Cost-Sensitive Selective Classification and its Applications to Online Fraud Management

by

Mehmet Yigit Yildirim

A Dissertation Presented in Partial Fulfillment
of the Requirement for the Degree
Doctor of Philosophy

Approved March 2019 by the
Graduate Supervisory Committee:

Hasan Davulcu, Chair
Dijiang Huang
Ihan Hsiao
Bertan Bakkaloglu

ARIZONA STATE UNIVERSITY

May 2019

ABSTRACT

Fraud is defined as the utilization of deception for illegal gain by hiding the true nature of the activity. While organizations lose around $3.7 trillion in revenue due to financial crimes and fraud worldwide, they can affect all levels of society significantly. In this dissertation, I focus on credit card fraud in online transactions. Every online transaction comes with a fraud 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, the highly effective manual review process is overlooked. First, I propose Profit Optimizing Neural Risk Manager (PONRM), a selective classifier that (a) constitutes optimal collaboration between machine learning models and human expertise under industrial constraints, (b) is cost and profit sensitive. I suggest directions on how to characterize fraudulent behavior and assess the risk of a transaction. I show that my framework outperforms cost-sensitive and cost-insensitive baselines on three real-world merchant datasets. While PONRM is able to work with many supervised learners and obtain convincing results, utilizing probability outputs directly from the trained model itself can pose problems, especially in deep learning as softmax output is not a true uncertainty measure. This phenomenon, and the wide and rapid adoption of deep learning by practitioners brought unintended consequences in many situations such as in the infamous case of Google Photos' racist image recognition algorithm; thus, necessitated the utilization of the quantified uncertainty for each prediction. There have been recent efforts towards quantifying uncertainty in conventional deep learning methods (e.g., dropout as Bayesian approximation); however, their optimal use in decision making is often overlooked and understudied. Thus, I present a mixed-integer pro-

gramming framework for selective classification called MIPSC, that investigates and combines model uncertainty and predictive mean to identify optimal classification and rejection regions. I also extend this framework to cost-sensitive settings (MIPCSC) and focus on the critical real-world problem, online fraud management and show that my approach outperforms industry standard methods significantly for online fraud management in real-world settings.

ACKNOWLEDGMENTS

I am eternally grateful to my family: my mother Rana, my father Faruk, and my brother Toygun. You always believed in me even when I doubted myself. I owe it all to you. I am who I am today thanks to you.

I would like to express my gratitude to my PhD advisor, Hasan Davulcu especially for his guidance, open-mindedness, and constant support among many other things. Also, I thank my valuable dissertation committee members Ihan Hsiao, Dijiang Huang, and Bertan Bakkaloglu for their contribution and insightful feedback on my research and dissertation.

I have been very lucky to have a very talented and dedicated partner in crime throughout this journey. He has been a dear friend in good days and in not so good days, a meticulous co-author bearing with me until literally the last minute of paper submission deadlines, and an avid hiker forcing me to experience the "beautiful" desert tirelessly. Thank you, Mert Ozer; this dissertation would not be possible without your collaboration.

I would like to thank my friends and colleagues for accepting nothing less than excellence from me and my research. Your support and friendship has pushed me forward whenever I struggled during this long process. Especially, thank you Burhan Senturk and Ayse Gundogdu Senturk for your comradery and your backing no matter where you are.

With a special mention to Ozgun Baris Bekki and Amador Testa, my sincere thanks go to the Emailage Team. It has been fantastic to have the opportunity to work with you all and make this research a real-world application rather than a hypothetical study. I sincerely appreciated your support and encouragement during my studies and benefited greatly from your extensive fraud expertise. I am looking

forward to taking online fraud management to the next level with you. I also thank the Early Warning team and in particular Scott Alcorn for their contribution and collaboration.

Last but not least, I would like to thank my dearest girlfriend Sultan Kilinc. None of this would be possible without your love, patience, support, and kindness. Thank you for being who you are and thank you for being with me.

TABLE OF CONTENTS

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

Chapter 1

RESEARCH OVERVIEW

1.1 Introduction

Following the huge success of the recent advances in machine learning and deep learning, developing strategies to make use of these models optimally becomes imperative. The ability to abstain from making an automated decision when the model is uncertain about an individual inference is essential to design these strategies. Concisely, this dissertation presents novel methods operating in the intersection of several areas such as selective classification, uncertainty representation, cost-sensitive learning, and operations research to make optimal decisions under uncertainty in real-world applications. Rest of this dissertation is organized as follows. In Chapter 1, I introduce the related work around uncertainty representation, selective classification, mixed-integer programming, and cost-sensitive learning. Following this background, I propose a cost-sensitive fraud management framework compatible with any supervised learning algorithm in Chapter 2. Then, Chapter 3 focuses on a generalizable selective classification framework, its cost-sensitive extension, and its applications to fraud management. Finally, in Chapter 4, I conclude the dissertation with key findings and future research direction.

1.2 Related Work

1.2.1 Cost-Sensitive Learning

Cost-sensitive learning is a largely studied data mining field in which models consider different types of costs including asymmetric misclassification costs when performing learning and prediction tasks. Cost-sensitive learning can treat the loss of a false positive differently than a false negative whereas regular (cost-sensitive) methods cannot make a distinction directly. Costs are usually represented as by positive values whereas benefits are denoted as negative in the cost matrix [Elkan (2001)]. Besides the misclassification costs, test costs such as feature retrieval and label acquisition costs can be of high importance in the modeling of a problem [Turney (1995)]. Although many categorizations are possible, here, we first categorize two main veins of cost-sensitive learning as "misclassification cost-sensitive learning" and "test cost-sensitive learning" for simplicity; then we investigate the literature hierarchy deeper within this categorization.

Misclassification cost-sensitive learning aims to handle different costs arising from the application domain or class imbalance. For example, in the problem of medical diagnosis not identifying a serious illness does not incur the same as falsely detecting a false sickness. Similarly, in the fraud detection domain, transaction amount brings a dynamic cost to each decision together with the costs of customer retention and fraud management. There are two main approaches in the cost-sensitive literature to handle different misclassification costs:

Direct approaches directly incorporate these costs in the loss function in the framework. A pioneering work in these type of methods is Turney (1995). Misclassification costs are utilized in the fitness of genetic algorithms by the ICET. Differently, Ling *et al.* (2004) incorporate misclassification costs in the cost sensitive decision in de-

cision tree framework. In another study, Drummond and Holte (2000) examine the cost-sensitivity in relation to attribute selection criteria of decision tree learning and argue that impurity models cost sensitivity the best. Work of Fan *et al.* (1999) is the first study exploring cost-sensitive learning using boosting. Authors achieve this by making cost-sensitive updates to the AdaBoost's weak learners at each iteration. Sun *et al.* (2007) develop another cost-sensitive boosting algorithm based on AdaBoost and demonstrate the effectiveness on imbalanced datasets. Authors claim that their method is more sensitive to cost aspects compared to Fan *et al.* (1999). Unlike these studies, Masnadi-Shirazi and Vasconcelos (2011) propose a cost-sensitive boosting framework based on boosting's statistical interpretation. Authors modify several boosting algorithms such as RealBoost, AdaBoost, and LogitBoost and show superiority in terms of cost-minimization. The same authors later introduce a cost-sensitive SVM framework in Masnadi-Shirazi *et al.* (2012).

Another approach is converting cost-insensitive learners into cost-sensitive ones by performing pre-processing or post-processing. We refer to these methods as meta cost-sensitive learners as in Ling and Sheng (2008). The well-known study by Domingos (1999) introduces the method, MetaCost. As the name suggests, this method is meta-cost sensitive learner that can be applied to any type of classifier. It uses a bagging variant and works as follows. MetaCost bootstraps training examples and learners multiple models. Then, using average voting it generates probabilities and weights them with the cost matrix, finally relabels the instances with the expected class labels to minimize cost. Sheng and Ling (2006) propose a similar meta-learning method which does not require accurate probability estimates instead uses accurate rankings. It relies on cross-validation to search for the best threshold of probability to find the optimal cut-off point. As these can be seen as post-processing based meta cost-sensitive learners, there also are pre-processors. Zadrozny *et al.* (2003)

use sampling to modify class distributions with respect to costs associated with the labels.

Cost-sensitive classification literature focuses on the problem of "test costs" more specifically than the cost-sensitive learning literature. These costs are explained as the cost incurred for acquiring extra information in terms of features. Medical diagnosis problem is again a good analogy for explaining the concept of test costs. A doctor may require additional tests to make a more confident diagnosis; however, it comes with time and monetary costs. So, it is the doctor's decision to make this investment or not based on the expected benefit of acquiring the result of the tests. Similarly, in the cost-sensitive classification domain, a framework could choose to "invest" in acquiring more information (features) if the expected benefit justifies the cost. In Turney (1995), the essential problem of minimizing the cost of classification when the tests are expensive is investigated. It argues that decision trees are the intuitive structure for this problem and introduced a hybrid genetic decision tree induction algorithm called ICET to generate low cost decision trees. Misclassification and test costs are incorporated in the fitness function. Zubek and Dietterich (2002) model cost-sensitive classification problem considering both misclassification and test costs as Markov Decision Process (MDP). Each observed feature brings the model to a new state and incurs a cost and changes the expected benefit of the model. Authors develop and combine statistical pruning and systematic search techniques to find a heuristic to the optimal solution in feasible time. Ling *et al.* (2004) also explore this problem using a decision trees by combining test and misclassifications costs using static cost structure. Authors interpret and scale misclassification costs in terms of monetary value as in the test costs. In Chai *et al.* (2004), authors develop a test-cost sensitive naive bayes learner unlike previous decision tree based approaches. In a similar vein, Zhang *et al.* (2005) do not focus on developing new techniques but in-

vestigates the effect of missing data in test-cost sensitive classification and concludes that missing data notion can be useful for decisioning. These can be perceived similar to obtaining a label in active inference literature but the costly information is not the label but the feature itself. Attenberg and Provost (2011) develop techniques for active cost-sensitive classification problem and investigates the effect of and the optimal choice for obtaining a ground-truth label at prediction time. Authors aim to estimate how many times they are likely to see the same instance in a streaming setting and how they can incorporate this estimation when making a label acquisition decision considering its cost and expected benefit. This is rather different from our work as we do not see an instance multiple times and our decision of label acquisition is based on uncertainty and monetary expectations.

Here, we move on to the extensions of the cost-sensitive learning frameworks. Reinforcement learning is utilized to make sequential cost-sensitive decisions to maximize long term profits in campaigns by Pednault *et al.* (2002). Margineantu (2005) extends the cost-sensitive principles introduced by Elkan (2001) to active learning settings using bagged probability estimation trees described in Provost and Domingos (2003). Attenberg and Provost (2011) propose the first online active cost-sensitive inference framework where the cost and benefit of encountering a labeled instance multiple times is considered when making a label acquisition decision. In a similar vein, Yang *et al.* (2009) use random forest based conformal prediction framework for medical diagnosis. This is the first paper extending cost-sensitive learning to conformal prediction settings which promise reliable confidence levels for each prediction. Kim (2010) proposes cost-sensitive condition random fields for structured learning. Authors demonstrate the framework's effectiveness using three applications such as human walking motion identification, oceanography biome characteristics prediction,

and object recognition in hierarchy. Semi-supervised cost-sensitive learning is investigated in Wang *et al.* (2012). They claim that cost-sensitive learning frameworks are effective in the existence of adequate labeled data but this is the first extension where a lot of unlabeled and a few labeled data are are available. Recently, cost-sensitive classification is modeled using robust minimax approach to allow direct minimization of the cost of mistakes as a convex optimization problem in Asif *et al.* (2015). In contrast, previous methods minimize a convex heuristic of the loss function. Experiments show their method's effectiveness to be better or comparable to the existing cost-sensitive boosting and SVM methods. One interesting study in face recognition, Li *et al.* (2016) show that even an seemingly unrelated domain can benefit from cost-sensitive learning. They minimize the misclassification cost instead of the misclassification error to incur and model varying costs of not recognizing a face accurately.

1.2.2 Fraud Detection using Machine Learning

Fraud detection has been an active area for data mining researchers since Ghosh and Reilly (1994); 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. While Fawcett and Provost (1997) and Chan *et al.* (1999) propose techniques for specific fraud detection applications, Fawcett and Provost (1999) formalize the class of activity monitoring problems which includes fraud detection. They may not be directly applicable or state-of-the-art today; however, they remain very relevant in terms of ideas they introduce and foundation they provide for future development.

In the more recent years, major studies focused on credit card fraud detection such

as Bolton and Hand (2002), Maes *et al.* (2002), and Van Vlasselaer *et al.* (2015). Due to the popularity of social media and user-generated content, fraudulent or misleading content detection became important. Recently, graph mining approaches have gained more traction and the product review fraud detection by Hooi *et al.* (2016) have received widespread attention. Theoretical contributions on graph mining focusing on fraud detection applications such as studied in the work from Zhang *et al.* (2017) and Zhou *et al.* (2017) are also made. For more comprehensive survey papers on fraud detection methods please refer to the studies, Phua *et al.* (2010) and Ngai *et al.* (2011).

Although fraud loss is an enormous problem for e-commerce merchants, there is only a pair of studies by Halvaiee and Akbari (2014) and Carneiro *et al.* (2017) investigating this problem from a merchant's perspective. However, these works aim to improve the accuracy of fraud detection alone, instead of 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 and the literature in covered in detail in 1.2.1. However, none of cost-sensitive learning frameworks in fraud detection domain approaches the problem from a selective classification perspective. Being the closest study, Carneiro *et al.* (2017) recognize the role of manual reviews in fraud prevention process; however, they do not provide a systematic analysis on how to integrate machine learning based detection with manual reviews under cost and capacity constraints. In Chapter 3, we develop

a cost-sensitive fraud management framework incorporating all relevant capacities, costs and evaluate its financial impact with multiple real-world merchant datasets.

1.2.3 Uncertainty Representation and Applications

Many practitioners and researchers make use of the probability outputs from the trained model (i.e., softmax output in deep learning) as an uncertainty measure; however, many classifiers output distorted probabilities according to Niculescu-Mizil and Caruana (2005) and this may lead to misleading actions. Moreover, even when corrected by proposed probability calibration methods such as Isotonic Regression by Kruskal (1964) or Platt Scaling by Platt *et al.* (1999), posterior probabilities as point estimates lack the detail and information to provide a correct interpretation of the model uncertainty. So, Bayesian approaches such as Polson *et al.* (2017); Rasmussen (2006) are the intuitive methods to quantify and represent the model uncertainty correctly. Due to the computational complexity of the Bayesian methods, Gal proposes using Monte Carlo sampling over dropout neural networks as an approximation to Bayesian inference in Gal and Ghahramani (2016). This approach's effectiveness is demonstrated in a medical-domain application in Leibig *et al.* (2017). Our work in Chapter 4 builds upon this framework by combining model uncertainty and predictive mean optimally for classification with reject option or selective classification.

1.2.4 Selective Classification

Selective classification or classification with reject option has been studied since the 1970's and it has started gaining traction again in the recent decade. It is defined as giving an option to the classifier to express uncertainty and to reject making a certain prediction. Chow (1970), being the first study in the field, introduces the concept and proposes a decision theoretic framework to find the Bayesian-optimal

reject threshold. Tortorella (2000), and Santos-Pereira and Pires (2005) propose cost-sensitive learning extensions to classification with reject option methods with arbitrary cost-functions. Herbei and Wegkamp (2006) develop excess risk bounds for the classification with a reject option for both cost-sensitive and cost-insensitive cases. On the other hand, El-Yaniv and Wiener (2010) find these cost models unsuitable as it is difficult to quantify the cost of rejection in many cases. Instead, authors focus on theoretical risk-coverage (RC) trade-off without considering explicit costs. Researchers have been adapting this idea to different classifiers and recently Geifman and El-Yaniv (2017) modified deep neural networks for selective classification. Our work differs fundamentally from Geifman and El-Yaniv (2017) by (1) not being built-in within the deep neural network itself; so it becomes compatible with any existing trained models and systems, and (2) utilizing dropout MC sampling for uncertainty estimation.

1.2.5 Mixed-Integer Programming

Mixed-Integer programming (MIP) is a powerful modeling tool that has been around for decades. MIP has been commonly utilized by the operations research community; however, practitioners and researchers from other domains hesitated to adopt it due to its computational and theoretical complexity Bixby (2010). During the last three decades, algorithmic advances in integer optimization combined with hardware improvements have enabled a 200 billion factor speedup in solving MIP problems according to Bertsimas *et al.* (2016). Now, mixed integer linear techniques are viewed as mature, fast, and robust; thus are applied to the problems with up to millions of variables Geißler *et al.* (2012). Machine learning community also started employing MIP techniques in several problems, such as for optimal feature selection as in Bertsimas *et al.* (2016) and for deriving interpretable machine learning algorithms

as shown by Goh and Rudin (2014). The key factors for our decision to use an MIP formulation are (1) its ability to naturally express the problem, the objective, and the constraints, (2) its capability to provide an exact optimal solution, and (3) its ease of extensibility to more specific settings.

Chapter 2

COST-SENSITIVE DECISION MAKING FOR ONLINE FRAUD MANAGEMENT

2.1 Introduction

According to Pickett and Pickett (2002), financial crime is the utilization of deception for illegal gain by hiding the true nature of the activity. They use the terms financial crime and fraud interchangeably since financial crime very often involves fraud. Financial crime can be committed through many fraud schemes such as check and credit card fraud, mortgage fraud, medical fraud, corporate fraud, bank account fraud, and health care fraud. These types of crimes involve relevant illegal activities such as identity theft, cyber attacks, money laundering, and social engineering according to Gottschalk (2010). While organizations lose around $3.7 trillion in revenue due to financial crimes and fraud worldwide (ACFE (2016)), they can affect all levels of society significantly (Interpol (2009)). Thus, fraud is a huge problem and its detection, prevention, and management is critical.

In 2016, card fraud alone cost businesses over $20 billion and continues to grow dramatically (Nilson (2016)). Around 60% of this loss was caused by online transactions, as e-commerce fraud rates 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 (Juniper (2017)).

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 (Montague (2010)). KS&R (2016) reports that for every dollar of loss, merchants end up losing $2.40 on average as fraud management costs. 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 (CyberSource (2016)).

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 (2017), manual review is an established mechanism for fraud prevention with adoption by 79% of North American businesses.

Despite all efforts to fight fraud, significant improvements can still be made by investigating and answering 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 and revisions 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 chapter, 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

2.2 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.

2.3 Methodology

Figure 2.1 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 2.1.

2.3.1 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

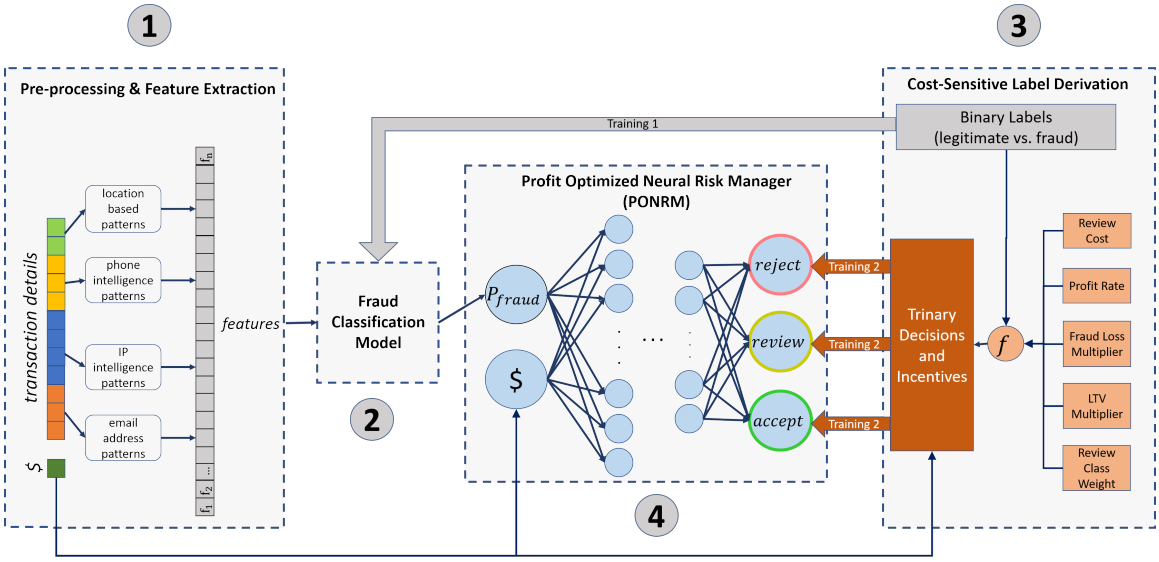


Figure 2.1: System Overview

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 Zafarani and Liu (2015), 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 was 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.

2.3.2 Fraud Classification Model & Risk Score Calculation

Risk score constitute 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 a transaction being fraud given its features. We propose using any supervised learner (θ) providing a robust posterior probability for fraud probability estimation such as:

$$\mathbf{f}_i = P(\mathbf{Y}_{i2} = 1 | \mathbf{X}_i; \theta) \tag{2.1}$$

where $\mathbf{f} = \{\mathbf{f}_i; \mathbf{f}_i \in [0, 1] \wedge i = 1 \dots N\}$. As given in Equation 2.1, $\mathbf{f}$ 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;

$$\mathbf{R} = \left[\mathbf{f}, \$\right] \tag{2.2}$$

2.3.3 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 compared to accept or reject decisions, proportionally.

Computing Cost-Sensitive Decision Incentives:

By following the fraud management strategy considerations from Section 2.2, 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

Table 2.1: 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	$flm * \$_i - rc$	$flm * \$_i$

presented in Table 2.1. 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.

2.3.4 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;

$$\mathbf{R}_i = [\mathbf{f}_i, \$_i] \tag{2.3}$$

$$\mathbf{H}^{(0)} = \sigma(\mathbf{W}^{(0)}\mathbf{R} + \mathbf{b}^{(0)}) \tag{2.4}$$

$$\mathbf{H}^{(i)} = \sigma(\mathbf{W}^{(i)}\mathbf{H}^{(i-1)} + \mathbf{b}^{(i)}) \qquad \text{for } i = 1, ..., l \tag{2.5}$$

$$\hat{\mathbf{Z}} = softmax(\mathbf{W}^{(l+1)}\mathbf{H}^{(l)} + \mathbf{b}^{(l+1)}) \tag{2.6}$$

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 $\hat{\mathbf{Z}} \in [0,1]^{N\times 3}$. To learn the parameters of the model, we use log loss multiplied by cost sensitive incentives and minimize the loss function by tuning $\mathbf{W}^{(i)}$,$\mathbf{b}^{(i)}$:

$$Loss = -\frac{1}{N}\left[\sum_{i=1}^{N}\sum_{c=1}^{3}\overbrace{\left[\mathbf{Z}_{ic}\log\hat{\mathbf{Z}}_{ic}\right]}^{log-loss}\overbrace{\mathbf{B}_{ic}}^{incentive}\right] + \overbrace{\sum_{i=1}^{l}\alpha_i||\mathbf{W}^{(i)}||_2^2}^{regularization} \tag{2.7}$$

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. $\hat{\mathbf{Z}}_{ic}$ 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 Abadi *et al.* (2015).

2.4 Experiments

In this section, 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.

2.4.1 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:

$$profit\ gain = \frac{\$_m - \$_{nofraudmanagement}}{\$_{oracle} - \$_{nofraudmanagement}} \tag{2.8}$$

where $\$_m$ is the profit of the model under experimentation. While calculating the profits, not-offset decision incentives in Table 2.1 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 con-

Table 2.2: 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%

figuration, best performing setting in terms of PG is reported as the representative performance of a model.

2.4.2 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 2.3.1 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 CyberSource (2017). Table 2.2 presents the datasets' descriptive statistics.

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 ex-

periment with logistic regression(LR), gradient boosting machine (GBM), multilayer perceptron (MLP), and random forests (RF).

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

In this experiment set, we investigate PONRM's performance in different setting 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 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 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 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 achieves superior results than models without review decision (MLP, CostMLP). Cost sensitive approaches (CostMLP, CostMLPwithR) performs 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.

2.4.4 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

Table 2.3: Comparison between PONRM and Cost-Sensitive and Cost-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	0.7110	0.4541	0.5021
MLP+PONRM	**0.8113**	**0.8690**	**0.6514**	**0.8523**	**0.5876**	**0.6661**

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 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 in each experiment. We explore different parameters of RF, GBM, and MLP and report the best results.

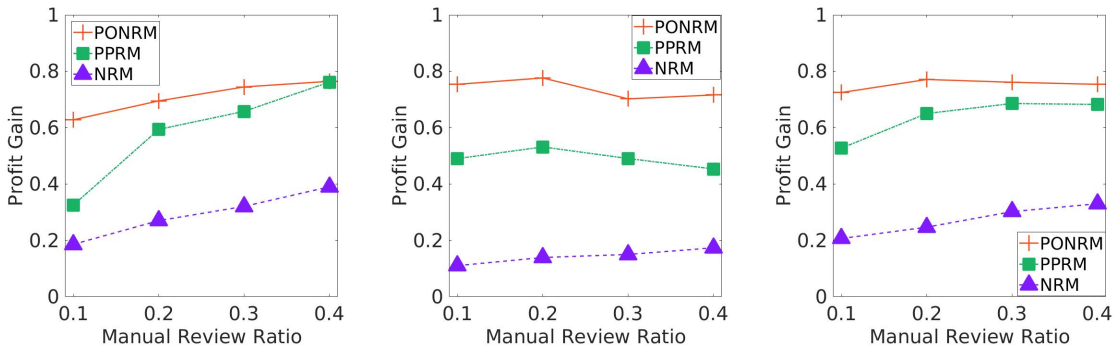
We run experiments with review ratios of 10%, 20%, 30%, and 40% and report their profit gain accordingly. Since there is no standard setting to enforce PONRM to produce any of the review ratios of 10%, 20%, 30% or 40%, we experiment with different values of the parameter *review class weight* (r) between 0.4 and 1.1 with 0.05 increments. According to the review ratio each PONRM experiment produces, we chunk 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 2.2, Figure 2.3, and Figure 2.4 show 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 almost always performs significantly superior to the baseline methods regardless of the Fraud Classification Model used in the framework. Some other key findings are given below:

- 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 provided sending as much transactions as possible to review makes sense.

- According to Figure 2.2(b), Figure 2.3(b), Figure 2.4(b), Figure 2.2(c), Figure 2.3(c), and Figure 2.4(c), sending most transactions for manual revision may



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

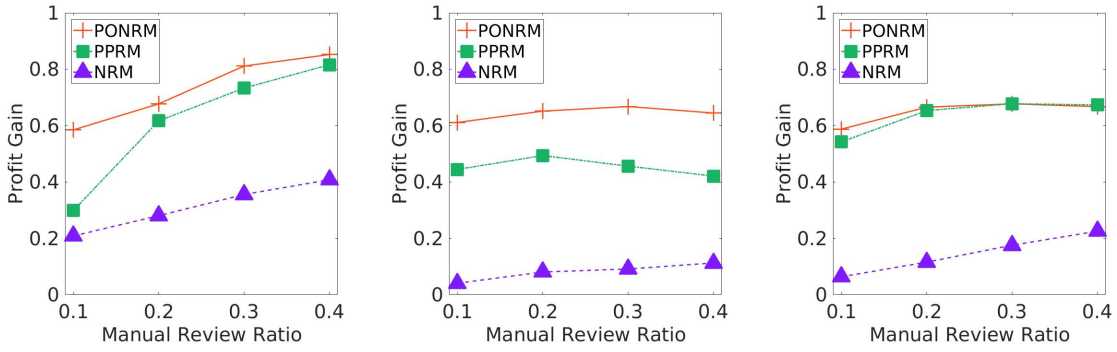
Figure 2.2: Performance of Risk Managers under Different Review Capacities Using Random Forest as the Fraud Classification Model

not be a sound strategy for PHY and DGS datasets again due to the transaction amount distribution. So, end-users could identify the optimal manual review ratio and implement their model accordingly. This would also let them save time and human resources as they would automating the process more.

- We observe the biggest performance differences when manual review ratio is 0.1 which is the most common capacity for larger merchants. PONRM performs between up to 3 times better than PPRM and 4 times better than NRM in the best case, however, PPRM slowly catches up when the manual review ratio is unrealistically high.

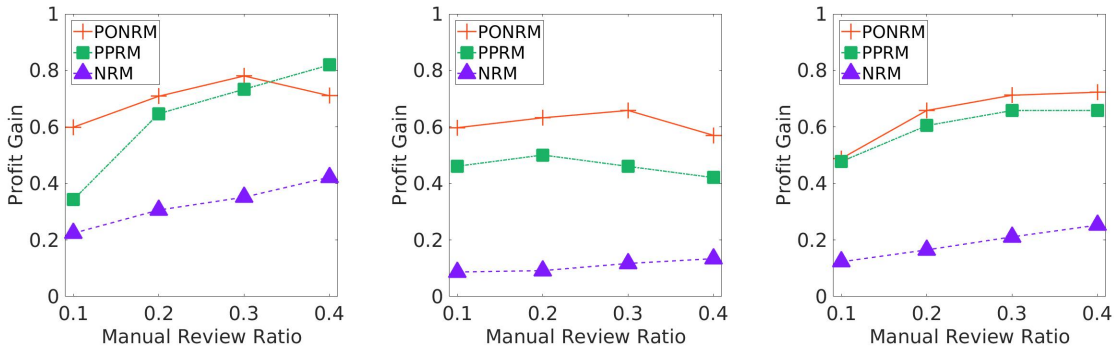
- PPRM's constantly superior performance compared to NRM asserts that consideration of the transaction amount is crucial for risk management.

- Random Forest performs the best overall among all fraud classification models when used with PONRM. We recommend using Random Forest as the Fraud Classification Model if there is no capacity to experiment with several options.



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

Figure 2.3: Performance of Risk Managers under Different Review Capacities using Multilayer Perceptron as the Fraud Classification Model



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

Figure 2.4: Performance of Risk Managers under Different Review Capacities Using Gradient Boosting as the Fraud Classification Model

2.4.5 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 2.4.4 and results with best parameter combinations are reported here for the sake of brevity.

2.4.6 Experimental Setup and Baselines

Similar to the previous experiment, Profit Gain is our primary metric in this set of experiments. Precision, accuracy, and F-measure are also provided to enable additional analysis and comparisons. These metrics are constructed using the "Fraud" outcome as the target value due to its appropriateness to the fraud detection context. Accuracy, despite being a gold-standard measure, is not reported because of its uninformativeness with imbalanced datasets and cost-sensitive nature of the problem.

First, we compare the performance of the Fraud Classification Models without integrating manual review process to observe their characteristics on all datasets. Then, we compare different PONRM models to base fraud classifiers to reveal the necessity of utilizing manual review process optimally to boost performance. Finally, we investigate how the specified Fraud Classification Model and selected Cost-Sensitive Decision Maker interacts.

As mentioned in Section 2.4.4, manual review capacity of vendors differ by revenue. Here, we employ the expected manual review rates accordingly and assess the effectiveness of our framework with different fraud classification models. So, OTA's manual review rate is assumed to be 0.3, while PGS's being 0.2 and DGS's 0.1 as calculated by their reported revenue.

Results

Table 2.4, Table 2.5, and Table 2.6 demonstrate the performance of PONRM and Fraud Classification Model itself. We report precision, recall and F-measure metric scores alongside profit gain. Some major findings are given as follows:

- RF based fraud classification model with no risk manager often produce 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. In the detailed experiments, we recognize that increasing the number of trees in the RF improves the profit gain of RF+PONRM as it reduces the variance of the fraud classification model and smooths the posterior probability distribution. Hence, we recommend the utilization of RF with higher number of trees if there is not enough resources to experiment with various models.

- 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 since it does not represent uncertainty accurately when complex.

- GBM + PONRM does not perform well as Gradient Boosting is known to distort its posterior probabilities in any dataset. As PONRM depends greatly on the accuracy of posterior probabilities, Gradient Boosting is not an appropriate choice for our purposes. Probability calibration using a method such as Isotonic Regression can help remedy this problem and may be considered if a classifier with distorted probabilities are desired to be used Niculescu-Mizil and Caruana (2005).

- Logistic regression does not do an adequate job for our purposes as decision boundaries often present non-linear patterns.

- We observe no correlation between Fraud Classification Model's performance based on gold-standard measures and PONRM's profit gain. So, Fraud Classification Model selection based on F-measure, precision, or recall is not sensible

Table 2.4: Classifier Performance - Online Travel Agency (OTA)

	Profit Gain		**Precision**		**Recall**		**F-Measure**	
Risk Mgr	**None**	**PONRM**	**None**	**PONRM**	**None**	**PONRM**	**None**	**PONRM**
LR	0.118	0.743	0.669	0.9672	0.168	0.756	0.269	0.849
GBM	**0.164**	0.781	0.365	0.940	**0.304**	0.795	**0.332**	0.861
MLP	0.121	**0.811**	0.656	0.940	0.176	**0.808**	0.277	**0.869**
RF	0.097	0.744	**0.805**	**1.000**	0.107	0.767	0.188	0.868

Table 2.5: Classifier Performance - Physical Goods Store (PGS)

	Profit Gain		**Precision**		**Recall**		**F-Measure**	
Risk Mgr	**None**	**PONRM**	**None**	**PONRM**	**None**	**PONRM**	**None**	**PONRM**
LR	0.017	0.650	0.386	**1.000**	0.100	0.746	0.159	0.855
GBM	0.069	0.633	0.329	0.697	**0.319**	0.777	**0.324**	0.735
MLP	0.017	0.651	0.308	0.9439	0.315	0.777	0.312	0.852
RF	**0.091**	**0.776**	**0.943**	**1.000**	0.143	**0.854**	0.249	**0.921**

for our purposes. We recognize that there is a need for a novel metric to define the relationship between Fraud Classification Model performance and PONRM effectiveness. As a heuristic, profit gain of the Fraud Classification Model can be used since it is highly correlated with the profit gain of the PONRM.

- PONRM's performance is noted to be positively correlated with the number of layers in the PONRM component regardless of the utilized fraud classification model. Thus, even deeper PONRM models may yield further promising results.

Table 2.6: Classifier Performance - Digital Goods Store (DGS)

	Profit Gain		**Precision**		**Recall**		**F-Measure**	
Risk Mgr	**None**	**PONRM**	**None**	**PONRM**	**None**	**PONRM**	**None**	**PONRM**
LR	0.005	0.393	0.364	0.927	0.044	0.313	0.079	0.468
GBM	0.069	0.489	0.696	0.9083	0.122	0.505	0.207	0.649
MLP	0.173	0.588	0.475	0.805	**0.367**	0.568	**0.414**	0.666
RF	**0.174**	**0.724**	**0.993**	**1.000**	0.201	**0.694**	0.335	**0.820**

Chapter 3

LEVERAGING UNCERTAINTY IN DEEP LEARNING FOR SELECTIVE CLASSIFICATION

3.1 Introduction

Machine learning classifiers are far from outputting perfect results due to several reasons: data quality, feature informativeness, model selection, and hyper-parameter tuning are just some of the factors contributing to the variability of the outcomes. Although well-trained models offer high level of accuracy on the macro level, making confident inferences for individual instances is difficult, nevertheless necessary.

Bayesian literature offers a rich set of classification techniques (Polson *et al.* (2017); Rasmussen (2006)) for jointly quantifying uncertainty and prediction at inference level. A recent application of dropout neural networks as Bayesian approximation of deep Gaussian Process by Gal et al. open a new avenue of quantifying uncertainty in traditional deep learning settings where a simple dropout mechanism is applicable (Gal and Ghahramani (2016)).

The gained ability to effectively represent the uncertainty within existing deep learning architectures has been an important step for democratizing AI safety (Amodei *et al.* (2016)). Nevertheless, the following question still remains open: how can one make use of the model uncertainty to make optimal decisions? The approach we focus on in this study is called *selective classification* also known as *classification with reject option* where the classifier rejects making a decision when uncertain.

Selective classification is critical for many applications, and the concept of "rejection" can have different meanings in various contexts. In medical diagnosis, a doctor

might order diagnostic tests before making a decision. In fraud management, an expert human analyst would start a manual investigation. In self-driving cars, the human driver would be given control to operate the vehicle. In all cases, rejecting most of the instances would defeat the purpose and being inaccurate could result in fatal consequences. Hence, a practical framework for selective classification must be able to operate accurately under defined rejection capacity constraints.

A recent study in the medical domain by Leibig *et al.* (2017) has demonstrated the potential of the model uncertainty for selective classification. However, the authors' utilization of the measure is solely based on a simple ranking of it, which makes their work unsuitable for many online or streaming settings. To the best of our knowledge, how model uncertainty compares to or interacts with the more traditional ways of conducting selective classification such as using Bayes risk introduced by Chow (1970) has not been explored.

Hence, we propose a Mixed-Integer Programming (MIP) formulation for selective classification called MIPSC to address these requirements. MIPSC finds optimal classification and rejection regions by investigating the relationship between the model uncertainty and predictive mean with the desired rejection capacity without having to define arbitrary rejection costs. Furthermore, we develop cost-sensitive extensions to our MIP model and exhibit the framework's extensibility and usability in real-world problems such as fraud management, where defining domain-specific and example-dependent costs are necessary.

Main contributions of this chapter are:

1. Introducing the first mixed integer programming solution for selective classification,

2. Utilizing predictive mean and model uncertainty of dropout NNs for optimal

decision making,

3. Presenting an online fraud management case in a real-world setting.

3.2 Proposed Models

In this work, we propose a mixed integer programming model which finds optimal regions in deep neural network classifier output to reject making a classification. To take not only the output of the deep neural network classifier but also its uncertainty into consideration we choose to use dropout NNs (DNN) Gal (2016) throughout our modeling and experiments. Dropout NNs have been proven to approximate deep Gaussian processes which generate predictive mean(μ) and model uncertainty(σ) in the form of standard deviation. In the following sections, we explain how we make use of both outputs (predictive mean and model uncertainty) of dropout NNs for selective classification.

3.2.1 Mixed-Integer Programming based Selective Classification

Here, we define a mixed-integer programming model for selective classification to make optimal decisions of classifications and rejections under uncertainty in deep learning. Equivalent to other selective classification models, the aim is to "reject" making an automated classification for certain instances to increase the performance on non-rejected samples. Similar to many supervised algorithms, our MIP model has two main workflows: training and inference. In the training phase, given an already trained dropout neural network (DNN), we learn the optimal criteria to reject samples by minimizing the number of mistakes made after rejections. Besides, we design our model in a way that it does not reject the samples without increasing the accuracy in the non-rejected sample space. These properties give rise to our objective function

as follows:

$$\underset{\phi_D,\phi_R}{\text{minimize}} \sum_{i \in \phi_D} [f(x_i) \neq y_i] + \lambda \sum_{i \in \phi_R} 1$$

where $x_i \in \mathbb{R}^n$ is the set of features for an instance i, $y_i \in \{0, 1\}$ is the label for that instance, and $f : \mathbb{R}^n \rightarrow \{0, 1\}$ is the previously trained deep neural network, ϕ_R is the set of rejected instances, and ϕ_D is the set of non-rejected instances.

So, what does our model use to determine the rejection population, ϕ_R and the decision population, ϕ_D? As introduced by Gal and Ghahramani (2016), our model uses the concept "model uncertainty" and enhances it with predictive mean to express when the DNN is not confident with its prediction.

For every instance in the training set, we calculate the predictive mean (μ_i) and the model uncertainty (σ_i) and map the points (μ_i,σ_i) to a 2D space. One intuitively expects more homogeneous regions to be near lower values of the model uncertainty and extremes of the predictive mean. This intuition can also be observed in Figure 3.1. Hence, our formulation aims to exploit and optimize upon this structure and identifies the thresholds that define our model's classification and rejection regions. Before formally defining our model, we introduce the notation that we refer to throughout this section in Table 3.2.2. We characterize five decision areas of classification and rejection and graphically demonstrate these areas in Figure 3.1. A_1 defines the decision region for positive classification while A_4 represents the decision region for negative classification. A_2 and A_5 are rejection regions due to their high model uncertainty. Thresholds to determine these regions are not tied together for the purpose of handling imbalance or class specific patterns in the data. Finally, A_3 is another rejection region housing instances having predictive means close to 0.5. In this region, model uncertainty becomes trivial due to its context: it does not matter how "certain" the model is when making a decision similar to a coin toss.

Variable	Definition
y_i	Ground truth label of instance i
p_i	Positive classification indicator for instance i
n_i	Negative classification indicator for instance i
r_i	Rejection indicator for instance i
μ_i	Predictive mean for instance i
σ_i	Uncertainty for instance i
μ_L	Left boundary for rejection
μ_R	Right boundary for rejection
σ_L	Upper uncertainty boundary for positive decisions
σ_R	Upper uncertainty boundary for negative decisions
L_i	Left area indicator for instance i
R_i	Right area indicator for instance i
D_{L_i}	Down-left area indicator for instance i
D_{R_i}	Down-right area indicator for instance i
$rCap$	Rejection capacity

Table 3.1: Notation Table for MIPSC

Boundaries for these regions $(\sigma_L, \sigma_R, \mu_L, \mu_R)$ are determined by the following set of constraints operating in a supervised fashion through the objective. This is the essential process executed by solving our MIP formulation.

Here, we start describing our constraints formally. The following constraint regulates the samples which do not reside in the rejection region A_3 based on their predictive means but on the right hand side of A_3 such that $i \in A_4 \cup A_5$:

$$\mu_i > 0.5 + \mu_R \text{ iff } R_i = 1 \tag{3.1}$$

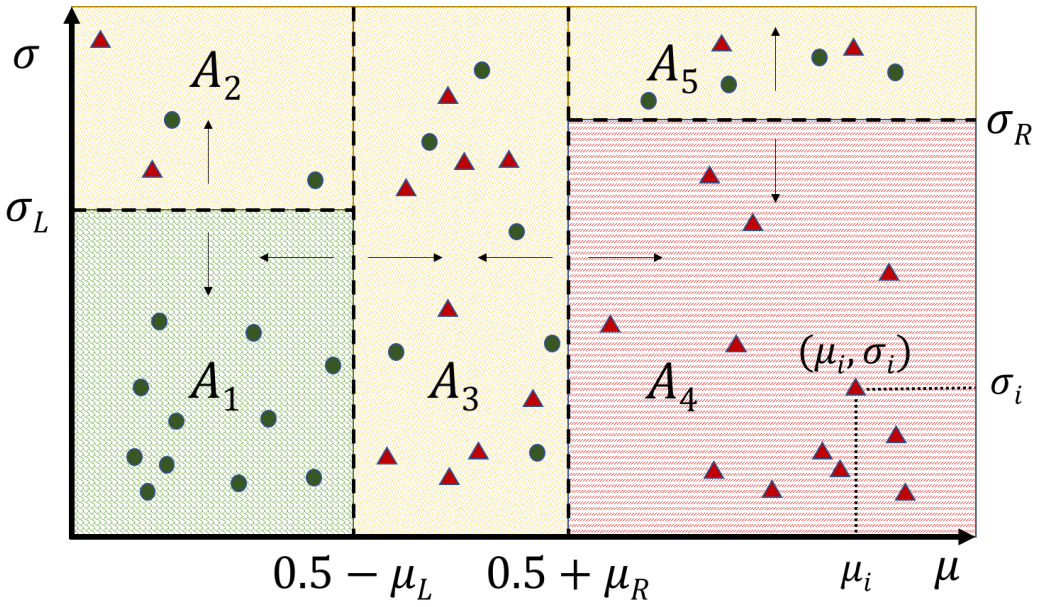


Figure 3.1: Graphical Illustration of the MIPSC Model

Now, we would like to distinguish the instances between A_4 and A_5 optimally such that our model would make a negative classification decision only when DNN is certain enough. The following constraints characterize the samples that conform to A_4 such that $i \in A_4$:

$$\sigma_i < \sigma_R \text{ iff } D_{R_i} = 1 \tag{3.2}$$

$$R_i + D_{R_i} > 1 \text{ iff } n_i = 1 \tag{3.3}$$

Similarly, the following constraint define the samples which do not reside in the rejection region A_3 based on their predictive means but on the left hand side of A_3 such that $i \in A_1 \cup A_2$:

$$\mu_i < 0.5 - \mu_L \text{ iff } L_i = 1 \tag{3.4}$$

Further, we would like to distinguish the instances between A_1 and A_2 optimally such that our model would make a positive classification decision only when DNN is certain enough. The following constraints characterize the samples that conform to A_1 such that $i \in A_1$:

$$\sigma_i < \sigma_L \text{ iff } D_{L_i} = 1 \tag{3.5}$$

$$L_i + D_{L_i} > 1 \text{ iff } p_i = 1 \tag{3.6}$$

As we have constrained our positive and negative classification decision regions, we reject the remaining instances covered by the constraint below:

$$p_i + n_i + r_i = 1 \tag{3.7}$$

where the reject decision is assigned when our model cannot a make positive or negative classification decision for instance i due to DNN uncertainty or predictive mean.

Finally we would like to enforce a certain number of rejections based on our application needs. This is given as:

$$\big(\sum_{i=1}^{m} r_i\big) \leq rCap \tag{3.8}$$

Combining our objective function and constraints together, then, setting M to be a very large positive constant and fixing ϵ to be a very small positive constant give rise to the formal definition of our model as follows:

$$\underset{\mu_L,\mu_R,\sigma_L,\sigma_R}{\text{minimize}} \sum_{i=1}^{m} (p_i y_i + n_i(1-y_i)) + \lambda \sum_{i=1}^{m} r_i \ \text{ s.t.} \tag{3.9}$$

$$\mu_R - \epsilon + MR_i \geq \mu_i - 0.5 \geq \mu_R - M(1-R_i), \forall i \tag{3.10}$$

$$M(1-L_i) - \mu_L \geq \mu_i - 0.5 \geq \epsilon - \mu_L - ML_i, \forall i \tag{3.11}$$

$$\sigma_L + M(1-D_{L_i}) \geq \sigma_i \geq \sigma_L + \epsilon - MD_{L_i}, \forall i \tag{3.12}$$

$$\sigma_R + M(1-D_{R_i}) \geq \sigma_i \geq \sigma_R + \epsilon - MD_{R_i}, \forall i \tag{3.13}$$

$$D_{L_i} + L_i \geq 2p_i \geq D_{L_i} + L_i - 1, \forall i \tag{3.14}$$

$$D_{R_i} + R_i \geq 2n_i \geq D_{R_i} + R_i - 1, \forall i \tag{3.15}$$

$$p_i + n_i + r_i = 1, \forall i \tag{3.16}$$

$$\Big(\sum_{i=1}^{m} r_i\Big) \leq rCap \tag{3.17}$$

$$\forall p_i, r_i, n_i, R_i, L_i, D_{L_i}, D_{R_i} \in \{0,1\} \tag{3.18}$$

$$\forall i \in \{1...m\}, and \ \ \mu_L, \mu_R, \sigma_L, \sigma_R, \lambda \in \mathbb{R} \tag{3.19}$$

In this formulation, constraint (3.10) is derived from (3.1), (3.11) is derived from (3.4), (3.12) is derived from (3.5), (3.13) is derived from (3.2), (3.14) is derived from (3.6), and (3.15) is derived from (3.3) following the Big-M method as shown in Griva *et al.* (2009).

Following the training, inference is rather straightforward. After acquiring the predictive mean and model uncertainty from DNN for the new sample, a user of our model can arithmetically decide the region the new sample belongs to and make the decision based on the optimal thresholds identified.

3.2.2 *Cost-Sensitive Selective Classification*

Many classification with reject option problems are cost-sensitive by nature. For instance, in medical diagnosis, consequences from a false negative decision can be fatal if the diagnosis in question is cancer but not as critical if it is the common cold. Within the same context, a doctor can order more tests with varying costs if uncertain depending on the severity of the illness under study. We follow Elkan's definition Elkan (2001) and extend our model to "example and class-dependent cost sensitive" settings where each instance belonging to each class has a different cost or benefit of making a correct or incorrect classification. Since the value add another dimension to our problem, we extend the previously introduced five decision regions to three dimensions and use simple thresholds for the value dimension for each region. A graphical interpretation of this extension can be viewed in Figure 3.2. Retaining our decision variables $(\sigma_L, \sigma_R, \mu_L, \mu_R)$, we introduce five more thresholds $(t_{DR}, t_{UR}, t_{DL}, t_{UL}, t_M,)$ based on the third dimension, value (cost/benefit). Finally, we assign the cost of rejection c to every reject decision, thus remove the rejection regularizer from the objective function.

Inheriting constraints (3.1), (3.2), (3.4), (3.5), and (3.8); we extend our constraints with the following statements:

The following constraint focuses on the region A_1 and finds the value threshold for that region. If the transaction corresponds to A_1 region and its value is less than the region's value threshold, then our model makes a positive decision.

$$t_i < t_{DL} \text{ iff } S_{DL_i} = 1 \tag{3.20}$$

$$L_i + D_{L_i} + S_{DL_i} > 2 \text{ iff } p_{i1} = 1 \tag{3.21}$$

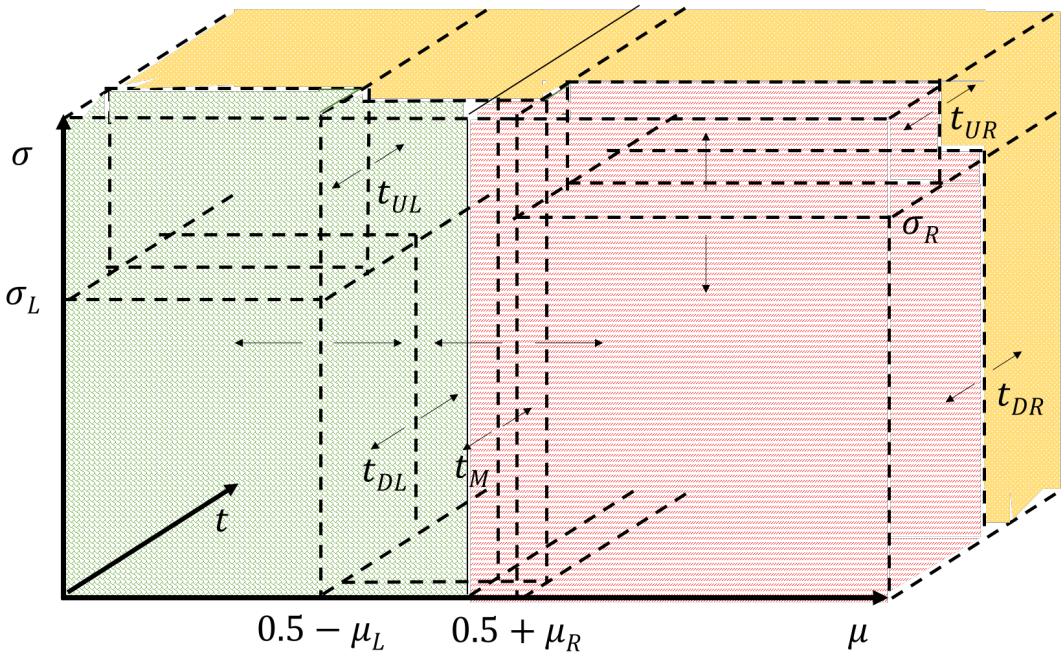


Figure 3.2: Graphical Illustration of the MIPCSC Model

Now, we would like to find our decision threshold for A_2. Similarly to the previous constraints, if the transaction corresponds to A_2 region and its value is less than the region's value threshold, then our model makes a positive decision.

$$t_i < t_{UL} \text{ iff } S_{UL_i} = 1 \tag{3.22}$$

$$L_i + (1 - D_{L_i}) + S_{UL_i} > 2 \text{ iff } p_{i2} = 1 \tag{3.23}$$

Similar to the positive decision regions, now, we focus on the negative decision regions: A_4 and A_5. The following constraint focuses on the region A_4 and finds the value threshold for that region. If the transaction corresponds to A_4 region and its value is less than the region's value threshold, then our model makes a negative decision.

Variable	Definition
t_i	Value for instance i
p_{ij}	Positive classification indicator for instance i and area j
n_{ij}	Negative classification indicator for instance i and area j
r_i	Rejection indicator for instance i
t_{DL}	Down-left area value boundary for rejection
t_{UL}	Upper-left area value boundary for rejection
t_M	Middle area value boundary for rejection
t_{DR}	Down-right area value boundary for rejection
t_{UR}	Upper-right area value boundary for rejection
S_{DL_i}	Surface-down-left area indicator for instance i
S_{DR_i}	Surface-down-right area indicator for instance i
S_{UL_i}	Surface-up-left area indicator for instance i
S_{UR_i}	Surface-up-right area indicator for instance i
S_{M_i}	Surface-down-middle area indicator for instance i

Table 3.2: Additional Notation Table for MIPCSC

$$t_i < t_{DR} \text{ iff } S_{DR_i} = 1 \tag{3.24}$$

$$R_i + D_{R_i} + S_{DR_i} > 2 \text{ iff } n_{i1} = 1 \tag{3.25}$$

Now, we would like to find our decision threshold for A_5. Similarly to the previous constraints, if the transaction corresponds to A_5 region and its value is less than the region's value threshold, then our model makes a negative decision.

$$t_i < t_{UR} \text{ iff } S_{UR_i} = 1 \tag{3.26}$$

$$R_i + (1 - D_{R_i}) + S_{UR_i} > 2 \text{ iff } n_{i2} = 1 \tag{3.27}$$

Finally, we move onto our middle region, A_3. Here, we would like our model to make a positive or negative decision using the predictive mean of 0.5 as the threshold and considering the value threshold we optimally determine by solving the problem, t_M.

$$\mu_i > 0.5 \text{ iff } Q_i = 1 \tag{3.28}$$

$$t_i < t_M \text{ iff } S_{M_i} = 1 \tag{3.29}$$

$$(2 - L_i - R_i) + S_{M_i} + (1 - Q_i) > 3 \text{ iff } p_{i3} = 1 \tag{3.30}$$

$$(2 - L_i - R_i) + S_{M_i} + Q_i > 3 \text{ iff } n_{i3} = 1 \tag{3.31}$$

As we have constrained our positive and negative classification decision regions, we reject the remaining instances covered by the constraint below:

$$\sum_{j=1}^{3} [p_{ij}] + \sum_{j=1}^{3} [n_{ij}] + r_i = 1, \forall i \tag{3.32}$$

where the reject decision is assigned when our model cannot a make positive or negative classification decision for instance i due to DNN uncertainty, predictive mean, or it does not make financial sense to spend money on a reject decision.

Following our definition and our constraints, we propose our cost-sensitive framework called Mixed-Integer Programming based Cost-Sensitive Selective Classification (MIPCSC) formally as follows:

$$\begin{aligned}\underset{\substack{\mu_L,\mu_R,\sigma_L,\sigma_R,\\ t_{DL},t_{UL},t_M,t_{DR},t_{UR}}}{\text{maximize}} \quad & \omega_{tp}(\sum_{i=1}^{n}\sum_{j=1}^{3} p_{ij}(1-y_i)t_i + \sum_{i=1}^{n} r_i(1-y_i)t_i) \\ & +\omega_{tn}(\sum_{i=1}^{m}\sum_{j=1}^{3} n_{ij}y_i t_i + \sum_{i=1}^{m} r_i y_i t_i) \\ & -\omega_{fn}(\sum_{i=1}^{m}\sum_{j=1}^{3} n_{ij}(1-y_i)t_i) \\ & -\omega_{fp}(\sum_{i=1}^{m}\sum_{j=1}^{3} p_{ij}y_i t_i) - c\sum_{i=1}^{m} r_i \end{aligned} \tag{3.33}$$

$$\mu_R - \epsilon + MR_i \geq \mu_i - 0.5 \geq \mu_R - M(1-R_i), \forall i \tag{3.34}$$

$$M(1-L_i) - \mu_L \geq \mu_i - 0.5 \geq \epsilon - \mu_L - ML_i, \forall i \tag{3.35}$$

$$\sigma_L + M(1-D_{L_i}) \geq \sigma_i \geq \sigma_L + \epsilon - MD_{L_i}, \forall i \tag{3.36}$$

$$\sigma_R + M(1-D_{R_i}) \geq \sigma_i \geq \sigma_R + \epsilon - MD_{R_i}, \forall i \tag{3.37}$$

$$0.5 + \epsilon + MQ_i \geq \mu_i \geq 0.5 + M(Q_i - 1), \forall i \tag{3.38}$$

$$t_{DL} + M(1-S_{DL_i}) \geq t_i \geq t_{DL} + \epsilon - S_{DL_i}, \forall i \tag{3.39}$$

$$t_{UL} + M(1-S_{UL_i}) \geq t_i \geq t_{UL} + \epsilon - S_{UL_i}, \forall i \tag{3.40}$$

$$t_M + M(1-S_{M_i}) \geq t_i \geq t_M + \epsilon - S_{M_i}, \forall i \tag{3.41}$$

$$t_{DR} + M(1-S_{DR_i}) \geq t_i \geq t_{DR} + \epsilon - S_{DR_i}, \forall i \tag{3.42}$$

$$t_{UR} + M(1-S_{UR_i}) \geq t_i \geq t_{UR} + \epsilon - S_{UR_i}, \forall i \tag{3.43}$$

$$D_{L_i} + L_i + S_{DL_i} \geq 3p_{i1}, \forall i \tag{3.44}$$

$$D_{L_i} + L_i + S_{DL_i} - 2 \leq 3p_{i1}, \forall i \tag{3.45}$$

$$(1 - D_{L_i}) + L_i + S_{DL_i} \geq 3p_{i2}, \forall i \tag{3.46}$$

$$(1 - D_{L_i}) + L_i + S_{DL_i} - 2 \leq 3p_{i2}, \forall i \tag{3.47}$$

$$D_{R_i} + R_i + S_{DR_i} \geq 3n_{i1}, \forall i \tag{3.48}$$

$$D_{R_i} + R_i + S_{DR_i} - 2 \leq 3n_{i1}, \forall i \tag{3.49}$$

$$(1 - D_{R_i}) + R_i + S_{DR_i} \geq 3n_{i2}, \forall i \tag{3.50}$$

$$(1 - D_{R_i}) + R_i + S_{DR_i} - 2 \leq 3n_{i2}, \forall i \tag{3.51}$$

$$(1 - L_i) + (1 - R_i) + S_{M_i} + (1 - Q_i) \geq 4p_{i3}, \forall i \tag{3.52}$$

$$(1 - L_i) + (1 - R_i) + S_{M_i} + (1 - Q_i) - 3 \leq 4p_{i3}, \forall i \tag{3.53}$$

$$(1 - L_i) + (1 - R_i) + S_{M_i} + Q_i \geq 4n_{i3}, \forall i \tag{3.54}$$

$$(1 - L_i) + (1 - R_i) + S_{M_i} + Q_i - 3 \leq 4n_{i3}, \forall i \tag{3.55}$$

$$\sum_{j=1}^{3} [p_{ij}] + \sum_{j=1}^{3} [n_{ij}] + r_i = 1, \forall i \tag{3.56}$$

$$\forall p_{ij}, r_i, n_{ij}, R_i, L_i, D_{L_i}, D_{R_i}, S_{DL_i}, S_{UL_i}, S_{M_i}, S_{DR_i}, S_{UR_i} \in \{0, 1\} \tag{3.57}$$

$$\forall i \in \{1...m\}, and \;\; \mu_L, \mu_R, \sigma_L, \sigma_R \in \mathbb{R} \tag{3.58}$$

In this formulation, constraint (3.40) is derived from (3.28), (3.41) is derived from (3.20), (3.42) is derived from (3.22), (3.43) is derived from (3.29), (3.44) is derived from (3.24), and (3.45) is derived from (3.26).

Next, (3.46) and (3.47) are derived from (3.21), (3.48) and (3.49) are derived from (3.23), (3.50) and (3.51) are derived from (3.25), and (3.52) and (3.53) are derived from (3.27).

Finally, (3.54) and (3.55) are derived from (3.30), and (3.54) and (3.55) are derived from (3.31) following the Big-M method as shown in Griva *et al.* (2009).

Inference using MIPCSC also follows similar steps to MIPSC. After predictive mean, model uncertainty, and the cost/benefit for the new sample are obtained, a user can arithmetically decide the region the new sample belongs to and make the decision based on the optimal thresholds identified.

3.3 Experiments

3.3.1 Experimental Setup

We develop two sets of experiments for classification with reject option and its cost-sensitive extension. For both tasks, we divide the dataset into four distinct sets; the first to train the dropout neural network(DNN), the second to find optimal dropout rate and regularization coefficient to quantify uncertainty, the third to train the proposed MIP models, and the fourth to test the performance of the proposed MIP models.

To quantify model uncertainty and predictive mean, we train a dropout neural network of 2 hidden layers with relu activations and dropout applied before each layer. For the implementation of DNN, we make use of the source codes of the original authors made publicly available at their website [1] . We apply a grid search among dropout rates of (0.05, 0.01, 0.02) and regularization coefficients (0.1, 0.25) to achieve optimal DNN configuration.

3.3.2 Evaluation Metrics

Conventional measures of performance introduced for supervised classification tasks do not represent the performance of a model with reject option under study, comprehensively Condessa *et al.* (2017). Here we present four recently introduced

[1] https://github.com/yaringal/DropoutUncertaintyExps

metrics for classification with reject optionCondessa *et al.* (2017) and cost sensitive learningYildirim *et al.* (2018). Ideally, a classifier with reject option should classify as many instances as possible correctly and reject to classify the ones that it would misclassify. A cost-sensitive classifier with reject option makes these decisions based on the profit or loss it would get from each instance. We use c for accurately classified and non-rejected samples, $\overline{c}$ for misclassified and non-rejected samples, r for misclassified and rejected samples, $\overline{r}$ for accurately classified and rejected samples.

Non-rejected Accuracy measures the performance of classification of the model on non-rejected samples. It is defined as $c/(c+\overline{c})$.

Classification Quality measures the performance of both classification and rejection of the model. It is defined as $(c+r)/(c+r+\overline{c}+\overline{r})$

Rejection Quality measures the relative performance of rejection to the overall performance of classification. It is defined as $(r/\overline{r})/\big((\overline{c}+r)/(c+\overline{r})\big)$

Profit Gain measures the level of gained profit from the model outcome relative to perfectly classifying every instance without any rejection and assigning every instance to the majority class. Let $\$_{model}$ be the profit gain of model under study, $\$_{oracle}$ be the profit gain of perfect model, and $\$_{majority}$ be the profit gain of majority class assigning model, we define profit gain as;

$$\big(\$_{model} - \$_{majority}\big)/\big(\$_{oracle} - \$_{majority}\big)$$

3.3.3 Experiments with UCI Datasets

In this section, we discuss how the performance of our framework is on several publicly available datasets. We experiment with 11 datasets from UCI classification repository and report our performance.

We setup experiments of binary classification with reject option on datasets coming from various application areas. We refer readers to Table 3.3 for simple statistics of datasets. They span applications of credit card applications(*australian*), medical diagno-

	Instances	Features	Majority Class
australian	689	14	67%
breast	699	19	65%
diabetic	1151	19	54%
heart	303	20	54%
sonar	208	60	53%
ionosphere	351	34	64%
german	208	60	70%
haberman	208	60	74%
seismic	208	60	93%
pima	208	60	65%
house	208	60	62%

Table 3.3: UCI Dataset Statistics

sis(*breast,diabetic,heart*), and discriminating the bouncing source of sonar signals(*sonar*). The variety of imbalance from 53% to 93% among our datasets also helps us to stress our framework to label imbalances.

Baselines

We compare the MIPSC with three other baselines.

Random baseline chooses samples to reject randomly.

Predictive mean baseline chooses the closest samples to have 0.5 predictive mean to be rejected (Chow (1970); Grandvalet *et al.* (2009)).

Model uncertainty baseline chooses the samples with the highest standard deviation to be rejected (Gal and Ghahramani (2016); Leibig *et al.* (2017)).

Comparison with *random* baseline helps us to investigate if using *predictive mean* or *model uncertainty* adds any value to find optimal decisions when rejecting. Comparing

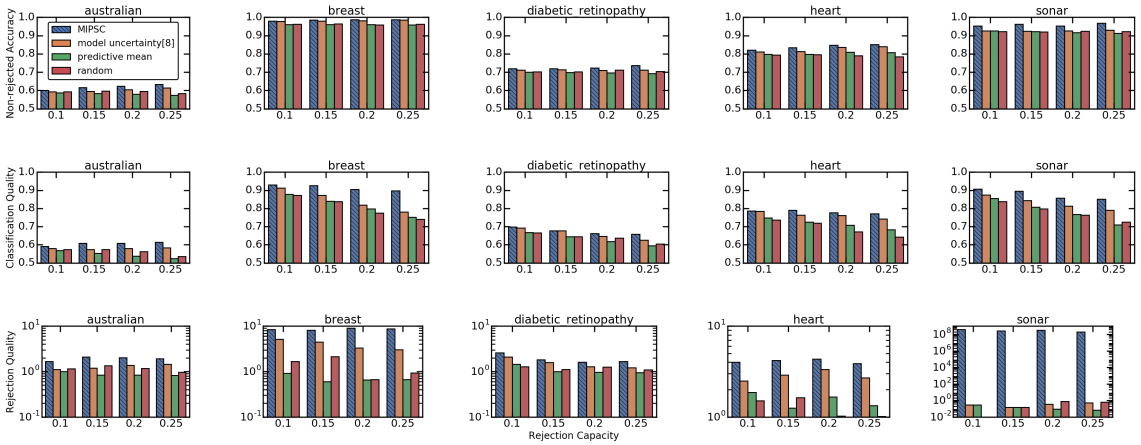


Figure 3.3: Performance of the MIPSC and Other Baselines under Varying Rejection Capacities. Notice the Superior Performance of MIPSC over the Recent State-of-the-art and Other Baselines in All of the Three Performance Metrics for Publicly Available Datasets: Australian, Breast, Diabetic, Heart, Sonar

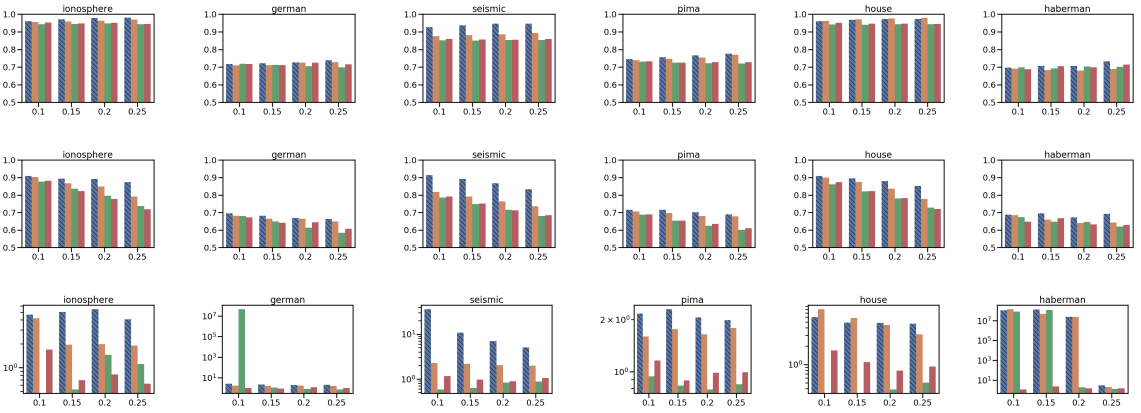


Figure 3.4: Performance of the MIPSC and Other Baselines under Varying Rejection Capacities. Notice the Superior Performance of MIPSC over the Recent State-of-the-art and Other Baselines in All of the Three Performance Metrics for Publicly Available Datasets: Ionosphere, German, Seismic, Pima, House, Haberman

MIPSC with *predictive mean* and *model uncertainty* separately allows us to investigate if they are complementary in optimal decision making for classification with reject option.

Results

Figure 3.3 and Figure **??** show the performance of MIPSC, and the other baselines. We make the following observations;

Store	Transactions	Fraud-Ratio	Avg. Amount($)
Digital Goods	67,215	8.1%	$79.29
Office Supplies	10,678	17.2%	$330.10
Sporting Goods	6,968	3.5%	$296.34

Table 3.4: Online Purchase Transactions Dataset Statistics

- In all datasets and three evaluation metrics, MIPSC achieves a consistent superior performance compared to the baselines.
- Higher rejection capacities yield higher non-rejected accuracy, as expected.
- Model uncertainty baseline consistently performs the second best signaling a better characterization of rejection than predictive mean.
- Predictive mean can be a complementary in classification with rejection task as evidenced by MIPSC higher performance than model uncertainty baseline. By itself, predictive mean baseline achieves similar performances to the random baseline.

3.3.4 Online Fraud Management

In this section, we discuss the contribution of our cost-sensitive framework MIPCSC over industry-standard baselines in online fraud management tasks. We design our experiments with three real-world e-commerce online transaction datasets coming from digital goods, office supplies, and sporting goods stores. Summary statistics of our datasets can be seen in Table 3.4.

In online fraud management, our base task is to classify each transaction instance as legitimate or fraudulent. Different than a standard classification task, benefits and costs of each true and false classification vary with the transaction amount of each transaction instance. Moreover, true classification of a legitimate transaction and a fraudulent transaction do not bring same amount of benefit. False classification of legitimate transaction and a fraudulent transaction incurs different costs as well (i.e., customer insult, fraud loss).

Our task also involves rejecting making classification when uncertain. In online fraud management domain, "rejecting to make a decision" equates to sending the transaction instance to an expert to be reviewed. This process of rejecting to make a decision also comes with a cost. By taking all these aforementioned costs and benefits of the task into consideration, here we present the final profit gain that our framework and several other fraud management strategies achieve on three real-world datasets.

Baselines

We compare the MIPCSC with four other baselines. We adopt two of them from the previous section (model uncertainty and random) and introduce two new cost-sensitive baselines.

Transaction amount baseline rejects to classify the instances with the largest transaction amounts. Majority of the transaction processors follows this conservative strategy.

Risk baseline rejects to classify the instances based on both model uncertainty and transaction amount. It multiplies the model uncertainty and transaction amount and rejects to classify the ones with the highest value.

Comparing MIPCSC with transaction amount baseline helps to assess whether our approach performs better than the most conservative fraud management strategy. Comparing MIPCSC with risk baseline assist with understanding if our approach is capable of making better assessments of cost-sensitive decisions than a simple arithmetic cost-sensitive risk measurement.

Results

Figure 3.5 shows the performance of MIPCSC compared to the other baselines. Our key observations are given as follows:

- Under varying capacities of rejection, MIPCSC always achieves the highest profit gain.
- In the Digital Goods dataset, underlying DNN performs worse than outputting a

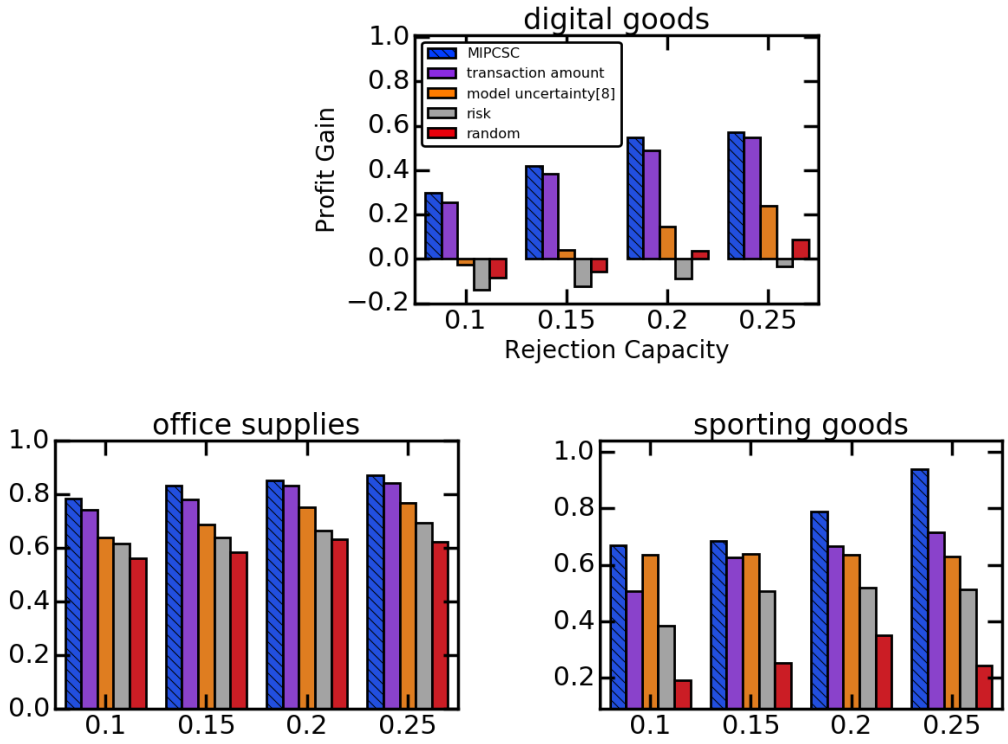


Figure 3.5: Profit Gain of MIPCSC vs. Baselines for Fraud Management.

trivial solution, thus causes uncertainty based baselines to obtain negative profit gain at various rejection capacities. It is clear that MIPCSC is robust to the underlying DNN performance giving the highest profit gain in all cases.

- Constant inferior performance of the Risk baseline suggests that simply combining uncertainty with a value aspect does not help making a cost-optimal decision. The necessity of a framework like MIPCSC becomes apparent observing its constant effectiveness.

Chapter 4

CONCLUSION

In this dissertation, I proposed novel cost-insensitive and cost-sensitive methods for selective classification and demonstrated their effectiveness in online fraud management domain. Here, I briefly summarize these methods and the contributions.

First, I provided a brief introduction to the selective classification problem in fraud management and gave a detailed literature survey in cost-sensitive learning, in Chapter 1 and Chapter 2.

In Chapter 3, I proposed a cost-sensitive decision making framework and demonstrate its effectiveness in fraud management. I revealed 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, investigating our framework's generalizability in other domains such as loan evaluation and healthcare decision support might be of interest.

In Chapter 4, I introduced MIPSC: a novel and extensible selective classification model that effectively utilizes uncertainty in deep learning and combines it with predictive mean to make optimal decisions. I demonstrated MIPSC's effectiveness using state-of-the-art selective classification metrics in publicly available datasets from various domains. I found that predictive mean is complementary to model uncertainty for making optimal reject decisions. Furthermore, I showcased a real-world use-case of online fraud management using our cost-sensitive extension, MIPSCS. Future work includes (1) experimenting with other Bayesian frameworks and (2) optimizing the MIP performance by designing novel column generation techniques.

REFERENCES

Abadi, M. *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems", URL `https://www.tensorflow.org/` (2015).

ACFE, "Report to the nations on occupational abuse and fraud: 2016 global fraud study", Report, Association of Certified Fraud Examiners, Inc (2016).

Amodei, D., C. Olah, J. Steinhardt, P. Christiano, J. Schulman and D. Mané, "Concrete problems in ai safety", arXiv preprint arXiv:1606.06565 (2016).

Asif, K., W. Xing, S. Behpour and B. D. Ziebart, "Adversarial cost-sensitive classification.", in "UAI", pp. 92–101 (2015).

Attenberg, J. and F. Provost, "Online active inference and learning", in "Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining", KDD '11, pp. 186–194 (ACM, New York, NY, USA, 2011), URL `http://doi.acm.org/10.1145/2020408.2020443`.

Bertsimas, D., A. King, R. Mazumder *et al.*, "Best subset selection via a modern optimization lens", The annals of statistics **44**, 2, 813–852 (2016).

Bixby, R. E., "Mixed-integer programming: It works better than you may think", (2010).

Bolton, R. J. and D. J. Hand, "Statistical fraud detection: A review", Statistical science pp. 235–249 (2002).

Carneiro, N., G. Figueira and M. Costa, "A data mining based system for credit-card fraud detection in e-tail", Decision Support Systems **95**, 91–101 (2017).

Chai, X., L. Deng, Q. Yang and C. X. Ling, "Test-cost sensitive naive bayes classification", in "Data Mining, 2004. ICDM'04. Fourth IEEE International Conference on", pp. 51–58 (IEEE, 2004).

Chan, P. K., W. Fan, A. L. Prodromidis and S. J. Stolfo, "Distributed data mining in credit card fraud detection", IEEE Intelligent Systems and Their Applications **14**, 6, 67–74 (1999).

Chow, C., "On optimum recognition error and reject tradeoff", IEEE Transactions on information theory **16**, 1, 41–46 (1970).

Condessa, F., J. Bioucas-Dias and J. Kovaevi, "Performance measures for classification systems with rejection", Pattern Recognition **63**, 437 – 450, URL `http://www.sciencedirect.com/science/article/pii/S0031320316303260` (2017).

CyberSource, "2016 North America online fraud benchmark report", Report, CyberSource Corporation (2016).

CyberSource, "2017 North America online fraud benchmark report", Report, CyberSource Corporation (2017).

Domingos, P., "Metacost: A general method for making classifiers cost-sensitive", in "Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining", pp. 155–164 (ACM, 1999).

Drummond, C. and R. C. Holte, "Exploiting the cost (in) sensitivity of decision tree splitting criteria", in "ICML", vol. 1 (2000).

El-Yaniv, R. and Y. Wiener, "On the foundations of noise-free selective classification", Journal of Machine Learning Research **11**, May, 1605–1641 (2010).

Elkan, C., "The foundations of cost-sensitive learning", in "International joint conference on artificial intelligence", vol. 17, pp. 973–978 (LEA, 2001).

Fan, W., S. J. Stolfo, J. Zhang and P. K. Chan, "Adacost: Misclassification cost-sensitive boosting", in "Proceedings of the Sixteenth International Conference on Machine Learning", ICML '99, pp. 97–105 (Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999), URL `http://dl.acm.org/citation.cfm?id=645528.657651`.

Fawcett, T. and F. Provost, "Adaptive fraud detection", Data mining and knowledge discovery **1**, 3, 291–316 (1997).

Fawcett, T. and F. Provost, "Activity monitoring: Noticing interesting changes in behavior", in "Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining", pp. 53–62 (ACM, 1999).

Gal, Y., "Uncertainty in deep learning", University of Cambridge (2016).

Gal, Y. and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning", in "international conference on machine learning", pp. 1050–1059 (2016).

Geifman, Y. and R. El-Yaniv, "Selective classification for deep neural networks", in "Advances in Neural Information Processing Systems 30", edited by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan and R. Garnett, pp. 4878–4887 (Curran Associates, Inc., 2017), URL `http://papers.nips.cc/paper/7073-selective-classification-for-deep-neural-networks.`

Geißler, B., A. Martin, A. Morsi and L. Schewe, "Using piecewise linear functions for solving minlp s", in "Mixed integer nonlinear programming", pp. 287–314 (Springer, 2012).

Ghosh, S. and D. L. Reilly, "Credit card fraud detection with a neural-network", in "System Sciences, 1994. Proceedings of the Twenty-Seventh Hawaii International Conference on", vol. 3, pp. 621–630 (IEEE, 1994).

Goh, S. T. and C. Rudin, "Box drawings for learning with imbalanced data", in "Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining", pp. 333–342 (ACM, 2014).

Gottschalk, P., "Categories of financial crime", Journal of financial crime **17**, 4, 441–458 (2010).

Grandvalet, Y., A. Rakotomamonjy, J. Keshet and S. Canu, "Support vector machines with a reject option", in "Advances in Neural Information Processing Systems 21", edited by D. Koller, D. Schuurmans, Y. Bengio and L. Bottou, pp. 537–544 (Curran Associates, Inc., 2009), URL `http://papers.nips.cc/paper/3594-support-vector-machines-with-a-reject-option.pdf`.

Griva, I., S. G. Nash and A. Sofer, *Linear and nonlinear optimization*, vol. 108 (Siam, 2009).

Halvaiee, N. S. and M. K. Akbari, "A novel model for credit card fraud detection using artificial immune systems", Applied Soft Computing **24**, 40–49 (2014).

Herbei, R. and M. H. Wegkamp, "Classification with reject option", Canadian Journal of Statistics **34**, 4, 709–721 (2006).

Hooi, B. *et al.*, "Fraudar: Bounding graph fraud in the face of camouflage", in "Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining", pp. 895–904 (ACM, 2016).

Interpol, "Financial and high-tech crimes", Report, International Criminal Police Organization (2009).

Juniper, "Online payment fraud: Emerging threats, key vertical strategies & market forecasts 2017-2022", Whitepaper, Juniper Research (2017).

Kim, M., "Large margin cost-sensitive learning of conditional random fields", Pattern Recognition **43**, 10, 3683–3692 (2010).

Kruskal, J. B., "Nonmetric multidimensional scaling: a numerical method", Psychometrika **29**, 2, 115–129 (1964).

KS&R, "2016 LexisNexis true cost of fraud study", Report, LexisNexis Risk Solutions (2016).

Leibig, C., V. Allken, M. S. Ayhan, P. Berens and S. Wahl, "Leveraging uncertainty information from deep neural networks for disease detection", Scientific reports **7**, 1, 17816 (2017).

Li, H., L. Zhang, B. Huang and X. Zhou, "Sequential three-way decision and granulation for cost-sensitive face recognition", Knowledge-Based Systems **91**, 241–251 (2016).

Ling, C. and V. Sheng, "Cost-sensitive learning and the class imbalance problem", (2008).

Ling, C. X., Q. Yang, J. Wang and S. Zhang, "Decision trees with minimal costs", in "Proceedings of the twenty-first international conference on Machine learning", p. 69 (ACM, 2004).

Maes, S., 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).

Margineantu, D. D., "Active cost-sensitive learning", in "IJCAI", vol. 5, pp. 1622–1623 (2005).

Masnadi-Shirazi, H. and N. Vasconcelos, "Cost-sensitive boosting", IEEE Transactions on pattern analysis and machine intelligence **33**, 2, 294–309 (2011).

Masnadi-Shirazi, H., N. Vasconcelos and A. Iranmehr, "Cost-sensitive support vector machines", arXiv preprint arXiv:1212.0975 (2012).

Montague, D. A., *Essentials of online payment security and fraud prevention*, vol. 54 (John Wiley & Sons, 2010).

Ngai, E. *et al.*, "The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature", Decision Support Systems **50**, 3, 559–569 (2011).

Niculescu-Mizil, A. and R. Caruana, "Obtaining calibrated probabilities from boosting.", in "UAI", p. 413 (2005).

Nilson, "Card fraud losses reaches $21.84 billion", Report, The Nilson Report (2016).

Pednault, E., N. Abe and B. Zadrozny, "Sequential cost-sensitive decision making with reinforcement learning", in "Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining", pp. 259–268 (ACM, 2002).

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

Pickett, K. S. and J. M. Pickett, *Financial crime investigation and control* (John Wiley & Sons, 2002).

Platt, J. *et al.*, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods", Advances in large margin classifiers **10**, 3, 61–74 (1999).

Polson, N. G., V. Sokolov *et al.*, "Deep learning: a bayesian perspective", Bayesian Analysis **12**, 4, 1275–1304 (2017).

Provost, F. and P. Domingos, "Tree induction for probability-based ranking", Machine learning **52**, 3, 199–215 (2003).

Rasmussen, C. E., "Gaussian processes for machine learning", (MIT Press, 2006).

Santos-Pereira, C. M. and A. M. Pires, "On optimal reject rules and roc curves", Pattern Recognition Letters **26**, 7, 943 – 952, URL http://www.sciencedirect.com/science/article/pii/S0167865504002892 (2005).

Sheng, V. S. and C. X. Ling, "Thresholding for making classifiers cost-sensitive", in "AAAI", pp. 476–481 (2006).

Sun, Y., M. S. Kamel, A. K. Wong and Y. Wang, "Cost-sensitive boosting for classification of imbalanced data", Pattern Recognition **40**, 12, 3358–3378 (2007).

Tortorella, F., "An optimal reject rule for binary classifiers", in "Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)", pp. 611–620 (Springer, 2000).

Turney, P. D., "Cost-sensitive classification: Empirical evaluation of a hybrid genetic decision tree induction algorithm", Journal of artificial intelligence research **2**, 369–409 (1995).

Van Vlasselaer, V. *et al.*, "Apate: A novel approach for automated credit card transaction fraud detection using network-based extensions", Decision Support Systems **75**, 38–48 (2015).

Wang, T., Z. Qin, S. Zhang and C. Zhang, "Cost-sensitive classification with inadequate labeled data", Information Systems **37**, 5, 508–516 (2012).

Yang, F., H.-z. Wang, H. Mi, W.-w. Cai *et al.*, "Using random forest for reliable classification and cost-sensitive learning for medical diagnosis", BMC bioinformatics **10**, 1, S22 (2009).

Yildirim, M. Y., M. Ozer and H. Davulcu, "Cost-sensitive decision making for online fraud management", in "Artificial Intelligence Applications and Innovations", edited by L. Iliadis, I. Maglogiannis and V. Plagianakos, pp. 323–336 (Springer International Publishing, Cham, 2018).

Zadrozny, B., J. Langford and N. Abe, "Cost-sensitive learning by cost-proportionate example weighting", in "Data Mining, 2003. ICDM 2003. Third IEEE International Conference on", pp. 435–442 (IEEE, 2003).

Zafarani, R. and H. Liu, "10 bits of surprise: Detecting malicious users with minimum information", in "Proceedings of the 24th ACM International on Conference on Information and Knowledge Management", pp. 423–431 (ACM, 2015).

Zhang, S., Z. Qin, C. X. Ling and S. Sheng, "" missing is useful": missing values in cost-sensitive decision trees", IEEE transactions on knowledge and data engineering **17**, 12, 1689–1693 (2005).

Zhang, S. *et al.*, "Hidden: Hierarchical dense subgraph detection with application to financial fraud detection", in "Proceedings of the 2017 SIAM International Conference on Data Mining", pp. 570–578 (SIAM, 2017).

Zhou, D. *et al.*, "A local algorithm for structure-preserving graph cut", in "Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining", pp. 655–664 (ACM, 2017).

Zubek, V. B. and T. G. Dietterich, "Pruning improves heuristic search for cost-sensitive learning", in "ICML", (2002).