2019 IEEE Second International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)

Building a robust mobile payment fraud detection system with adversarial examples

Simon Delecourt
Worldline
Lille, France
simon.delecourt@worldline.com

Li Guo
Worldline
Lille, France
li.guo@worldline.com

***Abstract*—Mobile payment is becoming a major payment method in many countries. However, the rate of payment fraud with mobile is higher than with credit card. One potential reason is that mobile data is easier to be modified than credit card data by fraudsters, which degrades our data-driven fraud detection system. Supervised learning methods are pervasively used in fraud detection. However, these supervised learning methods used in fraud detection have traditionally been developed following the assumption that the environment is benign; there are no adversaries trying to evade fraud detection system. In this paper, we took potential reactions of fraudsters into consideration to build a robust mobile fraud detection system using adversarial examples. Experimental results showed that the performance of our proposed method was improved in both benign and adversarial environments.**

***Index Terms*—fraud detection, adversarial machine learning, oversampling**

I. Introduction

Thanks to its simplicity, freeness and real-time, mobile payments are growing in popularity. Almost all of the giant IT corporations launched their mobile payment solutions, such as Google wallet, Apple pay, Samsung pay, Alipay, WeChat... In China mobile payment is expected to grow at the rate of 22 percent each year, starting from $29.7 trillion in 2017 and heading to $96.7 trillion in 2023[1]. This also attracts hackers and fraudsters. As a consequence, more and more frauds in mobile payments are being reported.

A large number of machine learning methods have been proposed for fraud detection problems [1], [2], which can be divided into supervised, unsupervised and semi-supervised. In supervised learning, we know both fraudulent and genuine transactions in the past data, we suppose the future is similar to the past. However, fraudsters are changing their strategies all the time. Supervised methods are powerless to detect new-type frauds. In unsupervised learning, We do not know neither fraudulent nor genuine transactions in the past data, but we know frauds are different from most transactions. Unsupervised methods can detect new type of frauds. But their false alert rate are generally higher than supervised methods. Semi-supervised methods try to take advantage of both supervised and unsupervised learning and they are often used in cases where there are many unlabeled data points and few labeled ones.

Although, many fraud detection methods have been introduced, most existing methods were proposed in a "benign" environment. They did not take reactions of fraudsters into consideration. They supposed that fraudsters know nothing about fraud detection systems and machine learning techniques. Unfortunately, this benign assumption is not correct. Actually, fraudsters are well organized, they are proficient in our fraud detection methods in order to avoid being detected. Especially, compared to face to face credit card payments where there are material constraints (i.e. having a functional card), mobile payments are almost fully digital, which are possible to be hacked by those professional fraudsters. In this paper, we proposed a mobile fraud detection method in a real adversarial environment. We supposed that: 1. Fraudsters know that we apply machine learning methods to fraud detection. 2. Fraudsters also use machine learning methods to create their strategies. 3. Fraudsters analyzed our fraud detection replies of their previous fraudulent transactions to improve their strategies.

Adversarial machine learning is the study of attacks on, and defenses for, machine learning systems. Nowadays, machine learning techniques are used in a variety of application domains, especially after the advent of deep learning, many impressive performances have been reported. Thus, people were astonished at the fact that our deliberately developed methods can easily be fooled by adversarial examples, when Szegedy et al [3] kidded the celebrated deep learning. Since then, many methods have been proposed to create adversarial examples, such as, FGSM, L-BFGS, JSMA [4] [5] [6]. Therefore, making machine learning methods robust to adversarial attacks became a new issue, a large amount of work has been proposed [7], [8], one of the most well-known solutions is to introduce adversarial examples at training time [9]. Many applications have also been reported [7], [10]. Mary et al [11] used a game theoretical adversarial learning approach to model the interactions between a fraudster and the fraud detection system. At each round fraud strategies are clustered with Gaussian Mixture Model. The strategies are evaluated and the best one is oversampled using SMOTE [12] creating a resampled training dataset. This enables to train the model on its weaknesses.

[1]source : https://www.mobilepaymentstoday.com/articles/chinese-economic-headwinds-raise-questions-about-mobile-payment-growth

978-1-7281-1488-0/19/$31.00 ©2019 IEEE
DOI 10.1109/AIKE.2019.00026

In this paper, we compared some of the most popular fraud detection methods in an adversarial attack condition. Then, we oversampled these adversarial examples to construct a more robust mobile payment fraud detector.

The contributions of this paper are the following :

- The first research on comparing different fraud detection methods in an adversarial attack condition.
- The first solution on mobile payment fraud detection with adversarial learning.

II. Adversarial Attack

The challenge for the adversary is to figure out how to generate an input which fools the targeted classifier [13]. Especially, in an adversary attack setting, we focus on wild patterns (adversarial examples) [14] creation methods, which confuses machine learning models. A possible strategy is to minimize distance (or small perturbation) [15]. As shown in formula 1, it generates an adversarial example x^{adv} by adding a well crafted perturbation to a existing sample $x^{'}$, and the classification results of our model $f(x)$ on this adversarial example x^{adv} and sample $x^{'}$ are different. The main idea of this strategy is to find out some "exceptions" to one of basic machine learning conditions: *similar samples, similar labels*.

$$\begin{aligned} &\min && ||x^{adv} - x^{'}|| \\ &\text{subject to} && f(x^{adv}) \neq f(x^{'}) \end{aligned} \tag{1}$$

According to different levels of adversary knowledge on target model, adversarial attack scenarios can be generally categorized into [16]:

- **White box attack**. The adversary has full knowledge of the model; such as, architecture, parameters.
- **Black box attack with probing**. The adversary does not know the model but can probe or query the model, i.e. feed some inputs and observe outputs.
- **Black box attack without probing** The adversary does not have any knowledge on the model and is not allowed to probe or query the model. In this case, the attacker must construct adversarial examples that fool most machine learning models.

A. White-box attack

In a white-box attack context, most successful attacks are gradient-based methods. One of popular methods in this category is the Fast Gradient Sign Method (FGSM). As shown in eq. 2, FGSM generates an adversarial example x^{adv} by adding a small perturbation of magnitude ϵ in the direction of the gradient of the loss function $J(x, y_{true})$ of a target model.

$$x^{adv} = x + \epsilon \cdot sign\left(\nabla_x J(x, y_{true})\right) \tag{2}$$

B. Black-box attack with probing

Szegedy et al [3] observed that the same adversarial sample is often misclassified by a variety of classifiers. This transferability property of adversarial samples is used to craft adversarial examples in the black box scenario. An attacker can firstly train his own (white box) substitute model, then generate adversarial samples and finally apply the adversarial samples to the target ML model.

III. Experimental Results

A. Dataset

We performed our experiments on a dataset containing transactions made with a payment smartphone application. Each transaction is defined by 243 features. Among them, there is information about the transaction itself (amount, time, status) as well as smartphone information (brand, OS version, year, battery power, country, operator, ...). We also have features giving information on past transactions and the overall payer use of the applications (delta-time with the last transaction/last application opening, date of enrollment, number of transaction in the last day/week/month, ...). Given that [17] pointed out that performing too much features selection leads to a robustness loss, we decided to keep most of these features. Number of instances, repartition of genuine and fraud transactions can be seen in Tab. I

	Training set	Test Set
Genuine	812 040	203 010
Frauds	55	15
Total	812 095	203 025

TABLE I
Repartition of Genuine and frauds transaction in the dataset

Since we want to study the robustness of our fraud detection system on purely numerical attacks, we assumed that all features can be manipulated by an adversary, even "physical" ones.

B. A comparison analysis of fraud detection methods under adversarial attack

Machine learning model selection [18] is the first and the most important step to a machine learning application. Many fraud detection method comparisons have been reported [19], [20]. However, all of these comparisons were made in a benign environment. Therefore, comparing different fraud detection methods in an adversarial setting should be interesting for many fraud detection researchers.

In this study, we supposed that our fraud detection methods are under black box with probing attack. A simple Multi-Layer Perceptron (MLP) is served as attackers' substitute model. We selected MLP, Logistic Model (LR), Decision Tree (DT) and Random Forest (RF), 4 of the most popular machine learning methods as our fraud detection models (oracle models). A Jacobian dataset augmentation [21] method was applied by attackers. We supposed that attackers have already collected 20 frauds and 80 normal transactions as S_0, but they did not know their real labels. Attackers queried our oracle models for classification results $\tilde{O}(x)$ on these transactions in S_0, then, a MLP was constructed on these labeled data as attackers' substitute model. The choice of MLP to subsitute the oracle model is motivated by its property to learn online as new

	Substitute	Substitute attacked	Oracle	Oracle attacked
MLP	0.261 ± 0.10	0.161 ± 0.07	0.283 ± 0.11	0.244 ± 0.13
LR	0.235 ± 0.11	0.116 ± 0.08	0.295 ± 0.10	0.269 ± 0.10
DT	0.212 ± 0.10	0.084 ± 0.09	0.138 ± 0.10	0.032 ± 0.04
RF	0.226 ± 0.10	0.122 ± 0.10	0.113 ± 0.09	0.150 ± 0.09

TABLE II
PRECISION AT 100 METRICS COMPARED ACROSS DIFFERENT MODELS. WHERE MLP IS MULTI LAYER PERCEPTRON, LR IS LOGISTIC REGRESSION, DT IS DECISION TREE AND RF IS RANDOM FOREST

data is retrieved. After that, attackers will generate adversarial instances with their substitute model and query our oracle models for output labels on these adversarial instances to augment their dataset and optimize their attack models and strategies.

Algorithm 1 Jacobian dataset augmentation [21]

1) Collecting some (limited) data S_p, where p is number of iterations
2) Querying Oracle labels $(\tilde{O}(x), \forall x \in S_p)$
3) Training the substitute model (MLP) on these labeled data $(S_p, \tilde{O}(S_p))$
4) Augmenting data S_{p+1} according to equation (3)

$$S_{p+1} = S_p \cup \{x + \epsilon \cdot sign(\nabla_x J[x, \tilde{O}(x)]) : x \in Sp\} \quad (3)$$

Performing this kind of data augmentation is a way to query labels on point near the decision boundaries. Thus, with limited queries, it is possible to substitute the fraud detection system.

For our experiments we performed 3 iterations for Jacobian dataset augmentation with an initial dataset of 20 frauds among 100 instances. We used the FGSM with an ϵ equals to 0.01 for all features in range $[0; 1]$. We used FGSM for convenience as it is easy to compute.

Results can be seen in Table. II. First column shows how well the substitute model succeeds in imitating the oracle. Second column indicates the performance of the substitute model under attack. The third column indicates the performance of the oracle in a benign environment. While the last column indicates the performance of the oracle on examples crafted thanks to the substitute model.

Results show that most models are likely to be fooled by black-box attacks. Indeed we observe a significant precision loss for the oracle between the benign and adversarial environments. We observe an exception for random forest where we notice a weird phenomenon. Indeed the substitute model seems to be better than the detection system itself and the FGSM is not able to craft good adversarial examples. We believe that in general ensemble models are more robust to adversarial attacks than other methods. Especially when each classifier of the ensemble is trained with a different subset of features. This result should motivate more research in this topic.

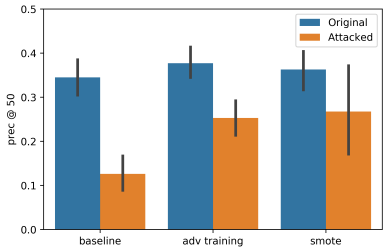

Fig. 1. Comparison of precision and robustness between oversampling methods using adversarial attack and SMOTE

C. Improving the robustness of fraud detection methods with adversarial instances

As mentioned before, adversarial instances of a ML model are some instances difficult correctly classified by this model. Obviously, one could add these generated adversarial instances into the training set to improve the robustness of a ML model. In addition, one major problem of a fraud detection model is the extreme imbalanceness of datasets. In this part, we used adversarial example generation method as oversampling method to balance our fraud detection data and improve the performance of fraud detection methods.

We compare our adversarial oversampling with SMOTE oversampling. We used the same training procedure for both of them. At each epoch, the training set is oversampled in order to double the number of fraud transactions. The oversampled instances of the previous epoch are dropped in order to assure that the oversampled instances remain in a box-constraint. We also compared with a baseline : the same MLP model trained with the raw training set. Each model created is evaluated with precision at 50 on the test and precision at 50 on adversarial examples using FGSM with $\epsilon = 0.01$.

As shown in Fig. 1. We observe that performing an oversampling technique closes the gap between precision and precision under attack leading to more robust models, adversarial training technique being better than SMOTE. We also spot a little gain in precision on the test set whereas SMOTE makes no difference with baseline.

We explain our results by the fact that in the context of extreme imbalanced fraud dataset, SMOTE can't operate properly. Indeed, SMOTE performs a linear combination of nearest points, thus for each fraud instance it must have K other close frauds which is not always true due to fraud label sparsity. Moreover, there is no guarantee that SMOTE synthesized points will help the model to reach the task decision boundary. As for adversarial oversampling, no strong assumption is made. Above all, it helps to push back the decision boundary towards the task decision boundary (i.e. the theoretical decision boundary for the task) by anticipating fraudsters next moves.

An example using a toy dataset can be seen in Fig. 2. This figure illustrates the strategy of fraudsters to transgress our fraud detection system and shows how our algorithm behaves

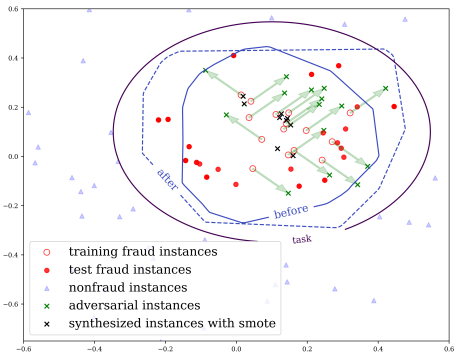

Fig. 2. Comparison between adversarial oversampling and SMOTE

to anticipate it and leaves no clues for fraudster on which strategy to choose next.

IV. Conclusion and Future Work

In this work, we studied the effect of adversarial attacks on a fraud detection system in a context of a smartphone payment application. A comparison of different mobile payment fraud detection methods in an adversarial setting was discussed. We showed that fraud detection systems, as other machine learning techniques are subject to adversarial attacks. We also proposed to use adversarial examples to improve the robustness of fraud detection models and to balance the training data. Experimental results showed that performance of our proposed method was improved in both benign and adversarial environments. In the future, we would like to extend this work towards implementation of a more general framework to evaluate robustness of fraud detection system in an adversarial environment.

References

[1] K. K. Tripathi and M. A. Pavaskar, "Survey on credit card fraud detection methods," *International Journal of Emerging Technology and Advanced Engineering*, vol. 2, no. 11, pp. 721–726, 2012.

[2] A. Srivastava, A. Kundu, S. Sural, and A. Majumdar, "Credit card fraud detection using hidden markov model," *IEEE Transactions on dependable and secure computing*, vol. 5, no. 1, pp. 37–48, 2008.

[3] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[4] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 39–57, IEEE, 2017.

[5] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," *arXiv preprint arXiv:1607.02533*, 2016.

[6] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017.

[7] K. Grosse, N. Papernot, P. Manoharan, M. Backes, and P. McDaniel, "Adversarial examples for malware detection," in *European Symposium on Research in Computer Security*, pp. 62–79, Springer, 2017.

[8] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *2016 IEEE Symposium on Security and Privacy (SP)*, pp. 582–597, IEEE, 2016.

[9] M. Barreno, B. Nelson, R. Sears, A. D. Joseph, and J. D. Tygar, "Can machine learning be secure?," in *Proceedings of the 2006 ACM Symposium on Information, computer and communications security*, pp. 16–25, ACM, 2006.

[10] K. Grosse, N. Papernot, P. Manoharan, M. Backes, and P. McDaniel, "Adversarial perturbations against deep neural networks for malware classification," *arXiv preprint arXiv:1606.04435*, 2016.

[11] M. F. Zeager, A. Sridhar, N. Fogal, S. Adams, D. E. Brown, and P. A. Beling, "Adversarial learning in credit card fraud detection," in *2017 Systems and Information Engineering Design Symposium (SIEDS)*, pp. 112–116, IEEE, 2017.

[12] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[13] I. Goodfellow, P. McDaniel, and N. Papernot, "Making machine learning robust against adversarial inputs," *Communications of the ACM*, vol. 61, no. 7, pp. 56–66, 2018.

[14] B. Biggio and F. Roli, "Wild patterns: Ten years after the rise of adversarial machine learning," *Pattern Recognition*, vol. 84, pp. 317–331, 2018.

[15] A. Kantchelian, J. Tygar, and A. Joseph, "Evasion and hardening of tree ensemble classifiers," in *International Conference on Machine Learning*, pp. 2387–2396, 2016.

[16] A. Kurakin, I. Goodfellow, S. Bengio, Y. Dong, F. Liao, M. Liang, T. Pang, J. Zhu, X. Hu, C. Xie, *et al.*, "Adversarial attacks and defences competition," in *The NIPS'17 Competition: Building Intelligent Systems*, pp. 195–231, Springer, 2018.

[17] F. Zhang, P. P. Chan, B. Biggio, D. S. Yeung, and F. Roli, "Adversarial feature selection against evasion attacks," *IEEE transactions on cybernetics*, vol. 46, no. 3, pp. 766–777, 2016.

[18] S. Arlot, A. Celisse, *et al.*, "A survey of cross-validation procedures for model selection," *Statistics surveys*, vol. 4, pp. 40–79, 2010.

[19] Z. Zojaji, R. E. Atani, A. H. Monadjemi, *et al.*, "A survey of credit card fraud detection techniques: Data and technique oriented perspective," *arXiv preprint arXiv:1611.06439*, 2016.

[20] C. Phua, V. Lee, K. Smith, and R. Gayler, "A comprehensive survey of data mining-based fraud detection research," *arXiv preprint arXiv:1009.6119*, 2010.

[21] N. Papernot, P. McDaniel, and I. Goodfellow, "Transferability in machine learning: from phenomena to black-box attacks using adversarial samples," *arXiv preprint arXiv:1605.07277*, 2016.

[22] D. Sánchez, M. Vila, L. Cerda, and J.-M. Serrano, "Association rules applied to credit card fraud detection," *Expert systems with applications*, vol. 36, no. 2, pp. 3630–3640, 2009.

[23] S. Maes, K. Tuyls, B. Vanschoenwinkel, and B. Manderick, "Credit card fraud detection using bayesian and neural networks," in *Proceedings of the 1st international naiso congress on neuro fuzzy technologies*, pp. 261–270, 2002.

[24] G. L. Wittel and S. F. Wu, "On attacking statistical spam filters.," in *CEAS*, 2004.

[25] A. Shen, R. Tong, and Y. Deng, "Application of classification models on credit card fraud detection," in *2007 International Conference on Service Systems and Service Management*, pp. 1–4, IEEE, 2007.

[26] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *2008 Eighth IEEE International Conference on Data Mining*, pp. 413–422, IEEE, 2008.

[27] R. Tomsett, A. Widdicombe, T. Xing, S. Chakraborty, S. Julier, P. Gurram, R. Rao, and M. Srivastava, "Why the failure? how adversarial examples can provide insights for interpretable machine learning," in *2018 21st International Conference on Information Fusion (FUSION)*, pp. 838–845, IEEE, 2018.

[28] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[29] S. Baluja and I. Fischer, "Adversarial transformation networks: Learning to generate adversarial examples," *arXiv preprint arXiv:1703.09387*, 2017.

[30] S. Gu and L. Rigazio, "Towards deep neural network architectures robust to adversarial examples," *arXiv preprint arXiv:1412.5068*, 2014.

[31] A. Mead, T. Lewris, S. Prasanth, S. Adams, P. Alonzi, and P. Beling, "Detecting fraud in adversarial environments: A reinforcement learning approach," in *2018 Systems and Information Engineering Design Symposium (SIEDS)*, pp. 118–122, IEEE, 2018.