POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Executive Summary of the Thesis

# Improving Poisoning Attacks against Banking Fraud Detection Systems

Laurea Magistrale in Computer Science and Engineering - Ingegneria Informatica

**Author:** Andrea Ventura

**Advisor:** Prof. Michele Carminati

**Co-advisor:** Prof. Stefano Zanero

**Academic year:** 2021-2022

## 1. Introduction

It has been proven that machine learning algorithms, applied to the banking fraud domain, can be deceived and corrupted through evasion and poisoning attacks [1, 3]. In particular, banking detectors are periodically trained according to a specific update policy. An adversary can exploit the re-training process to perform poisoning attacks. They craft fraudulent transactions which, if considered legitimate, are included in the training set that will be used for the learning task.

In this work, we focus on poisoning attacks against banking detection systems. We improve the results obtained by Monti [1], which is the first work in the context of poisoning attacks applied to banking FDSs. Our approach considers different degrees of knowledge about the target system: White Box (perfect knowledge), Grey Box (partial knowledge), and Black Box (no knowledge). According to specific metrics, we evaluate poisoning attacks against eight detectors, trained with a weekly and bi-weekly update policy. We explore three different attack strategies and we analyze each knowledge scenario separately. We summarize our contributions:

- We present a novel method for crafting fraudulent transactions, which is able to control a larger number of features with respect to [1];
- We show a novel approach to building a reliable Oracle, by combining multiple learners with an ensemble method;
- We use a new transaction process, by which the adversary can generate several features during the attack and we deeply analyze which features are convenient to modify at runtime.

## 2. Dataset Analysis and Engineering

We work on two datasets composed of real executed transactions. We select a subset of the features that really interest our purposes. The most relevant features are IP address, session ID, timestamp, amount, user ID, IBAN, confirmation SMS, IBAN_CC, and CC_ASN. In Table 1 we report general information about the two datasets.

| Dataset | Time Window | Users | Transactions | Mean (€) | Max-Min(€) |
|---|---|---|---|---|---|
| 2012-13 | 01/12/12-10/09/13 | 53764 | 567550 | 1786.38 | 0.01-50000 |
| 2014-15 | 22/10/14-23/02/15 | 58507 | 471766 | 1778.99 | 0.01-50000 |

Table 1: General Information about the Datasets

### 2.1. Synthetic Fraud Generation

The dataset 2012-13 has been completely cleaned from frauds, while the banking group made available a list of 606 fraud reports concerning the dataset 2014-15. Since we are in a supervised learning setting and, according to the literature [1–4], frauds usually constitute between 0.1% and 1% of the total transactions, we need to craft fraud samples to effectively face the classification task. We replicate malicious behaviors according to two fraudulent patterns, information stealing and transaction hijacking.
To synthesize fraudulent wire transfers, we exploit the same strategy used by Monti [1], with minor modifications. In Table 2, we summarize the results of the fraud generation process.

| Dataset | IS frauds | TH frauds | Reported frauds | Total frauds | Frauds percentage |
|---|---|---|---|---|---|
| 2012-13 | 3982 | 808 | 0 | 4790 | 0.85% |
| 2014-25 | 4534 | 759 | 606 | 4899 | 1.15% |

Table 2: Generation Frauds Results

## 3. Fraud Detection Systems: Tuning, Training, and Evaluation

We present 8 different detectors: Random Forest (**RF**), XGBoost (**XGB**), Light Gradient Boosting (**LGB**), CatBoost (**CB**), Support Vector Machine (**SVM**), Artificial Neural Networks (**ANN**), Logistic Regression (**LR**), and Active Learning (**AL**). We train and evaluate each model after having performed feature aggregation, feature selection and hyperparameter tuning tasks.

### 3.1. Feature Aggregation

To create powerful detectors, we need to train the machine learning algorithms on a dataset that collects as much relevant information as possible. This is why direct features are not enough: we need to aggregate them to capture the user's spending pattern and his or her behavior in a certain time period. With our aggregation strategy, we are able to extract 196 numerical features.

### 3.2. Feature Selection and Hyperparameter Tuning

Feature selection consists of extracting from the entire set of features, those which best fit each specific model. For computational reasons, we exploit a filter solution, which inspects how much every feature impacts the true label. Thanks to this approach, we decrease the feature number from 196 to about 80 for each algorithm and, on average, we lose 0.05% of our proportional accuracy, an acceptable percentage.
To find the optimal hyperparameter set of each model, we adopt a Random Grid Search solution, according to which 30 different combinations of hyperparameters are evaluated with a 3-fold cross-validation strategy, a good compromise for an accurate search computationally acceptable.

### 3.3. Model Evaluation

We periodically train our models according to two update policies, weekly and biweekly, so that detectors can incrementally update their training set including new examples. In Table 3, we show the performances of the models trained according to a weekly update policy. Active Learning performs better than the others, achieving 99.27% in proportional accuracy. Then, we have CatBoost, XGBoost, LightGB, Random Forest, Logistic Regression, Artificial Neural Networks, and finally Support Vector Machine. In general, our custom metric is above 93.64%, except for SVM, which is the less powerful detector.

| Model | P-acc | Precision | Recall | F1 | F2 | FPR | W-MCC | ROC-AUC | PRC-AUC |
|---|---|---|---|---|---|---|---|---|---|
| LightGB | 97.99 | 23.96 | 97.89 | 38.50 | 60.53 | 1.91 | 95.98 | 99.84 | 60.32 |
| CatBoost | 98.78 | 32.36 | 98.84 | 48.75 | 70.05 | 1.27 | 97.56 | 99.92 | 64.99 |
| XGBoost | 98.52 | 28.31 | 98.58 | 43.99 | 65.88 | 1.54 | 97.04 | 99.90 | 62.84 |
| AL | 99.27 | 39.65 | 99.48 | 56.70 | 76.42 | 0.93 | 98.55 | 99.96 | 68.95 |
| LR | 95.97 | 21.42 | 94.06 | 34.89 | 56.04 | 2.13 | 91.99 | 99.22 | 57.14 |
| RF | 97.74 | 28.44 | 96.99 | 43.98 | 65.44 | 1.50 | 95.49 | 99.81 | 62.10 |
| SVM | 88.74 | 3.2 | 95.18 | 6.2 | 14.13 | 17.70 | 78.12 | 94.27 | 48.59 |
| ANN | 93.64 | 94.92 | 87.31 | 90.96 | 88.74 | 0.04 | 80.15 | 96.31 | 52.11 |

Table 3: Fraud Detection Systems Metrics, Weekly Update

# 4. Poisoning Attacks

In this chapter, we explain the details that identify the attacker's approach to mount poisoning attacks.

## 4.1. Attack Approach Overview

The attack is divided into different phases. The first step is the understanding of which instruments he or she has at his or her disposal, i.e., the scenario. Then, the adversary selects the victims to who execute the attacks. In this work, the fraudster selects 15 victims, an empirical number that allows attacking customers with different spending patterns and guarantees an acceptable computational effort. The next step consists of retrieving the past transactions executed by the chosen victims, collecting all the information necessary to build users spending profiles and, consequently, crafting evasive frauds which partially replicate victims' behaviors. After crafting frauds, the adversary trains the Oracle, i.e., the model which takes care of validating and regenerating the malicious transactions. If the Oracle classifies them as legitimate, they are subjected to the target system; otherwise, they are regenerated (or deleted, in the worst case) and submitted again, until they overcome the Oracle check. If the proposed transactions are considered legit also by the target detector, another attack, after some days, depending on the update policy, will be performed. The bank system is now trained on data that contain the transactions crafted by the attacker. On the other hand, if the target system detects at least one fraud among those subjected, the attack against that victim ends and the adversary will affect another customer. The attack against one user lasts as long as the dataset ends (i.e., 8 weeks after the start of the attack) or when a fraud is detected.

## 4.2. Scenario and Strategy

In order to model the adversary's knowledge, we rely on Monti and Carminati et al. [1, 3]. We list all the relevant terms which refer to the three possible scenarios: training data on which the target model is trained ($\Delta$), set of features used to train the target algorithm ($\Phi$), the algorithm used to create the fraud detection system ($A$), the hyper-parameters used to train the machine learning model ($P$), past users transactions to identify the user spending pattern ($T$), and update policy of the target model ($\Pi$).

$\Theta_{wb} = (\Delta, \Phi, \mathrm{A}, \mathrm{P}, \mathrm{T}, \Pi)$, $\Theta_{gb} = (\delta, \Phi, \alpha, \rho, \tau, \Pi)$,

$$\Theta_{bb} = (\delta, \phi, \alpha, \rho, \tau, \pi)$$

We present three different strategies.

**Poisoning amount**. The attacker steals money in a small time window, without worrying about being detected. He or she focuses on poisoning the transactions amount, increasing it in a consistent way every iteration.

**Poisoning count**. The adversary poisons the count of transactions per week, crafting frauds that have an amount similar to the mean of legit transactions executed by the victim. According to Monti [1], increasing the count is more cautious than focusing on the amount.

**Poisoning both**. A hybrid approach in which the attacker's goal is to steal as much money as possible, poisoning both count and amount, without the worry to be detected.

Each strategy presents a conservative and a greedy version.

## 4.3. Retrieval and Crafting

In the White Box scenario, the attacker has all the previous transactions belonging to the victim, while in the Grey and Black Box scenarios he or she has partial knowledge (i.e., one month's transactions history). Once the retrieval phase is concluded, the fraudster crafts malicious transactions to start the poisoning process. In [1], Monti considered as controllable features only the amount, the timestamp, and the count. In this work, the attacker can also manipulate the IP address, the CC_ASN identifier, the IBAN, and the confirmation SMS. For each transaction, in order to select appropriate features, the adversary exploits specific algorithms that study the victim's spending profile and mimic the victim's behavior.

## 4.4. Oracle and Regeneration Process

The Oracle is the machine learning model which is built by the attacker to have a reliable imitation of the target Fraud Detection System. In [1] and [2], the authors propose to overcome this problem by using the best algorithm found, respectively XGBoost and Random Forest; the scope of this work is to propose and show an alternative method, based on ensembling learning,

which allows creating a very strong Oracle, that is reliable and closer to the target machine. After having explored and compared different ensembling solutions, namely Bagging, Boosting, Stacking, and Majority Voting, we can conclude that the most powerful Oracle found is based on Light Gradient Boosting algorithm, improved by Bagging with 20 bootstraps.
Based on the outcome of the Oracle, the attacker either submits them to the target FDS, or regenerates them by changing the IP address, IBAN, or CC_ASN, or by lowering the amount.

## 5. Experimental Evaluation

We show the metrics used to evaluate the attacks, the results concerning the selection of the Oracle, the poisoning attacks, and the regeneration process.

### 5.1. Metrics

To analyze the poisoning processes against the Fraud Detection Systems, we need to rely on specific metrics. We report the most relevant ones.
**Injection Rate**: $IR = \frac{|L|}{|F|}$, where L represents the frauds considered legitimate by the Oracle and F the fraudulent transactions proposed by the attacker.
**Detection Rate**. $DR = \frac{|D|}{|V|}$, where D represent the detected frauds and V is the set of victims.
**Average Detection Time**. $ADT = \frac{\sum_D T_d}{|D|}$, where $T_d$ is the difference between the attack start time and the detection time of the transaction.
**Money Stolen**. It specifies the amount of money that the adversary steals.

### 5.2. Poisoning Process Results

Our poisoning attacks affect 15 victims, chosen according to their spending pattern and their nationality (national or foreign). In particular, we only focus on the conservative strategy, which allows us to underline the most significant results.
Table 7 refers to White Box scenario. In the weekly update, the attacker can steal up to 10,550,761€ against an AL detector. It means that there are no direct consequences between the accuracy of the FDSs and their reaction to poisoning attacks. XGB is the detector that counters best national frauds, while LGB works well against foreign malicious transactions. However, you can notice that for the bi-weekly update we get different results. The best model against national frauds is still XGB, but LR, which is the worst against them, outperforms other models regarding foreign ones. The amount of money stolen is higher in weekly update cases, because the attacker carries on a faster poisoning process. Since the adversary can build a perfect replica of the target system, the evasion rate is always 100% while the detection rate is 0% for all models. The injection rates are always between 31.73%, achieved by XGB, and 80%, by LR. XGB pushes the attacker to regenerate the proposed frauds while LR is weaker and doesn't detect them.

White Box

| | | Metric | User | RF | XGB | LGB | CB | SVM | ANN | LR | AL |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Weekly Update | Conservative | Injection Rate (%) | Nat | 42.65 | 31.73 | 53.3 | 42.76 | 26.51 | 45.51 | 74.9 | 64.08 |
| | | | For | 21.1 | 20.24 | 17.54 | 21.26 | 23.77 | 22.43 | 43.9 | 21.76 |
| | | Detection Rate (%) | Nat | - | - | - | - | - | - | - | - |
| | | | For | - | - | - | - | - | - | - | - |
| | | Detection Time (days) | Nat | - | - | - | - | - | - | - | - |
| | | | For | - | - | - | - | - | - | - | - |
| | | Money Stolen (€) | Nat | 8,733,248 | 6,333,868 | 8,891,680 | 8,718,549 | 5,554,781 | 8,124,589 | 11,258,042 | 10,550,761 |
| | | | For | 518,874 | 260,121 | 200,569 | 268,075 | 301,606 | 201,439 | 31,323 | 351,496 |

| | | Metric | User | RF | XGB | LGB | CB | SVM | ANN | LR | AL |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Bi-weekly Update | Conservative | Injection Rate (%) | Nat | 38.59 | 34.89 | 51.61 | 42.76 | 37.36 | 50.76 | 80.06 | 51.77 |
| | | | For | 26.43 | 25.71 | 23.44 | 21.26 | 31.82 | 25.78 | 40.0 | 26.43 |
| | | Detection Rate (%) | Nat | - | - | - | - | - | - | - | - |
| | | | For | - | - | - | - | - | - | - | - |
| | | Detection Time (days) | Nat | - | - | - | - | - | - | - | - |
| | | | For | - | - | - | - | - | - | - | - |
| | | Money Stolen (€) | Nat | 2,073,919 | 1,675,701 | 2,191,912 | 1,987,482 | 974,924 | 2,089,803 | 2,674,183 | 2,143,119 |
| | | | For | 175,240 | 103,973 | 95,677 | 84,006 | 120,677 | 97,764 | 69,246 | 93,702 |

Table 4: White Box Attacks

Table 5 refers to Grey Box attacks. For what concerns the standard (i.e., which poisons both count and amount) strategy against a machine with a weekly update, the detection rates are between 27% and 63% for national users and between 0% and 50% for foreign ones. The detection time is reasonably high (from 43 to 51.5 days) and the amount of stolen money is almost 0.25 with respect to the White Box scenario. About the bi-weekly policy, the results of national users are similar to those related to the White Box scenario. Our Oracle is more restrictive about foreign transactions and allows the attacker to be undetected in some cases, such as XGB and CB. In general, the injection rates are low (between 1.83% and 17.61%), because the Oracle pushes the adversary to regenerate the features very frequently.
Through the amount strategy, the attacker can steal an amount of money slightly lower than the standard strategy, but he or she is capable to decrease the attack detection rate consistently. Poisoning just one feature makes the attacks more evasive and effective. In addition, we found

out that for foreign transactions, this strategy is much more powerful, because you are able to increase the amount stolen and decrease the detection rate. This is true for every target system and each update policy. In the bi-weekly update, this is more evident: an attacker is able to steal 53,909€ from foreign users against XGB, which is more than the standard conservative strategy against XGB trained according to a weekly policy (30,218€).

Grey Box

| | | Metric | User | RF | XGB | LGB | CB | SVM | ANN | LR | AL |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Weekly Update | Conservative | Injection Rate (%) | Nat | 6.55 | 7.24 | 9 | 5.52 | 7.71 | 6.01 | 6.78 | 6.8 |
| | | | For | 3.29 | 3.67 | 4.55 | 1.83 | 2.29 | 5.12 | 5.81 | 3.21 |
| | | Detection Rate (%) | Nat | - | 27.27 | 63.64 | 27.27 | 72.72 | 63.64 | 27.27 | 54.55 |
| | | | For | - | - | 50 | - | 50 | - | 50 | - |
| | | Detection Time (days) | Nat | - | 47 | 43 | 46.47 | 38.5 | 52 | 45 | 51.5 |
| | | | For | - | - | 15.5 | - | 16 | - | 33.5 | - |
| | | Money Stolen (€) | Nat | 2,178,902 | 1,935,610 | 1,219,945 | 1,981,969 | 968,439 | 1,311,237 | 2,286,037 | 1,952,087 |
| | | | For | 41,572 | 30,218 | 23,290 | 38,560 | 35,915 | 27,450 | 29,779 | 33,756 |
| | | Metric | | RF | XGB | LGB | CB | SVM | ANN | LR | AL |
| Bi-weekly Update | Conservative | Injection Rate (%) | Nat | 16.22 | 17.53 | 17.81 | 17.36 | 17.01 | 17.12 | 17.27 | 16.45 |
| | | | For | 10.91 | 11.43 | 8.51 | 9.29 | 9.92 | 8.02 | 13.95 | 8.57 |
| | | Detection Rate (%) | Nat | 9.09 | 9.09 | 45.45 | 9.09 | 54.54 | 9.09 | - | - |
| | | | For | - | - | 25 | - | 25 | - | 50 | - |
| | | Detection Time (days) | Nat | 60 | 16 | 50 | 58 | 46 | 59 | - | - |
| | | | For | - | - | 56 | - | 50 | - | 30.5 | - |
| | | Money Stolen (€) | Nat | 1,525,357 | 1,016,85 | 972,268 | 1,316,138 | 996,755 | 1,002,137 | 1,420,298 | 1,386,852 |
| | | | For | 49,571 | 45,036 | 40,676 | 44,993 | 47,392 | 46,543 | 31,491 | 47,843 |
| | | Metric | User | RF | XGB | LGB | CB | SVM | ANN | LR | AL |
| Weekly Update | Conservative Am. | Injection Rate (%) | Nat | 22.64 | 24.19 | 23.68 | 22.03 | 24.54 | 22.22 | 23.46 | 22.31 |
| | | | For | 10.23 | 15.12 | 9.52 | 10.23 | 14.77 | 11.73 | 13.85 | 9.2 |
| | | Detection Rate (%) | Nat | 9.09 | 45.45 | 72.73 | 9.09 | 63.64 | 27.27 | - | 27.27 |
| | | | For | - | - | 50 | - | - | 25 | 25 | - |
| | | Detection Time (days) | Nat | 57 | 43.4 | 57.5 | 59 | 33.71 | 42.5 | - | 39 |
| | | | For | - | - | 33.5 | - | - | 7 | 10 | - |
| | | Money Stolen (€) | Nat | 1,923,720 | 1,584,052 | 1,788,769 | 1,938,419 | 960,885 | 1,114,234 | 2,014,528 | 1,612,753 |
| | | | For | 43,628 | 43,745 | 39,175 | 48,188 | 59,955 | 42,178 | 35,903 | 32,113 |
| | | Metric | | RF | XGB | LGB | CB | SVM | ANN | LR | AL |
| Bi-weekly Update | Conservative Am. | Injection Rate (%) | Nat | 28.76 | 28.77 | 28.92 | 27.41 | 27.49 | 27.78 | 27.66 | 27.91 |
| | | | For | 27.91 | 20.73 | 18 | 26.14 | 25 | 25.51 | 21.95 | 15.85 |
| | | Detection Rate (%) | Nat | - | - | 18.18 | - | 9.09 | - | - | 9.09 |
| | | | For | - | - | 25 | - | 50 | - | - | - |
| | | Detection Time (days) | Nat | - | - | 45.45 | - | 3 | - | - | 60 |
| | | | For | - | - | 44 | - | 30 | - | - | - |
| | | Money Stolen (€) | Nat | 1,119,353 | 1,025,160 | 864,941 | 1,174,870 | 910,211 | 1,023,341 | 1,188,937 | 1,147,223 |
| | | | For | 45,535 | 53,909 | 41,117 | 40,368 | 40,181 | 49,512 | 52,872 | 49,021 |
| | | Metric | User | RF | XGB | LGB | CB | SVM | ANN | LR | AL |
| Weekly Update | Conservative Count | Injection Rate (%) | Nat | 6.12 | 5.83 | 5.55 | 5.85 | 7.72 | 5.98 | 6.81 | 6.19 |
| | | | For | 10.76 | 11.47 | 9.83 | 7.34 | 15.62 | 10.66 | 11.9 | 8.26 |
| | | Detection Rate (%) | Nat | - | 18.18 | 54.54 | 36.36 | 72.73 | 36.36 | 27.27 | 9.09 |
| | | | For | - | - | 50 | - | 75 | 25 | 50 | - |
| | | Detection Time (days) | Nat | - | 25.5 | 45.25 | 47.75 | 42.25 | 52.22 | 43.33 | 49 |
| | | | For | - | - | 38 | - | 18.67 | 21.5 | 42.5 | - |
| | | Money Stolen (€) | Nat | 1,250,611 | 1,172,173 | 1,150,799 | 1,231,771 | 851,342 | 921,433 | 1,170,488 | 1,186,580 |
| | | | For | 20,322 | 17,114 | 11,506 | 14,571 | 9,556 | 10,782 | 14,016 | 13,464 |
| | | Metric | | RF | XGB | LGB | CB | SVM | ANN | LR | AL |
| Bi-weekly Update | Conservative Count | Injection Rate (%) | Nat | 15.94 | 16.84 | 16.37 | 16.28 | 15.61 | 16.34 | 16.67 | 16.28 |
| | | | For | 18.51 | 17.86 | 21.45 | 19.29 | 30.56 | 20.15 | 20.18 | 20 |
| | | Detection Rate (%) | Nat | - | 9.09 | 9-09 | - | 27.27 | - | 9.09 | - |
| | | | For | - | - | 25 | - | 100 | - | 25 | - |
| | | Detection Time (days) | Nat | - | 48 | 43 | - | 21.33 | - | 60 | - |
| | | | For | - | - | 12 | - | 26.5 | - | 30 | - |
| | | Money Stolen (€) | Nat | 728,472 | 752,212 | 709,332 | 713,060 | 714,761 | 752,890 | 765,419 | 755,390 |
| | | | For | 25,498 | 24,372 | 21,107 | 26,482 | 14,122 | 16,433 | 22,069 | 22,698 |

Table 5: Grey Box Attacks

Looking at the results of the poisoning count strategy, we notice that it doesn't bring any advantages to the fraudster. The amount of money stolen is always less than the two previous strategies, especially for the weekly update and foreign transactions. However, this type of attack allows the attacker to decrease the detection rate against some models, such as XGB.

Table 6 refers to the Black Box scenario, in which the attacker trains the Oracle with just 50 features and chooses a weekly policy as update policy to make the poisoning process faster. Concerning the standard strategy and detectors with a weekly update policy, we can state that the results are worse than those of the Grey Box. This is why the attacker has a weaker Oracle and he or she adopts a weekly policy that makes him or her more suspicious. However, the update policy used by the adversary is beneficial for foreign frauds crafted against some detectors, such as CB (56,737€ vs 38,560€). SVM and LR are completely resistant to foreign frauds. This result confirms that SVM and LR are the most powerful models against not national frauds. Moreover, SVM is the model from which the adversary steals the minimum amount of money. We obtain better results with models trained with a bi-weekly update policy. The adversary can steal more money with respect to the Grey Box scenario. This is not true for LGB, from which an attacker steals less money, 733,666€ against 972,268€. However, the attack detection rates are higher, since the adversary adopts a weekly update policy: the RF model detects 45.45% of national frauds crafted according to a greedy strategy, whereas 36.36% when trained with a weekly update policy. Regarding foreign fraudulent transactions, detectors behave very differently.

Black Box

| | | Metric | User | RF | XGB | LGB | CB | SVM | ANN | LR | AL |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Weekly Update | Conservative | Injection Rate (%) | Nat | 13.1 | 15.54 | 14.16 | 13.55 | 13.48 | 14.33 | 14.01 | 14.32 |
| | | | For | 7.12 | 6.9 | 5.88 | 9.59 | 0 | 7.21 | 0 | 5.91 |
| | | Detection Rate (%) | Nat | - | 27.27 | 45.45 | 36.36 | 36.36 | 36.36 | 27.28 | 54.55 |
| | | | For | - | 50 | 50 | - | 100 | 50 | 100 | 25 |
| | | Detection Time (days) | Nat | - | 30.67 | 42.4 | 49 | 13.75 | 52.24 | 35.33 | 57.17 |
| | | | For | - | 10 | 16.5 | - | 0 | 25 | 0 | 31 |
| | | Money Stolen (€) | Nat | 1,658,038 | 1,022,444 | 849,368 | 1,428,393 | 491,297 | 1,005,478 | 1,592,290 | 1,526,330 |
| | | | For | 51,326 | 29,419 | 26,435 | 56,737 | 0 | 27.720 | 0 | 21,505 |
| | | Metric | | RF | XGB | LGB | CB | SVM | ANN | LR | AL |
| Bi-weekly Update | Conservative | Injection Rate (%) | Nat | 13.63 | 15 | 19.05 | 14.35 | 14.27 | 16.39 | 13.29 | 13.38 |
| | | | For | 11.86 | 8.93 | 11.11 | 8.64 | 0 | 9.51 | 0.1 | 5.77 |
| | | Detection Rate (%) | Nat | 27.27 | 45.45 | 81.82 | 36.36 | 54.55 | 36.36 | 18.18 | 36.36 |
| | | | For | - | 50 | 50 | 50 | 100 | 50 | 100 | 25 |
| | | Detection Time (days) | Nat | 51 | 33.8 | 37.22 | 41.5 | 25.5 | 54.65 | 23.5 | 52.5 |
| | | | For | - | 6.5 | 16 | 26 | 0 | 31 | 3.5 | 44 |
| | | Money Stolen (€) | Nat | 1,530,935 | 1,212,172 | 733,666 | 1,497,813 | 505,888 | 1,170,102 | 1,658,829 | 1,530,502 |
| | | | For | 50,763 | 54,100 | 50,150 | 31,821 | 0 | 30,402 | 201 | 23,472 |
| | | Metric | User | RF | XGB | LGB | CB | SVM | ANN | LR | AL |
| Weekly Update | Conservative Am. | Injection Rate (%) | Nat | 22.16 | 24.44 | 24.89 | 24.91 | 23.95 | 24.56 | 22.21 | 24.17 |
| | | | For | 10.26 | 15.62 | 14.29 | 11.54 | 12.44 | 13.71 | 12.19 | 11.62 |
| | | Detection Rate (%) | Nat | 18.18 | 63.64 | 81.82 | 45.45 | 72.73 | 36.36 | 18.18 | 36.36 |
| | | | For | - | 75 | 50 | - | - | 25 | 25 | 25 |
| | | Detection Time (days) | Nat | 60 | 46.29 | 40.44 | 60 | 30.31 | 38.82 | 21 | 37.5 |
| | | | For | - | 16.67 | 23.5 | - | - | 51.5 | 15 | 50 |
| | | Money Stolen (€) | Nat | 1,544,023 | 1,061,736 | 769,669 | 1,566,890 | 781,309 | 997,552 | 1,762,124 | 1,409,019 |
| | | | For | 65,805 | 53,278 | 42,939 | 82,181 | 62,507 | 48,920 | 32,331 | 40,915 |
| | | Metric | | RF | XGB | LGB | CB | SVM | ANN | LR | AL |
| Bi-weekly Update | Conservative Amount | Injection Rate (%) | Nat | 22.5 | 23.83 | 26.2 | 22.76 | 21.74 | 25.74 | 23.36 | 21.91 |
| | | | For | 11.54 | 13.89 | 16.67 | 11.76 | 0 | 12.93 | 0 | 13.79 |
| | | Detection Rate (%) | Nat | 63.64 | 54.55 | 81.82 | 36.36 | 72.73 | 54.55 | 36.36 | 36.36 |
| | | | For | - | 50 | 50 | 25 | 100 | 25 | 100 | 25 |
| | | Detection Time (days) | Nat | 56.71 | 36 | 38.89 | 38.25 | 33.62 | 40.5 | 37.5 | 34.75 |
| | | | For | - | 3 | 24 | 38 | 0 | 58 | 0 | 24 |
| | | Money Stolen (€) | Nat | 1,586,556 | 1,007,285 | 739,457 | 1,767,987 | 880,111 | 1,110,561 | 1,511,972 | 1,568,341 |
| | | | For | 67,588 | 77,720 | 80,946 | 114,075 | 0 | 69,744 | 0 | 92,788 |

Table 6: Black Box Attacks

The poisoning amount strategy is very beneficial against foreign victims. Considering a CB detector trained according to a bi-weekly policy, an attacker steals 114,075€ from foreign users, while in White Box just 84,006€. White Box attacks represent the best case possible, but with this approach, the attacker is able to outperform it.

### 5.3. Regeneration Process Results

Table 7 shows the results of the regeneration process. In the White Box scenario, each detector shows a particular behavior. Regarding the weekly update, RF requires the regeneration of the IBAN only 17.14% of the total number of national frauds, while we notice a 72.47% when dealing with foreign ones. This happens because RF, like the other models, gives more importance to the IBAN when evaluating foreign transactions.

White Box

| | | Feature | User | RF | XGB | LGB | CB | SVM | ANN | LR | AL |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Weekly Update | Conservative | IP (%) | Nat | 57.24 | 67.34 | 46.38 | 48.57 | 55.31 | 47.76 | 24.49 | 30.91 |
| | | | For | 77.98 | 79.16 | 81.87 | 77.77 | 67.27 | 80.79 | 56.01 | 75.64 |
| | | IBAN (%) | Nat | 17.14 | 42.85 | 33.71 | 22.04 | 5.30 | 35.72 | 6.73 | 11.43 |
| | | | For | 72.47 | 70.23 | 73.09 | 74.39 | 10.02 | 72.13 | 43.9 | 66.32 |
| | | CC_ASN (%) | Nat | 44.18 | 67.34 | 43.98 | 46.02 | 61.90 | 44.46 | 25.10 | 28.77 |
| | | | For | 78.44 | 79.16 | 81.87 | 77.29 | 68.18 | 80.34 | 56.09 | 75.64 |
| | | Amount (%) | Nat | 57.34 | 68.26 | 46.70 | 57.24 | 62.63 | 51.17 | 25.10 | 35.92 |
| | | | For | 78.89 | 79.26 | 82.45 | 78.74 | 68.18 | 81.16 | 56.09 | 78.24 |

Grey Box

| | | Feature | User | Detectors | | Feature | User | Detectors | | Feature | User | Detectors |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Weekly Update | Conservative | IP (%) | Nat | 87.98 | Conservative Am. | IP (%) | Nat | 71.97 | Conservative Count | IP (%) | Nat | 91.49 |
| | | | For | 82.56 | | | For | 54.65 | | | For | 74.77 |
| | | IBAN (%) | Nat | 81.13 | | IBAN (%) | Nat | 73.45 | | IBAN (%) | Nat | 85.43 |
| | | | For | 63.76 | | | For | 45.44 | | | For | 67.88 |
| | | CC_ASN (%) | Nat | 5.38 | | CC_ASN (%) | Nat | 4.11 | | CC_ASN (%) | Nat | 7.09 |
| | | | For | 16.51 | | | For | 12.79 | | | For | 4.13 |
| | | Amount (%) | Nat | 83.72 | | Amount (%) | Nat | 61.06 | | Amount (%) | Nat | 81.93 |
| | | | For | 96.33 | | | For | 84.88 | | | For | 88.53 |

Black Box

| | | Feature | User | Detectors | | Feature | User | Detectors | | Feature | User | Detectors |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Weekly Update | Conservative | IP (%) | Nat | 74.78 | Conservative Am. | IP (%) | Nat | 63.70 | Conservative Count | IP (%) | Nat | 78.19 |
| | | | For | 62.06 | | | For | 34.38 | | | For | 27.27 |
| | | IBAN (%) | Nat | 74.48 | | IBAN (%) | Nat | 69.26 | | IBAN (%) | Nat | 79.20 |
| | | | For | 62.06 | | | For | 40.63 | | | For | 26.25 |
| | | CC_ASN (%) | Nat | 0 | | CC_ASN (%) | Nat | 0 | | CC_ASN (%) | Nat | 0 |
| | | | For | 0 | | | For | 0 | | | For | 0 |
| | | Amount (%) | Nat | 62.27 | | Amount (%) | Nat | 47.40 | | Amount (%) | Nat | 49.87 |
| | | | For | 93.10 | | | For | 84.37 | | | For | 80.32 |

Table 7: Regeneration Process Results

This concept can be also applied to the other features: foreign frauds are always more suspicious, so the adversary needs to regenerate the features more frequently, including, if necessary, the amount.

In the Grey Box scenario, our Oracle often suggests changing the IP and IBAN for national frauds, while it hints to regenerate the CC_ASN and the amount for foreign ones. For each feature, the percentage of regenerated transactions is higher than that of the White Box: the reason is that our Oracle is a powerful model, which tries to filter transactions so that they could be less suspicious as possible. When adopting an amount strategy, the fraudster regenerates the transactions less frequently, since he or she wants to consistently increase the average transactions' amount of the victim. On the contrary, in the count strategy, the Oracle suggests change almost always the IP and the IBAN features.

In the Black Box scenario, the attacker builds the Oracle relying on just 50 features. We have no features related to the Country Code, this is why the attacker never regenerates it.

## 6. Conclusions

We have shown how the most popular state-of-art banking detectors behave when dealing with poisoning attacks. We propose a novel approach according to which an adversary can build a very reliable oracle and manipulate in a smart way a specific set of transaction features. With our approach, we are able to steal a consistent amount of money in every scenario. In Monti's work [1], in a Grey Box scenario the adversary was capable to steal up to 551,236€ and in a Black Box scenario up to 394,239€, by attacking 30 victims. In this work, we to perform malicious transactions that amount to more than 4 million euros in a Grey Box attack and to more than 3 in a Black Box one, by defrauding 15 customers. Moreover, our detection rates are all low, for both national and foreign users, sometimes even zero. We found out that poisoning the amount is less cautious and more effective than poisoning the count, especially for foreign users. The detection time is often very high, it goes from 30 to 60 days. On the contrary, Monti's attacks lasted on average, between two weeks and a month. Beyond the poisoning attacks results, we have deeply analyzed the feature regeneration process and we have studied which are the features that the adversary has to change more frequently at each iteration.

## References

[1] Monti. "*Poisoning Attacks Against Banking Fraud Detection Systems*".

[2] Carminati *et al.* "*BankSealer: A decision support system for online banking fraud analysis and investigation*".

[3] Carminati *et al.* "*Evasion Attacks against Banking Fraud Detection Systems*".

[4] Carminati *et al.* "*Security Evaluation of a Banking Fraud Analysis System*".