In *Proceedings of UAI*, volume 124 of *Proceedings of Machine Learning Research*, pages 999–1008. PMLR, 03–06 Aug 2020.

[2] M. Andriushchenko, F. Croce, N. Flammarion, and M. Hein. Square attack: A query-efficient black-box adversarial attack via random search. In *Proceedings of ECCV*, volume 12368 of *Lecture Notes in Computer Science*, pages 484–501. Springer, 2020.

[3] G. Apruzzese, H. S. Anderson, S. Dambra, D. Freeman, F. Pierazzi, and K. Roundy. "real attackers don't compute gradients": bridging the gap between adversarial ml research and practice. In *Proceedings of SaTML*, pages 339–364. IEEE, 2023.

[4] G. Apruzzese, H. S. Anderson, S. Dambra, D. Freeman, F. Pierazzi, and K. A. Roundy. "real attackers don't compute gradients": Bridging the gap between adversarial ML research and practice. In *Proceedings of SaTML*, pages 339–364. IEEE, 2023.

[5] S. Ö. Arik and T. Pfister. Tabnet: Attentive interpretable tabular learning. In *Proceedings of AAAI*, pages 6679–6687. AAAI Press, 2021.

[6] A. Asudeh, Z. Jin, and H. V. Jagadish. Assessing and remedying coverage for a given dataset. In *Proceedings of ICDE*, pages 554–565. IEEE, 2019.

[7] A. Asudeh, N. Shahbazi, Z. Jin, and H. V. Jagadish. Identifying insufficient data coverage for ordinal continuous-valued attributes. In *Proceedings of SIGMOD*, page 129–141. Association for Computing Machinery, 2021.

[8] A. Bhattad, M. J. Chong, K. Liang, B. Li, and D. A. Forsyth. Unrestricted adversarial examples via semantic manipulation. In *Proceedings of ICLR*. OpenReview.net, 2020.

[9] I. BPM. Ai in the banking sector: How fraud detection with ai is making banking safer, 2024.

[10] W. Brendel, J. Rauber, and M. Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. In *Proceedings of ICLR*. OpenReview.net, 2018.

[11] T. B. Brown, N. Carlini, C. Zhang, C. Olsson, P. F. Christiano, and I. J. Goodfellow. Unrestricted adversarial examples. *CoRR*, abs/1809.08352, 2018.

[12] N. Carlini and D. A. Wagner. Towards evaluating the robustness of neural networks. In *Proceedings of S&P*, pages 39–57. IEEE Computer Society, 2017.

[13] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. SMOTE: synthetic minority over-sampling technique. *J. Artif. Intell. Res.*, 16:321–357, 2002.

[14] H. Chen, H. Zhang, D. S. Boning, and C. Hsieh. Robust decision trees against adversarial examples. In *Proceedings of ICML*, volume 97 of *Proceedings of Machine Learning Research*, pages 1122–1131. PMLR, 2019.

[15] Z. Chen, B. Li, S. Wu, K. Jiang, S. Ding, and W. Zhang. Content-based unrestricted adversarial attack. In *Proceedings of NeurIPS*, 2023.

[16] S. Cheng, Y. Dong, T. Pang, H. Su, and J. Zhu. Improving black-box adversarial attacks with a transfer-based prior. In *Proceedings of NeurIPS*, pages 10932–10942, 2019.

[17] B. Clarke, B. Leuridan, and J. Williamson. Modelling mechanisms with causal cycles. *Synthese*, 191:1651–1681, 2014.

[18] CMS. The role, opportunities and challenges of ai in detecting financial fraud, 2024.

[19] F. Croce and M. Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *Proceedings of ICML*, volume 119, pages 2206–2216. PMLR, 2020.

[20] S. Cui, J. Zhang, J. Liang, B. Han, M. Sugiyama, and C. Zhang. Synergy-of-experts: Collaborate to improve adversarial robustness. In *Proceedings of NeurIPS*, 2022.

[21] A. Davitaia. Artificial intelligence and machine learning in fraud detection for digital payments. *International Journal of Science and Research Archive*, 15(3):714–719, 2025.

[22] Y. Dong, Z. Deng, T. Pang, J. Zhu, and H. Su. Adversarial distributional training for robust deep learning. In *Proceedings of NeurIPS*, 2020.

[23] F. Fang, T. H. Nguyen, R. Pickles, W. Y. Lam, G. R. Clements, B. An, A. Singh, M. Tambe, and A. Lemieux. Deploying PAWS: field optimization of the protection assistant for wildlife security. In *Proceedings of AAAI*, pages 3966–3973. AAAI Press, 2016.

[24] K. Fukunaga. *Introduction to statistical pattern recognition*. Elsevier, 2013.

[25] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. In *Proceedings of ICLR 2015*, 2015.

[26] B. Harsha. *Modeling rational adversaries: Predicting behavior and developing deterrents*. PhD thesis, Purdue University, 2021.

[27] M. Hernan and J. Robins. *Causal Inference: What If*. CRC Press, 2025.

[28] L. Hsiung, Y. Tsai, P. Chen, and T. Ho. Towards compositional adversarial robustness: Generalizing adversarial training to composite semantic perturbations. In *Proceedings of CVPR*, pages 24658–24667. IEEE, 2023.

[29] M. Huang, Y. Liu, X. Ao, K. Li, J. Chi, J. Feng, H. Yang, and Q. He. Auc-oriented graph neural network for fraud detection. In *Proceedings of WWW*, pages 1311–1321. ACM, 2022.

[30] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry. Adversarial examples are not bugs, they are features. In *Proceedings of NeurIPS*, pages 125–136, 2019.

[31] F. M. I. Inc. Ai in fraud management market snapshot, 2023.

[32] Z. Jin, M. Xu, C. Sun, A. Asudeh, and H. V. Jagadish. Mithracoverage: A system for investigating population bias for intersectional fairness. In *Proceedings of SIGMOD*, pages 2721–2724. ACM, 2020.

[33] A. Joshi, A. Mukherjee, S. Sarkar, and C. Hegde. Semantic adversarial attacks: Parametric transformations that fool deep classifiers. In *Proceedings of ICCV*, pages 4772–4782. IEEE, 2019.

[34] A. B. Journal. Survey finds fraud costs rising for banks, 2022.

[35] Kaggle. Credit Card Fraud Detection, 2017. URL: https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud.

[36] Kaggle. IEEE-CIS fraud detection, 2019. URL: https://www.kaggle.com/c/ieee-fraud-detection.

[37] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T. Liu. Lightgbm: A highly efficient gradient boosting decision tree. In *Proceeding of NeurIPS*, pages 3146–3154, 2017.

[38] C. Keeble, G. R. Law, S. Barber, and P. D. Baxter. Choosing a method to reduce selection bias: a tool for researchers. *Open Journal of Epidemiology*, 5(03):155–162, 2015.

[39] H. Kim. Torchattacks : A pytorch repository for adversarial attacks. *CoRR*, abs/2010.01950, 2020.

[40] J. Kim, B. Lee, and Y. M. Ro. Distilling robust and non-robust features in adversarial examples by information bottleneck. In *Proceedings of NeurIPS*, pages 17148–17159, 2021.

[41] K. Kireev, M. Andriushchenko, C. Troncoso, and N. Flammarion. Transferable adversarial robustness for categorical data via universal robust embeddings. In *Proceedings of NeurIPS*, 2023.

[42] K. Kireev, B. Kulynych, and C. Troncoso. Adversarial robustness for tabular data through cost and utility awareness. In *Proceedings of NDSS*. The Internet Society, 2023.

[43] Y. Lin, Y. Guan, A. Asudeh, and H. V. Jagadish. Identifying insufficient data coverage in databases with multiple relations. In *Proceedings of VLDB*, volume 13, page 2229–2242. VLDB Endowment, 2020.

[44] Y. Liu, M. Zhang, D. Li, K. Jee, Z. Li, Z. Wu, J. Rhee, and P. Mittal. Towards a timely causality analysis for enterprise security. In *Proceedings of NDSS*. The Internet Society, 2018.

[45] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. In *Proceedings of ICLR*. OpenReview.net, 2018.

[46] C. Malone. Combatting online payment fraud, 2023.

[47] E. Mariconti, J. Onaolapo, G. J. Ross, and G. Stringhini. The cause of all evils: Assessing causality between user actions and malware activity. In *Proceedings of Usenix Security*. USENIX Association, 2017.

[48] MasterCard. Industry perspectives on ai and transaction fraud detection, 2023.

[49] H. Miao, F. Ma, R. Quan, K. Zhan, and Y. Yang. Autonomous llm-enhanced adversarial attack for text-to-motion. *CoRR*, abs/2408.00352, 2024.

[50] B. Mohamed, D. H. Kheliouane, and A. T. Akrouf. Evaluation measures for models assessment over imbalanced data sets. *Journal of Information Engineering and Applications*, 3(10), 2013.

[51] I. Padhi, Y. Schiff, I. Melnyk, M. Rigotti, Y. Mroueh, P. L. Dognin, J. Ross, R. Nair, and E. Altman. Tabular transformers for modeling multivariate time series. In *Proceedings of ICASSP*, pages 3565–3569. IEEE, 2021.

[52] T. Pang, K. Xu, Y. Dong, C. Du, N. Chen, and J. Zhu. Rethinking softmax cross-entropy loss for adversarial robustness. In *Proceedings of ICLR*. OpenReview.net, 2020.

[53] T. Pang, K. Xu, C. Du, N. Chen, and J. Zhu. Improving adversarial robustness via promoting ensemble diversity. In *Proceedings of ICML*, volume 97, pages 4970–4979. PMLR, 2019.

[54] T. Pang, X. Yang, Y. Dong, H. Su, and J. Zhu. Bag of tricks for adversarial training. In *Proceedings of ICLR*. OpenReview.net, 2021.

[55] P. Paruchuri, J. P. Pearce, J. Marecki, M. Tambe, F. Ordóñez, and S. Kraus. Playing games for security: an efficient exact algorithm for solving bayesian stackelberg games. In *Proceedings of AAMAS*, pages 895–902. IFAAMAS, 2008.

[56] J. Pearl. *Causality*. Cambridge university press, 2009.

[57] J. Pearl, M. Glymour, and N. P. Jewell. *Causal inference in statistics: A primer*. John Wiley & Sons, 2016.

[58] J. Pearl and D. Mackenzie. *The book of why: the new science of cause and effect*. Basic books, 2018.

[59] J. Pita, M. Jain, J. Marecki, F. Ordóñez, C. Portway, M. Tambe, C. Western, P. Paruchuri, and S. Kraus. Deployed ARMOR protection: the application of a game theoretic model for security at the los angeles international airport. In *Proceedings of (AAMAS 2008)*, pages 125–132. IFAAMAS, 2008.

[60] L. Schmidt, S. Santurkar, D. Tsipras, K. Talwar, and A. Madry. Adversarially robust generalization requires more data. In *Proceedings of NeurIPS*, pages 5019–5031, 2018.

[61] L. Schwinn, R. Raab, A. Nguyen, D. Zanca, and B. M. Eskofier. Exploring misclassifications of robust neural networks to enhance adversarial attacks. *Appl. Intell.*, 53(17):19843–19859, 2023.

[62] A. S. Shamsabadi, R. Sánchez-Matilla, and A. Cavallaro. Colorfool: Semantic adversarial colorization. In *Proceedings of CVPR*, pages 1148–1157. Computer Vision Foundation / IEEE, 2020.

[63] Y. Song, R. Shu, N. Kushman, and S. Ermon. Constructing unrestricted adversarial examples with generative models. In *Proceedings of NeurIPS*, pages 8322–8333, 2018.

[64] J. M. Springer, M. Mitchell, and G. T. Kenyon. A little robustness goes a long way: Leveraging robust features for targeted transfer attacks. In *Proceedings of NeurIPS*, pages 9759–9773, 2021.

[65] J. Tack, S. Yu, J. Jeong, M. Kim, S. J. Hwang, and J. Shin. Consistency regularization for adversarial robustness. In *Proceedings of AAAI*, pages 8414–8422. AAAI Press, 2022.

[66] M. Tambe. *Security and Game Theory - Algorithms, Deployed Systems, Lessons Learned*. Cambridge University Press, 2012.

[67] Y. Tang, D. Li, Z. Li, M. Zhang, K. Jee, X. Xiao, Z. Wu, J. Rhee, F. Xu, and Q. Li. Nodemerge: Template based efficient data reduction for big-data causality analysis. In *Proceedings of CCS*, pages 1324–1337. ACM, 2018.

[68] F. Tramèr and D. Boneh. Adversarial training and robustness for multiple perturbations. In *Proceedings of NeurIPS*, pages 5858–5868, 2019.

[69] F. Tramèr, N. Carlini, W. Brendel, and A. Madry. On adaptive attacks to adversarial example defenses. In *Proceedings of NeurIPS*, 2020.

[70] J. Uesato, B. O'Donoghue, P. Kohli, and A. van den Oord. Adversarial risk and the dangers of evaluating against weak attacks. In *Proceedings of ICML*, volume 80 of *Proceedings of Machine Learning Research*, pages 5032–5041. PMLR, 2018.

[71] M. Uliasz. Ai advances: Smarter fraud detection for secure transactions, 2024.

[72] C. Wang, J. Duan, C. Xiao, E. Kim, M. C. Stamm, and K. Xu. Semantic adversarial attacks via diffusion models. In *Proceedings of BMVC*, page 271. BMVA Press, 2023.

[73] L. Wasserman. All of statistics: a concise course in statistical inference, 2013.

[74] J. West, M. Bhattacharya, and M. R. Islam. Intelligent financial fraud detection practices: An investigation. In *Proceedings of ICST*, volume 153, pages 186–203. Springer, 2014.

[75] E. Wong and J. Z. Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In J. G. Dy and A. Krause, editors, *Proceedings of ICML*, volume 80 of *Proceedings of Machine Learning Research*, pages 5283–5292. PMLR, 2018.

[76] E. Wong, L. Rice, and J. Z. Kolter. Fast is better than free: Revisiting adversarial training. In *Proceedings of ICLR*. OpenReview.net, 2020.

[77] D. Wu, S. Xia, and Y. Wang. Adversarial weight perturbation helps robust generalization. In *Proceedings of NeurIPS*, 2020.

[78] H. Xiao, B. Biggio, G. Brown, G. Fumera, C. Eckert, and F. Roli. Is feature selection secure against training data poisoning? In *Proceedings of ICML*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 1689–1698. JMLR.org, 2015.
[79] X. Xu, K. Kong, N. Liu, L. Cui, D. Wang, J. Zhang, and M. S. Kankanhalli. An LLM can fool itself: A prompt-based adversarial attack. In *Proceedings of ICLR*. OpenReview.net, 2024.
[80] H. Yan, J. Zhang, G. Niu, J. Feng, V. Y. F. Tan, and M. Sugiyama. CIFS: improving adversarial robustness of cnns via channel-wise importance-based feature selection. In *Proceedings of ICML*, volume 139 of *Proceedings of Machine Learning Research*, pages 11693–11703. PMLR, 2021.
[81] Z. Yin, D. Korzhyk, C. Kiekintveld, V. Conitzer, and M. Tambe. Stackelberg vs. nash in security games: interchangeability, equivalence, and uniqueness. In *Proceedings of AAMAS*, pages 1139–1146. IFAAMAS, 2010.
[82] J. Yoon, J. Jordon, and M. van der Schaar. INVASE: instance-wise variable selection using neural networks. In *Proceedings of ICLR*. OpenReview.net, 2019.
[83] Y. Yu, Z. Yang, E. Dobriban, J. Steinhardt, and Y. Ma. Understanding generalization in adversarial training via the bias-variance decomposition. *CoRR*, abs/2103.09947, 2021.
[84] X. Yuan, P. He, Q. Zhu, and X. Li. Adversarial examples: Attacks and defenses for deep learning. *IEEE Trans. Neural Networks Learn. Syst.*, 30(9):2805–2824, 2019.
[85] H. Zhang, Y. Yu, J. Jiao, E. P. Xing, L. E. Ghaoui, and M. I. Jordan. Theoretically principled trade-off between robustness and accuracy. In *Proceedings of ICML*, volume 97 of *Proceedings of Machine Learning Research*, pages 7472–7482. PMLR, 2019.
[86] J. Zhang, X. Xu, B. Han, T. Liu, G. Niu, L. Cui, and M. Sugiyama. Noilin: Do noisy labels always hurt adversarial training? *CoRR*, abs/2105.14676, 2021.
[87] Y. Zhang, S. Hu, L. Y. Zhang, J. Shi, M. Li, X. Liu, W. Wan, and H. Jin. Why does little robustness help? a further step towards understanding adversarial transferability. In *Proceedings of S&P*. IEEE, 2024.
[88] J. Zhou, J. Zhu, J. Zhang, T. Liu, G. Niu, B. Han, and M. Sugiyama. Adversarial training with complementary labels: On the benefit of gradually informative attacks. In *Proceedings of NeurIPS*, 2022.
[89] L. Zhou, P. Cui, Y. Jiang, and S. Yang. Adversarial eigen attack on black-box models. *CoRR*, abs/2009.00097, 2020.

APPENDIX

Detailed Analysis of the Motivation to Falsify Input Features. With the combinations of $y \in \{0, 1\}$ and $\hat{y} \in \{0, 1\}$, there are four cases for each user. As explained in Table VII, the combination of y and $\hat{y}$ results in distinct users behaviors. It is worth noting that the predicted label, i.e., $\hat{y}$, leads to differences in the users' behavior.

Specifically, if $\hat{y} = 0$, users have no incentive to modify the features regardless of the real label since they get the optimal results from the detection system, i.e., benign users can normally use the service and fraudsters can get illegal profits via the service. Otherwise, if $\hat{y} = 1$, either benign users or fraudsters will modify their behaviors (i.e., the features) to keep using the service, i.e., the benign users will regulate their behavior to avoid being misclassified and fraudsters will falsify features to escape the punishments of the enterprise and get illegal profits via keeping using the service.

***Game Selection* for GBDT.** As the Gradient Boosted Decision Tree (GBDT) is the state-of-the-art model on tabular data classification, we extend *Game Selection* to GBDT training. As the state-of-the-art GBDT-based models are addition models, the method based on Monte-Carlo algorithm to calculate the expectation, i.e., Algorithm 1, is unsuitable in this scenario.

TABLE VII
THE MOTIVATIONS TO FALSIFY INPUT FEATURES

Case	Explanation
$y = 0,\ \hat{y} = 0$	Correctly detected benign users, they *will not modify features* since they can use the service normally.
$y = 1,\ \hat{y} = 0$	Mis-classified fraudsters, they *will not consume resources to falsify features* since they can already obtain the illegal profits from fraudulent activities.
$y = 0,\ \hat{y} = 1$	Mis-classified benign users, they *will regulate (modify) their behaviors* to reuse the provided service.
$y = 1,\ \hat{y} = 1$	Correctly detected fraudsters, they *will consume resources to falsify features* to escape punishment and get the illegal profits.

To solve this issue, we change the target function of Eq. (4) to the following equation.

$$\begin{aligned} & \min \mathbb{E}_{\boldsymbol{X},\boldsymbol{m}} \left[\mathcal{L}(F(\boldsymbol{X} \odot \boldsymbol{m}); y)\right] \\ = \quad & \min \mathbb{E}_{\boldsymbol{m}} \left[\mathbb{E}_{\boldsymbol{X}} \left[\mathcal{L}(F(\boldsymbol{X} \odot \boldsymbol{m}); y)\right] | \boldsymbol{m}\right] \\ \leq \quad & \mathbb{E}_{\boldsymbol{m}} \min \left[\mathbb{E}_{\boldsymbol{X}} \left[\mathcal{L}(F(\boldsymbol{X} \odot \boldsymbol{m}); y)\right] | \boldsymbol{m}\right], \end{aligned} \tag{8}$$

where the equality depends on the property of conditional expectation and the inequality comes from the fact that the minimum of the expectation is less than the expectation of the conditional minimum. With the new target function, we can minimize Eq. (8) to further minimize the target function, hence we can apply *Game Selection* to the addition models, e.g., LightGBM (as shown in Algorithm 2).

Algorithm 2 Training LightGBM with *Game Selection*

Input: The training data $\boldsymbol{X}$, the iterations T, the profits $\{\Delta U_i\}_{i=1}^d$, the costs $\{C_i\}_{i=1}^d$
Output: Final model $\bar{F}(\cdot)$
1: **Get optimal selection rate:**
   $p_i = \min\left(1, \frac{C_i}{\Delta U_i}\right),\ i \in \{1, \cdots, d\}$
2: **Get multivariate Bernoulli distribution:**
   $\boldsymbol{m} = (m_1, \cdots, m_d)$, where $m_i \sim \text{Bernoulli}(p_i)$
3: **for** $t = 1$ **to** T **do**
4: Sample $\boldsymbol{m}^{(t)}$ from $\boldsymbol{m}$
5: Train LightGBM $F^{(t)}(\cdot)$ with $\boldsymbol{X} \odot \boldsymbol{m}^{(t)}$
6: **end for**
7: $\bar{F}(\cdot) = \frac{1}{T}\sum_{t=1}^{T} F^{(t)}(\cdot)$
8: **return** $\bar{F}(\cdot)$

The Details of the Simulated Experiments. In simulated experiments in section VII, we employ three real-world datasets in fraud detection to validate our design. All of these scenarios have social or economic implications. The details of these datasets are as follows.

- TabFormer [35]. The dataset contains 24 million transactions of 20,000 users and each transaction has 11 features. To keep the consistent data scale, we utilize a subset composed of all 29,342 fraudulent samples and 300K

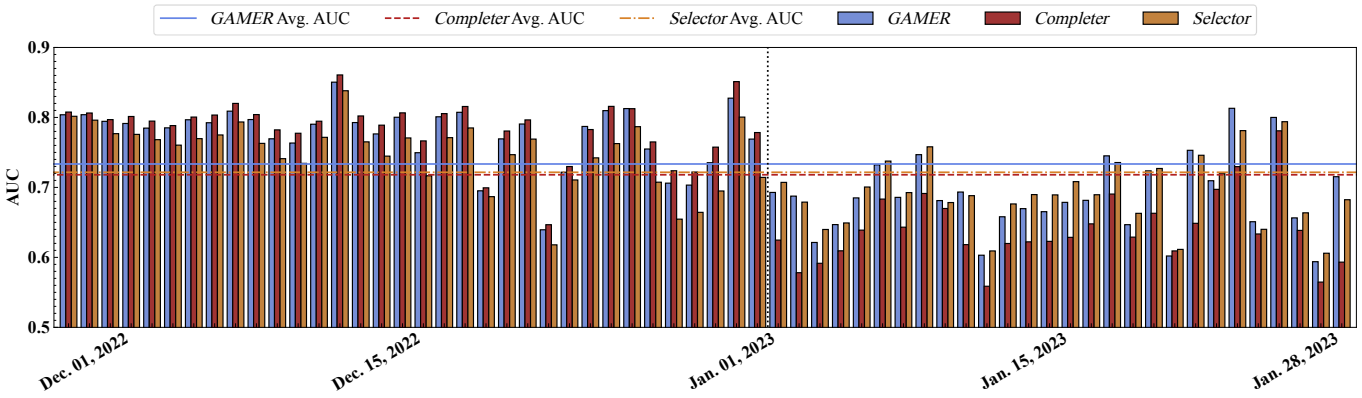


Fig. 20. Daily evaluation on real-world fraud detection (AUC).

rows that are randomly sampled from the non-fraudulent samples.

- CreditCard [35]. The dataset contains 284,807 transactions made by credit cards of European cardholders. There are 492 frauds out of these transactions. Moreover, each transaction is explained with 30 features.
- IEEE-CIS [36]. The dataset contains 600K financial transactions, and 20663 samples are labeled fraudulent. We use 173 features selected based on the best solutions of the Kaggle competition [36].

Moreover, in these experiments, we utilize a cost-aware attacker to simulate real-world falsifications in fraud detection [42]. These attackers are more realistic since they take the costs of falsifying different features into account. The details of the attack are indicated in Algorithm 3.

Algorithm 3 Cost Aware Attack

Input: Initial transaction $\boldsymbol{x}$, label y, costs $\boldsymbol{C}$, attack cost bound ϵ, optimization steps T
Output: falsified transaction $\boldsymbol{x}^*$

$\alpha := \frac{\epsilon}{T}$
$\boldsymbol{\delta} = \mathbf{0}$
for $t = 1$ **to** T **do**
  $\nabla = \nabla_{\boldsymbol{\delta}} \mathcal{L}(F(\boldsymbol{W}; \boldsymbol{x} + \frac{\boldsymbol{\delta}}{\boldsymbol{C}}); y)$
  $\boldsymbol{\delta} = \boldsymbol{\delta} + \alpha \cdot \text{sign}(\nabla) \cdot \frac{\boldsymbol{C}}{\|\boldsymbol{C}\|_1}$
  if $\|\boldsymbol{\delta}\|_1 > \epsilon$ **then**
    $\boldsymbol{\delta} = \boldsymbol{\delta} \cdot \frac{\epsilon}{\|\boldsymbol{\delta}\|_1}$
  end if
end for
$\boldsymbol{x}^* = \boldsymbol{x} + \frac{\boldsymbol{\delta}}{\boldsymbol{C}}$
return $\boldsymbol{x}^*$

The Details of the Real-world Data Collected from the World's Leading Online Payment Enterprise. The real-world data are collected from the scenario of inter-enterprise online transactions. In particular, there are about 200 to 300 transactions that are complaint by users (i.e., fraudulent activities) out of 110 thousand transactions a day.

The risk control department collects two-month transactions from Oct. 7 to Nov. 26, 2022 for us as training data, each transaction is described with 302 features, which capture various factors such as transaction amount, transaction frequency, historical records, and any irregularities and anomalies that may indicate potential financial risks. Moreover, the final training data contain 16070 fraudulent transactions, which are the complaints from users in two months. As the data are highly imbalanced, we employ the random under-sampling strategy to sample benign transactions with the factor of $1:2$.

After model training, we validate the resulting model on the test data, which are collected from the transactions from Nov. 28, 2022 to Jan. 28, 2023.

The Process of Deciding the Costs in Real-world Experiments. The costs were decided by the consensus of a group of security experts in the world's leading online payment enterprise. The steps were as follows:

- Experts divided all 302 features into three categories according to the cost (e.g., monetary or time) of falsification (high cost, medium cost, or low cost).
- For each category, the experts chose one feature from the set as the representative feature.
- Each expert individually evaluated the cost of the representative feature (normalized to \$100 profit), this cost was set as the cost of all features in the corresponding feature set.
- The costs set for each feature are averaged across all security experts.

Additional Experimental Results. In this part, we display additional experimental results to further indicate that our proposed methods are effective at detecting unrestrictedly falsified transactions.

Specifically, we display the AUC of the corresponding daily ablation evaluation of *GAMER* in Fig. 20. The experiments indicate that *GAMER* achieves higher AUC on average than the *Completer*, especially after Jan. 1, 2023 (i.e., the adversarial attacks were exacerbated from that day on), which means *GAMER* enhances the robustness of the detection model to combat real-world fraudsters. Moreover, the AUC of *GAMER* is higher than the *Selector* before Jan.1, 2023 (i.e., the falsification were minor before Jan. 1), which demonstrates that *GAMER* can make full use of all input features to accurately detect fraudulent activities compared to *Selector*. As a result, *GAMER* at most increases the AUC by 20.58% compared to the *Completer* and increases the AUC by up to 7.9% compared to the *Selector*

during the two-month evaluation. It is worth noting that for imbalanced data, which is the common case in fraud detection, AUC sometimes gives a misleading of model performance [50], hence it should be considered in conjunction with the F1 score to properly evaluate the model performance. Specifically, as indicated in Fig. 17, *GAMER* also achieves higher F1 score compared to *Completer* and *Selector* during the two-month evaluation. These experimental results are consistent with our theoretical analysis, indicating that *GAMER* can achieve high accuracy on unfalsified transactions and strong robustness on falsified transactions.

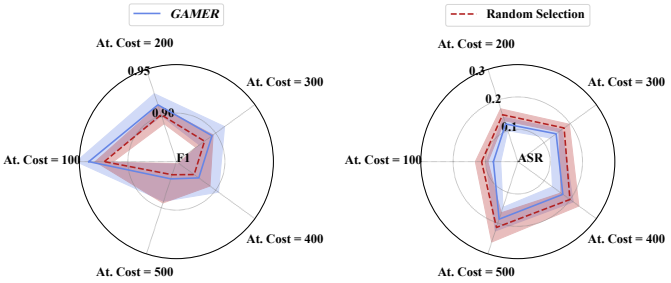


Fig. 21. The importance of the cost-profit knowledge (CreditCard).

Finally, Fig. 21 displays the comparisons between *GAMER* and the random selection (i.e., select each feature with the probability of 50%, which is the random guess without expert knowledge) on CreditCard dataset. The results indicate that when the cost-profit knowledge is employed, the maximum increase of the F1 score are 1.73%. Moreover, *GAMER* reduces ASR by up to 32.59%, which demonstrates that incorporating cost-profit knowledge is beneficial for accurate fraud detection.