Cost-Sensitive Decision Making for Online Fraud Management

Mehmet Yigit Yildirim, Mert Ozer, Hasan Davulcu

Cost-Sensitive Decision Making for Online Fraud Management

Mehmet Yigit Yildirim, Mert Ozer, and Hasan Davulcu

Arizona State University, Tempe AZ 85281, USA
{yigityildirim,mozer,hdavulcu}@asu.edu

Abstract. Every online transaction comes with a risk and it is the merchant's liability to detect and stop fraudulent transactions. Merchants utilize various mechanisms to prevent and manage fraud such as automated fraud detection systems and manual transaction reviews by expert fraud analysts. Many proposed solutions mostly focus on fraud detection accuracy and ignore financial considerations. Also, highly effective manual review process is overlooked. We propose Profit Optimizing Neural Risk Manager (PONRM), a decision maker that (a) constitutes optimal collaboration between machine learning models and human expertise under industrial constraints, (b) is cost and profit sensitive. We suggest directions on how to characterize fraudulent behavior and assess the risk of a transaction. We show that our framework outperforms cost-sensitive and cost-insensitive baselines on three real-world merchant datasets.

Keywords: fraud detection, cost-sensitive learning, risk management, e-commerce

1 Introduction

In 2016, card fraud cost businesses over $20 billion and it is still continuing to grow dramatically [4]. Around 60% of this loss was caused by online transactions, as e-commerce fraud rates have doubled since last year. E-commerce fraud magnitude is estimated to reach $71 billion during the next five years due to the steady rise in cost per fraudulent transaction while fraud rates continue to increase [13].

During fraud management, merchants are generally liable for paying for the fraud costs in the e-commerce ecosystem. They suffer the losses arising from shipped merchandise, shipping and handling costs alongside chargeback fees issued by the card processor [17]. Lexis Nexis reports that for every dollar of loss, merchants end up losing $2.40 on average as fraud management costs [14]. When aggregated, they lose around 1.5 percent of their total revenue to fraud today - three times increase during the last 3 years. So, they implement various strategies to fight fraud from automated fraud prevention systems to manual order reviews by expert fraud analysts [6].

One may think that manual reviews will be going away with advances in artificial intelligence; however, they remain very much relevant to the industry thanks to their accuracy. According to CyberSource, manual review is an established mechanism for fraud prevention with adoption by 79% of North American businesses [7]. Despite all efforts to fight fraud, significant improvements can still be made by investigating and answering the following questions: What are the most important characteristics of a fraudulent transaction that a merchant can capture without causing friction? As state-

of-the-art machine learning algorithms are not perfect how should a merchant use them? What is the cost optimal role of expert manual reviews in this process?

Improving fraud prevention is not as straightforward as increasing fraud detection accuracy due to several factors: firstly, rejecting a legitimate order and approving a fraudulent transaction do not incur the same cost, secondly, transaction amount varies greatly by order, thus affecting profitability of a sale. Hence, merchants need to implement cost and profit sensitive fraud prevention strategies.

In this work, we introduce Profit Optimizing Neural Risk Manager (PONRM), a cost-sensitive decision maker for e-commerce fraud management. Our framework infers the risk of a transaction being fraud and combines it with the transaction amount to make an optimal decision regarding its fraud management strategy (i.e. automated accept, reject or manual review). The main contributions of our work are:

- A cost-sensitive decision making framework to manage fraud while maximizing profits and minimizing costs;
- A transaction risk model incorporating fraud characteristics and financial constraints relevant to a merchant;
- An optimal collaboration strategy between human experts and machine learning models for fraud management

2 Related Work

Fraud detection has been an active area for data mining researchers since 1994 [10]; however, it has not been extensively studied due to private and confidential nature of financial data. Despite these limitations, researchers managed to conduct studies with industry partners on proprietary datasets. Major studies focusing on credit card fraud include: [3], [16], and [21. In related areas, product review fraud detection work [12] have also received attention. Theoretical contributions focusing on fraud detection applications such as [24] and [25] are also made. Survey papers on fraud detection methods include [20] and [18].

Although fraud loss is an enormous problem for e-commerce merchants, there is only a pair of studies [11] and [5] investigating this problem from a merchant's perspective. However, these works aim to improve the accuracy of fraud detection alone, instead of developing a profit and loss aware fraud management strategy.

Fraud prevention teams must take various complications that arise from allowing or rejecting a transaction into account. Declining a legitimate transaction would often result in a loss of that customer's business whereas approving a fraudulent transaction would force the merchant to cover the fraud costs. Simply training a machine learning classifier by overlooking various costs leads to a less than optimal fraud management strategy. Researchers have been developing cost-sensitive learning frameworks including [8], [9], [22], [2], and [15].

In [5], the role of manual reviews in fraud prevention process is recognized; however, authors do not provide a systematic analysis on how to integrate machine learning based detection with manual reviews under cost and capacity constraints. In this paper, we develop a cost-sensitive fraud management framework incorporating all relevant capacities, costs and evaluate its financial impact with real-world merchant datasets.

3 Problem Definition

Every online transaction comes with a risk of being fraudulent. As merchants are responsible for detecting fraud, they must take this risk into account or they would suffer from losses due to fraud. So, when a merchant receives an order it can accept, reject or manually review that transaction based on their risk assessment of that transaction. Brief explanation of each decision is as follows:

Accept: Accepting a transaction means that merchant approves the transaction and processes the payment. Accepting a legitimate transaction yields some profit. If the transaction turns out to be fraudulent, merchant becomes responsible for the dispute handling and losses.

Reject: Rejecting a transaction means that merchant declines the transaction and payment does not go through. In this case, sale does not happen, so they will not be earning a profit even if the order was legitimate. However, rejecting a legitimate transaction may cause the loss of lifetime value of the customer.

Review: In the case of sending the transaction to manual review, merchant halts the order and sends the transaction details to an expert fraud analyst for investigation. Fraud analyst would confirm the legitimacy of the order by manually analyzing the transaction details and by following-up with the consumer directly before approving or rejecting it. For the sake of our modeling, we assume that manual review always leads to correct decisions. However, expert fraud analysts are scarce and expensive resources and should be utilized wisely.

We refer to these decisions made for a set of transactions as the *fraud management strategy*. We define the task of finding an optimal fraud management strategy as follows: Given a streaming set of transactions, determine the accept, reject, and review populations to maximize profits by accepting most of the legitimate transactions; and achieve this objective by minimizing customer insults, fraud losses, and costly manual reviews.

4 Methodology

Figure 1 presents an overview of our system. It consists of two learning and a pair of data manipulation components. The workflow starts with a data preprocessing and feature extraction task. 2nd component of the system carries out the task of inferring the probability of each transaction being fraudulent. 3rd component of the system generates cost-sensitive labels. 4th and final component of the system learns a function to maximize the profit based on a criteria incorporating the transaction amount and its fraud risk probability. We call this component as Profit Optimizing Neural Risk Manager (PONRM). Each following subsection explains one component of our system in detail and their order is aligned with the numbering in Figure 1.

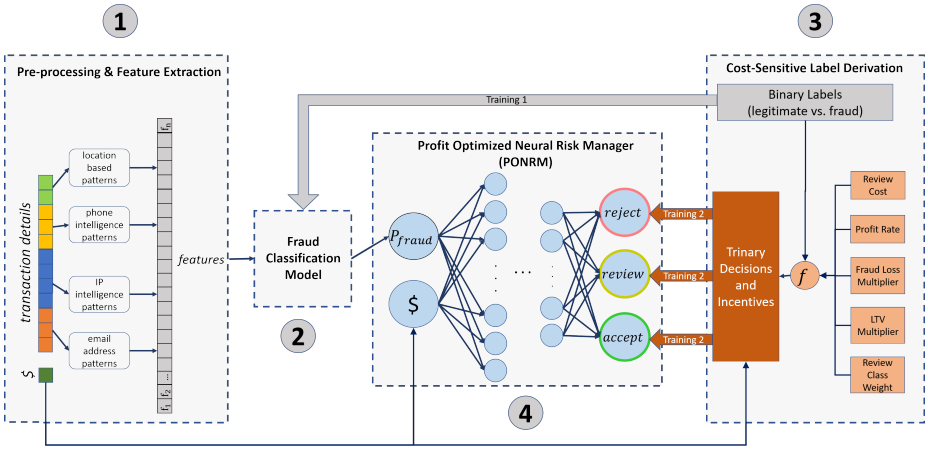


Fig. 1. System Overview

4.1 Feature Extraction

Identifying consumer behavior to detect fraud is a delicate task. Businesses are hesitant to implement multi-factor authentication systems since it can be a source of friction and collecting invasive information such as cookie mining and device fingerprinting may damage the merchant's reputation. However, it may be possible to develop fraud prevention models without above options since merchants already have access to a rich source of information about their customers: the order form. Customers provide their personal and contact information to ensure the delivery of their order, so these can be leveraged by the fraud teams to build models. We present 4 types of patterns that merchants can reproduce:

Location Based Patterns: We measure the distance between IP geolocation and physical addresses. We create risk profiles for zip codes based on historical fraud behaviors observed from corresponding districts.

Phone Intelligence Patterns: Usage of VOIP, prepaid, spoofed, or invalid phone number is detected and may indicate malicious intent. Area code of a phone number is used to verify the (in-)consistency with the physical address.

IP Intelligence Patterns: An IP address coming through a proxy or an anonymous network could indicate risky behavior. We also profile the risk based on historical fraudulent behavior observed from blocks of IPs.

Email Address Patterns: We create email domain related attributes such as existence, disposability, anonymity, tenure, and category. Informed by [23], we derive features directly from the email handle (i.e. different email address characteristics such as character diversity, typing efficiency, proportion of numbers, etc.) to determine if an email address has been created with malicious intent.

By normalizing, profiling and combining these patterns, we come up with a set of 102 features that is used in our fraud classification model.

4.2 Fraud Classification Model & Risk Score Calculation

Risk score constitutes the input of the proposed model, PONRM. It is composed of a pair of elements: first element is the transaction amount ($) and second element is a probability score of the transaction being fraudulent given its features. We propose using any supervised learner (θ) providing a robust posterior probability for fraud probability estimation such as:

$$\mathbf{f}_i = P(\mathbf{Y}_{i2} = 1 | \mathbf{X}_i; \theta)$$

where $\mathbf{f} = \{\mathbf{f}_i; \mathbf{f}_i \in [0,1] \wedge i = 1 \dots N\}$. As given in Equation [eq:f], $\mathbf{f}$ is assigned with the probability of a transaction being fraudulent. Finally, the risk score matrix $\mathbf{R}$ is built by concatenating $\mathbf{f}$ and the transaction amount ($) as;

$$\mathbf{R} = [\mathbf{f}, \$]$$

4.3 Cost-Sensitive Label Derivation

The 3rd component is concerned with the training labels that PONRM will use. Cost-sensitive models require a pair of entities to be trained with: ground-truth decisions and cost-sensitive incentives for those decisions (Elkan 2001). Possible decisions are to *accept*, *review*, and *reject* a transaction. Incentives are determined based on earnings and losses that may arise from accepting, reviewing, or rejecting.

From Binary Labels to Trinary Ground-Truth Decisions: In the ideal binary decision making process, the model would accept all legitimate and reject all fraudulent transactions. However, models often fall short in performance compared to time consuming expert manual reviews in reality. To optimally integrate highly accurate but costly manual reviews into a decision making framework, a translation from binary to trinary decisions is necessary. Weight of the review decisions should be manipulatable based on the review capacity of a merchant. Following these constraints, we translate binary (legitimate, fraudulent) labels to trinary (accept, review, reject) decisions as $[\mathbf{Z}_{i1}, \mathbf{Z}_{i2}, \mathbf{Z}_{i3}]$. After the translation, legitimate transactions become $\mathbf{Z}_i = [1, r, 0]$ while fraudulent transactions become $\mathbf{Z}_i = [0, r, 1]$ as ground-truth decisions. r is a parameter for tuning the number of review decisions vs. accept/reject decisions, proportionally.

Computing Cost-Sensitive Decision Incentives: By following the fraud management strategy considerations from Section 3, we incentivize our decisions with 4 parameters, namely: *profit rate* (pr), *lifetime value multiplier* (ltv), *fraud loss multiplier* (flm), and *review cost* (rc). *Profit rate* is defined as the percentage of the transaction amount the merchant is earning as profit. *Lifetime value multiplier* simply models the lost opportunity due to losing customer's future business when a legitimate transaction is rejected (customer insult). *Fraud loss multiplier* weights the losses due to fraudulent activity to represent associated legal and chargeback costs. Finally, *review cost* is the compensation expert manual reviewers are paid per transaction. Derivation of the incentives for each decision is presented in Table 1. Although rejecting a fraudulent transaction does not provide any benefit, it is still the most desirable decision for a fraudulent transaction. From an information theoretic perspective, there is a need for a positive scalar to

incentivize the learning process. To stay truthful to the initial incentives but represent most desirable decisions we offset the incentives: we add the initial incentive of accepting a fraudulent transaction to every decision incentive for fraudulent transactions. We add the initial incentive of rejecting a legitimate transaction to every decision incentive for legitimate transactions.

Table 1. Incentives for accepting, reviewing or rejecting a transaction

	Decision Incentives		
	Accept	**Review**	**Reject**
Legitimate	$pr * \$_i$	$pr * \$_i - rc$	$-pr * \$_i * ltv$
Legitimate - Offset	$(1 + ltv) * pr * \$_i$	$(1 + ltv) * pr * \$_i - rc$	0
Fraudulent	$-flm * \$_i$	$-rc$	0
Fraudulent - Offset	0	$flm * \$_i - rc$	$flm * \$_i$

4.4 Profit-Optimizing Neural Risk Manager

Many of the off-the-shelf classification models are cost-insensitive; thus are sub-optimal for our task. Cost of accepting a fraudulent transaction and cost of rejecting a legitimate transaction can vary largely in different settings. While these costs differ between legitimate and fraudulent cases, they are also dependent on the transaction amounts. Moreover, off-the-shelf classification tools are not very adaptable for the expert opinion to intervene when necessary.

Hence, we formally define Profit Optimizing Neural Risk Manager (PONRM) which produces decisions as accept, review, or reject for transactions according to each transaction's risk score. PONRM mostly mimics a multilayer perceptron structure with sigmoid activation functions;

$$\mathbf{R}_i = [\mathbf{f}_i, \$_i]$$
$$\mathbf{H}^{(0)} = \sigma(\mathbf{W}^{(0)}\mathbf{R} + \mathbf{b}^{(0)})$$
$$\mathbf{H}^{(i)} = \sigma(\mathbf{W}^{(i)}\mathbf{H}^{(i-1)} + \mathbf{b}^{(i)}) \qquad \text{for } i = 1, \dots, l$$
$$\widehat{\mathbf{Z}} = softmax(\mathbf{W}^{(l+1)}\mathbf{H}^{(l)} + \mathbf{b}^{(l+1)})$$

where $\mathbf{R} \in \mathbb{R}_+^{N\times 2}$ is the risk score matrix. Each $\mathbf{H}^{(i)} \in \mathbb{R}^{N\times \sqrt[i]{L}}$ is a higher dimensional($\sqrt[i]{L}$) internal representation of the risk score in the multilayer perceptron. It outputs the decisions for each transaction in the output layer $\widehat{\mathbf{Z}} \in [0,1]^{N\times 3}$. To learn the parameters of the model, we use log loss multiplied by cost sensitive incentives and minimize the loss function by tuning $\mathbf{W}^{(i)}$,$\mathbf{b}^{(i)}$:

$$Loss = -\frac{1}{N}[\sum_{i=1}^{N}\sum_{c=1}^{3}\overbrace{[\mathbf{Z}_{ic}\log\widehat{\mathbf{Z}}_{ic}]}^{log-loss}\overbrace{\widetilde{\mathbf{B}_{ic}}}^{incentive}] + \overbrace{\sum_{i=1}^{l}\alpha_i||\mathbf{W}^{(i)}||_2^2}^{regularization}$$

where N is the number of transactions. $\mathbf{Z}_{ic}$ quantifies the weight of assignment of the ground-truth decision c to the transaction i. $\widehat{\mathbf{Z}}_{ic}$ is the predicted assignments by the PONRM model for transaction i and decision c. $\mathbf{B} \in \mathbb{R}^{N\times 3}$ and $\mathbf{B}_{ic}$ quantifies the incentive of assigning the i^{th} transaction to decision c. We use L-BFGS quasi-newton optimization implementation of ScipyOptimizer interface of Tensorflow to minimize the proposed loss function [1].

5 Experiments

Here, we evaluate the performance of our framework in various settings. In the first experiment, we present the effectiveness of PONRM in comparison to other cost-sensitive and cost-insensitive approaches. Next, we evaluate the performance of our system alongside baseline risk managers under different manual review capacities. Finally, we explore how fraud classification models perform with and without risk managers.

5.1 Evaluation Metrics

We introduce a new metric, named profit gain (PG), to measure the performance of our framework and the baseline models in a financially sound way. We normalize this metric using two extreme fraud management strategies:

No Fraud Management: A merchant can choose not to interfere with any orders and accept all transactions as if they were legitimate. Then, it would suffer the maximum loss from fraudulent orders but not from any customer insults. We refer the total profit this company makes as $\$_{nofraudmanagement}$.

Oracle: If a merchant could model the fraud characteristics perfectly, it would be accepting all legitimate orders and rejecting the fraudulent ones. In this case, its fraud and customer insult loss would be zero. It would earn the profit from all the legitimate transactions. We refer its total profit as $\$_{oracle}$.

To robustly measure the financial performance gain with a standardized scoring mechanism, we introduce *profit gain* as:

$$profit\ gain = \frac{\$_m - \$_{nofraudmanagement}}{\$_{oracle} - \$_{nofraudmanagement}}$$

where $\$_m$ is the profit of the model under experimentation. While calculating the profits, not-offset decision incentives in Table 1 is used. Also, we use *F-measure* to evaluate our fraud detection performance. As we assume perfect decisions by reviewers, review decisions are treated as accept for legitimate and reject for fraudulent transactions in calculation of F-measure. Each experiment is run 16 times and the average performance is reported for each parameter setting. For each parameter configuration, the best performing setting in terms of PG is reported as the representative performance of a model.

5.2 Dataset & Parameter Settings

We work with online transactions of three e-commerce merchants; an online travel agency, a physical goods store, and a digital goods store. We sample 1 month of transactional data for each company (October 2017), and remove transactions that do not include a transaction amount. Since some of the transactions have different currencies than USD, all the transaction amounts are converted to USD equivalent. Next, features are extracted as described in Section 4.1 for all datasets. Categorical features are one-hot encoded to ensure compatibility across different classifiers. Missing values are imputed with mean-values for the numeric, with 'Category-other' for the categorical variables. We estimate each merchant's manual review capacity according to [7]. Table 2 presents the datasets' descriptive statistics.

Table 2. Descriptive Statistics

	OTA	PGS	DGS
Transactions	22,203	36,783	39,784
Fraudulent Transactions	349 (1.57%)	253 (0.69%)	1,536 (3.86%)
Transaction Amount Mean (μ)	$622.25	$177.22	$75.61
$\mu_{fraudulent}/\mu_{legimate}$	1.06	0.84	0.87
Manual Review Capacity	30%	20%	10%

We use the first 80% of the transactions as the training dataset, and the rest as the test dataset. To calculate the decision incentives, we set profit rate (pr) to 5%, lifetime value multiplier (ltv) to 3, fraud loss multiplier (flm) to 2.4, and review cost to $3 based on estimates from the merchants. For fraud classification models, we experiment with logistic regression (LR), gradient boosting machine (GBM), multilayer perceptron (MLP), and random forests (RF).

5.3 PONRM vs. Cost-Sensitive and Cost-Insensitive Baselines

In this experiment set, we investigate PONRM's performance in different settings in comparison with baseline cost sensitive and cost insensitive approaches.

Experimental Setup: Among all fraud classification models multilayer perceptron (MLP) resembles a similar structure to PONRM, hence, we report its performance characteristics alongside PONRM.

Baselines: We introduce the following baseline architectures:

- **MLP** is the multilayer perceptron classifier. We train a cost insensitive MLP classifier to detect legitimate and fraud detections. Transactions classified as legitimate are given *accept*, and those as fraudulent are given *reject* decisions.
- **CostMLP** is a cost sensitive binary classification model. It uses MLP as its learning component. Incentives of rejecting and accepting are given alongside with binary transaction labels. As in MLP, transactions classified as legitimate are given *accept*, and fraudulent are given *reject* decisions.
- **CostMLPwithR** is a cost sensitive trinary classification model. It uses MLP as its learning component. Incentives are given alongside trinary ground-truth decisions. Practically, it is the same as feeding transaction features to PONRM directly and bypassing the fraud classification model.
- **MLP+PONRM** is our proposed framework. It uses MLP as its fraud classification model component and PONRM as the risk manager.

We use profit gain (PG) and F-Measure to evaluate performances of above listed models. A grid search with $l = [0,1,2,3]$ and $\alpha = [0,0.0001]$ is performed for each MLP based model. First layer's layer size (L) is set to 300 in PONRM and other MLP based models. Each consecutive layer's size is calculated by square-rooting the previous layer's size.

Results: MLP+PONRM framework shows superior performance in terms of both performance metrics. Models with review decision options (CostMLPwithR, MLP+PONRM) also achieve superior results than models without review decision options (MLP, CostMLP). Cost sensitive approaches (CostMLP, CostMLPwithR) perform better than their cost insensitive counterpart (MLP) for maximizing the profit gain

and increasing F-Measure. One exception is the F-Measure performance in PGS dataset where having the smallest average fraudulent transaction amount leads to lower gains in decision incentives biased for rejecting fraudulent transactions. Thus, CostMLP performs worse than MLP.

Our proposed framework MLP+PONRM consistently overperforms CostMLPwithR. Even in CostMLPwithR's best performing case, MLP+PONRM achieves 20% greater profit gain and 24% better F-Measure overall.

Table 3. Comparison between PONRM and cost sensitive and insensitive baselines

	OTA		**PGS**		**DGS**	
	PG	F-Meas	PG	F-Meas	PG	F-Meas
MLP	0.1207	0.2769	0.0170	0.3115	0.1727	0.4143
CostMLP	0.0325	0.2874	0.0673	0.3048	0.2100	0.4222
CostMLPwithR	0.5954	0.7599	0.5280	0.7510	0.4541	0.5021
MLP+PONRM	**0.8113**	**0.8690**	**0.6514**	**0.8523**	**0.5876**	**0.6661**

5.4 PONRM vs. Risk Managers Under Different Review Capacities

In our third experiment set, we aim to show the efficacy of PONRM in comparison with other baseline risk managers in maximizing profit gain. We also explore the performance under different review capacities to ensure robust execution of our framework under various financial settings.

Baselines: Coupled with RF fraud classification model, we introduce 2 baseline fraud management strategies to compare with PONRM as follows:

- **Naive Risk Manager (NRM):** This model assigns accept/reject decisions based on a fraud classification model. If fraud classification model classifies the transaction as legitimate, it accepts, and if as fraudulent, it rejects. Next, it selects transactions randomly based on the review capacity and converts their decisions to review.
- **Price Prioritized Risk Manager (PPRM):** Similar to NRM, this risk manager uses a fraud classification model to produce initial decisions as *accept* or *reject*. Next, it assigns the transactions having the highest transaction amounts to review considering the capacity under experimentation. To achieve this, it first finds a transaction amount threshold based on the observed historical data, then sends the transactions exceeding this threshold until the specified review capacity is filled.

Experimental Setup: To be able to compare the performance of different risk managers, we fix the fraud classification model. Due to the space constraints, we only report the experiments with RF and others can be found in the supplementary material.[1] RF is chosen due to its superior performance. We explore different parameters of RF as number of trees being $n = [10,50,100,200]$.

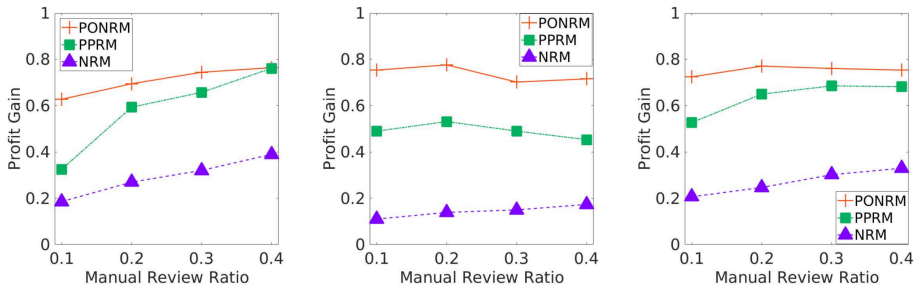
We run experiments with review ratios of 10%, 20%, 30%, and 40% and report their profit gain accordingly. Since there is no standard setting to enforce PONRM to produce any of the review ratios of 10%, 20%, 30% or 40%, we experiment with different values of the parameter *review class weight* (r) between 0.4 and 1.1 with 0.05 increments. According to the review ratio each PONRM experiment produces, we chunk

[1] http://www.public.asu.edu/~myildir3/cost_supp.pdf

them into bins of 10%, 20%, 30% or 40% review rates. We pick the best average performance of PONRM in the bins as the representative performance of the corresponding bin. Setting the review ratios for NRM and PPRM is straightforward.

Results: Figure 2 shows PONRM's performance in terms of Profit Gain when manual review capacity of the user is tweaked between 0.1 and 0.4. At first sight, it is clear that PONRM performs superior to the two baseline risk managers. Some other key findings are given as follows:

- Profit gain improves when manual review capacity is increased in OTA Dataset. For most of its transactions, review cost is negligible compared to the expected loss or profit, thus, when given maximum capacity, sending as much transactions as possible to review makes sense.
- Sending most transactions to manual review may not be a sound strategy for PGS and DGS datasets due to lower transaction amounts. Each merchant must identify the optimal manual review ratio and implement its model accordingly. This would also let the merchant save time and resources by automating the process more.
- When manual review capacity is 10%, PONRM performs up to **3 times** better than PPRM and **4 times** better than NRM. However, PPRM slowly catches up when the manual review ratio is unrealistically high.
- PPRM's constantly superior performance compared to NRM asserts that consideration of the transaction amount is crucial for risk management.



(a) OTA (b) PGS (c) DGS

Fig. 2. Performance of Risk Managers under Different Review Capacities

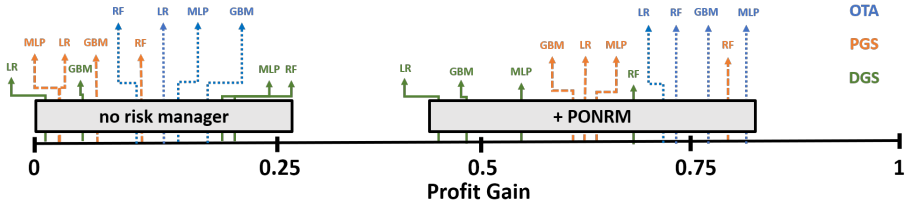
5.5 Which Classifier to Use as the Fraud Classification Model

Posterior probability distribution based on the selected classifier may greatly affect the performance of PONRM. Thus, we experiment with four previously mentioned supervised learners to demonstrate their effects in the framework. Experimental setup and parameter settings are explored as in Section 5.4 and results with best parameter combinations are reported here for the sake of brevity. More detailed parameter analysis and guidance can be found in the supplementary material.

Results: Figure 3 demonstrates the performance of fraud classification models with and without PONRM. Some major findings are given as follows:

- RF based fraud classification model with no risk manager often produces better results than the others with no risk manager. Especially its effectiveness in terms of profit gain contributes significantly to the RF+PONRM's performance, hence RF+PONRM generally gives the best performance.
- MLP+PONRM performs well on all datasets. Specifically, on OTA, it is marginally the best model where MLP uses only one hidden layer. There is a negative correlation between MLP+PONRM performance and number of layers in the MLP fraud classification model as it does not represent uncertainty accurately when complex.
- GBM+PONRM does not perform well as GBM is known to distort its posterior probabilities [19]. Since PONRM relies on the representation power of posterior probabilities, GBM is not an appropriate choice for our purposes.
- As a heuristic, profit gain of the fraud classification model can be used for model selection due to its positive correlation with PONRM's profit gain.

Fig. 3. Profit gain of fraud classification models with and without risk managers



6 Conclusion and Future Work

In this study, we propose a cost-sensitive decision making framework and demonstrate its effectiveness in fraud management. We reveal how human expertise can be combined with machine learning to make decisions under risk and cost considerations. Future work includes developing a novel metric to characterize the relationship between fraud classification models and PONRM performances. Also, we plan to investigate the generalizability of our framework to other domains such as loan evaluation and healthcare decision support.

Acknowledgements We thank Amador Testa and Ozgun Baris Bekki from Emailage Corp. for providing the datasets and their valuable industry insights.

References

1. Abadi, Martín, and others. 2015. "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems." https://www.tensorflow.org/.
2. Abe, Naoki, Bianca Zadrozny, and John Langford. 2004. "An Iterative Method for Multi-Class Cost-Sensitive Learning." In *Proceedings of the Tenth Acm Sigkdd International Conference on Knowledge Discovery and Data Mining*, 3–11. ACM.
3. Bolton, Richard J, and David J Hand. 2002. "Statistical Fraud Detection: A Review." *Statistical Science*. JSTOR, 235–49.
4. "Card Fraud Losses Reaches $21.84 Billion." 2016. Whitepaper. The Nilson Report.
5. Carneiro, Nuno, Gonçalo Figueira, and Miguel Costa. 2017. "A Data Mining Based System for Credit-Card Fraud Detection in E-Tail." *Decision Support Systems* 95. Elsevier:91–101.
6. CyberSource. 2016. "2016 North America Online Fraud Benchmark Report." Report. CyberSource Corporation.

7. CyberSource. 2017. “2017 North America Online Fraud Benchmark Report.” Report. CyberSource Corporation.
8. Domingos, Pedro. 1999. “Metacost: A General Method for Making Classifiers Cost-Sensitive.” In *Proceedings of the Fifth Acm Sigkdd International Conference on Knowledge Discovery and Data Mining*, 155–64. ACM.
9. Elkan, Charles. 2001. “The Foundations of Cost-Sensitive Learning.” In *International Joint Conference on Artificial Intelligence*, 17:973–78. Lawrence Erlbaum Associates Ltd.
10. Ghosh, Sushmito, and Douglas L Reilly. 1994. “Credit Card Fraud Detection with a Neural-Network.” In *System Sciences, 1994. Proceedings of the Twenty-Seventh Hawaii International Conference on*, 3:621–30. IEEE.
11. Halvaiee, Neda Soltani, and Mohammad Kazem Akbari. 2014. “A Novel Model for Credit Card Fraud Detection Using Artificial Immune Systems.” *Applied Soft Computing* 24. Elsevier:40–49.
12. Hooi, Bryan, and others. 2016. “Fraudar: Bounding Graph Fraud in the Face of Camouflage.” In *Proceedings of the 22nd Acm Sigkdd International Conference on Knowledge Discovery and Data Mining*, 895–904. ACM.
13. Juniper. 2017. “Online Payment Fraud: Emerging Threats, Key Vertical Strategies & Market Forecasts 2017-2022.” Whitepaper. Juniper Research.
14. KS&R. 2016. “2016 LexisNexis True Cost of Fraud Study.” Report. LexisNexis Risk Solutions.
15. Ling, Charles X, and Victor S Sheng. 2011. “Cost-Sensitive Learning.” In *Encyclopedia of Machine Learning*, 231–35. Springer.
16. Maes, Sam, Karl Tuyls, Bram Vanschoenwinkel, and Bernard Manderick. 2002. “Credit Card Fraud Detection Using Bayesian and Neural Networks.” In *Proceedings of the 1st International Naiso Congress on Neuro Fuzzy Technologies*, 261–70.
17. Montague, David A. 2010. *Essentials of Online Payment Security and Fraud Prevention*. Vol. 54. John Wiley & Sons.
18. Ngai, EWT, and others. 2011. “The Application of Data Mining Techniques in Financial Fraud Detection: A Classification Framework and an Academic Review of Literature.” *Decision Support Systems* 50 (3). Elsevier:559–69.
19. Niculescu-Mizil, Alexandru, and Rich Caruana. 2005. “Obtaining Calibrated Probabilities from Boosting.” In *UAI*, 413.
20. Phua, Clifton, Vincent Lee, Kate Smith, and Ross Gayler. 2010. “A Comprehensive Survey of Data Mining-Based Fraud Detection Research.” *arXiv Preprint arXiv:1009.6119*.
21. Van Vlasselaer, Véronique, and others. 2015. “APATE: A Novel Approach for Automated Credit Card Transaction Fraud Detection Using Network-Based Extensions.” *Decision Support Systems* 75. Elsevier:38–48.
22. Zadrozny, Bianca, John Langford, and Naoki Abe. 2003. “Cost-Sensitive Learning by Cost-Proportionate Example Weighting.” In *Data Mining, 2003. ICDM 2003. Third Ieee International Conference on*, 435–42. IEEE.
23. Zafarani, Reza, and Huan Liu. 2015. “10 Bits of Surprise: Detecting Malicious Users with Minimum Information.” In *Proceedings of the 24th Acm International on Conference on Information and Knowledge Management*, 423–31. ACM.
24. Zhang, Si, and others. 2017. “HiDDen: Hierarchical Dense Subgraph Detection with Application to Financial Fraud Detection.” In *Proceedings of the 2017 Siam International Conference on Data Mining*, 570–78. SIAM.
25. Zhou, Dawei, and others. 2017. “A Local Algorithm for Structure-Preserving Graph Cut.” In *Proceedings of the 23rd Acm Sigkdd International Conference on Knowledge Discovery and Data Mining*, 655–64. ACM.