UNIVERSITY OF CALIFORNIA
SANTA CRUZ

STRATEGIC CLASSIFICATION: A GAME-THEORETIC APPROACH

A dissertation submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

ELECTRICAL ENGINEERING

by

Lemonia Dritsoula

September 2015

The Dissertation of Lemonia Dritsoula is approved:

_____________________________
Professor John Musacchio, Chair

_____________________________
Professor Alyson Fletcher

_____________________________
Professor Daniel Friedman

_____________________________
Tyrus Miller
Vice Provost and Dean of Graduate Studies
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>List of Figures</td>
<td>vi</td>
</tr>
<tr>
<td>List of Tables</td>
<td>viii</td>
</tr>
<tr>
<td>Abstract</td>
<td>ix</td>
</tr>
<tr>
<td>Dedication</td>
<td>xi</td>
</tr>
<tr>
<td>Acknowledgments</td>
<td>xii</td>
</tr>
<tr>
<td><strong>1 Introduction</strong></td>
<td>1</td>
</tr>
<tr>
<td>1.1 Overview: Adversarial Classification (Part I)</td>
<td>3</td>
</tr>
<tr>
<td>1.2 Overview: Click Fraud Classification (Part II)</td>
<td>4</td>
</tr>
<tr>
<td>1.3 Research Contributions</td>
<td>7</td>
</tr>
<tr>
<td><strong>I Adversarial Classification</strong></td>
<td>9</td>
</tr>
<tr>
<td><strong>2 A Game Theoretic Analysis of Adversarial Classification</strong></td>
<td>10</td>
</tr>
<tr>
<td>2.1 The classification game</td>
<td>13</td>
</tr>
<tr>
<td>2.2 Justification of threshold strategies</td>
<td>17</td>
</tr>
<tr>
<td>2.2.1 Defender’s reduced strategy space</td>
<td>19</td>
</tr>
<tr>
<td>2.2.2 Reduced attacker’s strategy space and equilibrium structure</td>
<td>27</td>
</tr>
<tr>
<td>2.3 Nash Equilibrium Analysis</td>
<td>31</td>
</tr>
<tr>
<td>2.3.1 Form of optimal extreme points</td>
<td>37</td>
</tr>
<tr>
<td>2.3.2 Form of players’ equilibrium strategy</td>
<td>43</td>
</tr>
<tr>
<td>2.4 Numerical Results</td>
<td>46</td>
</tr>
<tr>
<td>2.4.1 Single target</td>
<td>46</td>
</tr>
<tr>
<td>2.4.2 Investment strategies of the defender</td>
<td>51</td>
</tr>
<tr>
<td>2.5 Related Work</td>
<td>54</td>
</tr>
<tr>
<td>2.6 Concluding Remarks</td>
<td>56</td>
</tr>
</tbody>
</table>
## Classification game with generalized payoff functions

3.1 Game Model .............................................................................. 60
   3.1.1 Spy’s cost function ......................................................... 61
   3.1.2 Defender’s payoff function ............................................. 62
   3.1.3 Players’ interactions ....................................................... 63
3.2 Game Theoretic Analysis .......................................................... 64
3.3 Evaluation with model examples .............................................. 74
   3.3.1 Example Model 1 ............................................................. 76
   3.3.2 Example Model 2 ............................................................. 77
3.4 Parameter effects in the Game .................................................. 78
   3.4.1 Effect of the probability of the spy \( p \) and false alarm cost \( c_{fa} \) ........................................ 79
   3.4.2 Effect of the detection cost \( c_d \), the classification window \( N \) and single FS attack cost \( c_a \) .................. 80
   3.4.3 Effect of the spammer’s distribution parameter \( \theta_0 \) .......... 83
3.5 Related work ........................................................................... 85
3.6 Conclusions ............................................................................ 87

## Click Fraud Classification

II Click Fraud Classification .......................................................... 89

4 Economic Incentives to Fight Click Fraud in Advertising Markets 90
   4.1 Economic Model ................................................................. 92
      4.1.1 Ad networks ................................................................. 93
      4.1.2 Publishers ................................................................. 94
      4.1.3 Advertisers ................................................................. 96
   4.2 Equilibria ............................................................................ 96
      4.2.1 Heterogeneous publishers \((g = 0)\). ............................... 97
      4.2.2 General model for publishers \((g > 0)\). ......................... 98
   4.3 Experiments ....................................................................... 99
   4.4 Related Work ..................................................................... 100
   4.5 Conclusions ....................................................................... 103

5 On Nash Equilibrium Investment Strategies Against Click Fraud in Online Advertising Markets 104
   5.1 Economic model ................................................................. 106
      5.1.1 Ad networks ................................................................. 107
      5.1.2 Publishers ................................................................. 112
      5.1.3 Advertisers ................................................................. 114
   5.2 Equilibrium Analysis ............................................................ 115
      5.2.1 One-shot game ........................................................... 116
      5.2.2 Stage game ................................................................. 117
   5.3 Numerical experiments ......................................................... 117
      5.3.1 Single-shot game ......................................................... 118
      5.3.2 Two-stage game ........................................................ 119
   5.4 Conclusion ......................................................................... 121
6 Conclusions and Future Work

6.1 Summary of Contributions ........................................... 123
6.2 Future Work ............................................................. 124
   6.2.1 Part I: Adversarial Classification .............................. 124
   6.2.2 Part II: Click Fraud Classification ............................. 125

A Deriving the advertisers’ bidding function using the Capital Asset Pricing Model

A.1 The Capital Asset Pricing Model ..................................... 127
A.2 Applying CAPM on the advertiser’s bidding functions ............ 128

Bibliography ................................................................. 131
## List of Figures

2.1 There are three cases for strategies selected in NE. ......................... 22
2.2 The attacker’s reduced strategy space is the set of all images of function $R.$ 27
2.3 The attacker uses a truncated version of the distribution of the non-attacker in NE. ................................................................. 49
2.4 As each attack becomes more and more valuable, the attacker’s payoff is increasing while the defender’s payoff is decreasing. ................. 50
2.5 As the false alarm penalty increases, the defender becomes more conservative in his classification rule and his Nash equilibrium payoff decreases, while the attacker benefits more. .................................................. 51
2.6 Scenario 1: Defender does not differentiate between low and high “inactive-pct”. Scenario 2: Attacker mistakenly believes he is only classified based on a single feature. Scenario 3: Defender mistakenly believes that attacker believes he is classified based on a single feature. Scenario 4: both players know that two sensors (features) determine classification. ............... 54
3.1 As $p$ increases, the NE defender gain with Model 2 increases. .............. 81
3.2 For a small detection cost, the spy is more conservative in Model 2. .......... 82
3.3 Effect of $\theta_0$ on the players’ NE strategies ($N = 5$, $c_a = 1$, $c_d = c_{fa} = 10$, $p = 0.3$). The bar left and right of numeral represents the defender and the spy respectively. ............................................................. 84
3.4 The defender’s NE payoff decreases as $\theta_0$ increases for all values of $p$ for Model 1, but only for $p < 0.5$ for Model 2. ................................. 85
4.1 Publishers’ preferences are uniformly distributed between $\text{AN}_1$ and $\text{AN}_2$. 92
4.2 ROC curve: If the ad networks are willing to tolerate a false positive rate of $x_i$, they can achieve a true positive rate of $x_i^\alpha$. ................................. 94
4.3 When symmetric, the ad networks select symmetric equilibrium prices. . . . 100
4.4 As publishers become more homogeneous, the inferior network ($\text{AN}_2$) offers a larger and larger share to publishers than the superior network. ............ 101
5.1 Precision vs. Recall curve: If the ad networks are willing to achieve a recall (true positive rate) of 0.8, they can achieve a precision of $p(0.8) = 0.97$. .... 110
5.2 The advertising market comprises of three entities: advertisers who pay for clicks deemed valid, ad networks who classify clicks as valid and publishers who display ads. .................................................. 111
5.3 Publishers’ preferences are uniformly distributed between AN\textsubscript{1} and AN\textsubscript{2}. . . 112
5.4 In stage 1 (\textit{G}_2) the players select how much to invest while in stage 2 (\textit{G}_1) they select their recall. ................................................................. 115
5.5 Equilibrium recall levels in a symmetric one-shot game (for a given \textit{N} = 3). 119
5.6 Equilibrium recall versus AN\textsubscript{2}’s investment: As the quality of one network increases, both networks select higher recall levels at equilibrium. . . . . . . . . 120
5.7 Over-investment versus under-investment for various values of revenue share \textit{h} and degree of heterogeneity \textit{t}. ................................................................. 121
# List of Tables

3.1 Defender's strategy in NE ($\beta_m = c_n/c_d$) ........................................ 77  
4.1 Ad networks make mistakes when filtering out invalid clicks. ............... 93  
4.2 List of variables introduced in Section 4.1. ........................................ 97  
5.1 Classification versus truth table ............................................................ 108  
5.2 List of variables introduced in Section 5.1 ........................................... 115
Abstract

Strategic Classification: A Game-Theoretic Approach

by

Lemonia Dritsoula

Entities of physical presence have always been susceptible to attacks. Entities of online presence are more prone to cyberattacks, as the lack of local boundaries permits anyone from anywhere in the world to launch a potentially successful attack. People aim to protect their physical and virtual assets daily: Locking doors and cars, setting up alarms and protecting passwords and credit card numbers are just a few examples. History has proven, though, that any defense system can be compromised. Defenders and attackers have delved into continuous action and counteraction: The more intelligent the defense strategy gets, the more sophisticated the attacker becomes, and the defense needs to adapt again. Game Theory provides us with the mathematical tools and concepts to capture these strategic interactions between players and their different and often antagonistic interests. Such a concept is a Nash equilibrium: a set of strategies that dictates what the players should do, such that the players have no incentive to deviate.

We model the strategic interactions between a strategic adversary and a classifier using Game Theory. Our analysis leads to Nash equilibria in randomized strategies, increasing the difficulty for the attacker to reconstruct the classification rule. We give a polynomial time algorithm to compute the Nash equilibrium strategies of the players. Our setting, albeit simple, gives interesting insights that could be beneficial in practical settings:
We quantify the impact of the various parameters like cost from detection, false alarms and statistics about the normal traffic on the equilibrium strategies and payoffs. We show that if the classification is based on several features, the defender should apply a threshold on the attacker’s reward. This is in contrast with known algorithms such as logistic regression which have a predefined shape of the boundary independently of the attacker’s goal.

In view of the recent security breaches, like Anthem, Target and Home Depot, we are motivated to study a class of games in which there are multiple defenders (companies) that compete to acquire legitimate users. To this purpose we analyze the economic incentives for advertising networks to fight click fraud. By click fraud, we define the act of clicking an ad without an interest to see the ad or buy the advertised product. When such clicks are counted as “valid” by the ad network, the advertiser pays for a useless click, and the publisher (webpage at which the ad was displayed) is rewarded for generating it. Thus there is an incentive for fraudulent publishers to inflate click numbers. Ad networks have an incentive to classify fraudulent clicks in order to deliver a more valuable service to advertisers. On the other hand, ad networks do receive revenue for fraudulent clicks, which creates an incentive in the opposite direction to fight fraud less.

We develop a model to capture the incentives of two competing ad networks to fight click fraud, under different assumptions about the bidding process of advertisers. Our equilibrium results show that ad networks are highly incentivized in combating click fraud. We explore the investment strategies of ad networks to improve their classification algorithms and investigate when ad networks over or under invest compared to social optimum.
To my husband Petros,

for always inspiring me to excel.
Acknowledgments

Foremost, I would like to thank my advisor, John Musacchio, for providing me with the opportunity to work on game-theoretic problems and being patient throughout my endeavors. I thank him for giving me the freedom to explore my interests and for teaching me among others, academic values, intellectual curiosity and scientific honesty.

The numerous meetings with John, his right-on advice, patience and feedback established my research direction and improved my research skills. John is an extremely smart person but not an intimidating one. Being close to him and watching him explain complex concepts in simple terms has always been an inspiration and an example to follow when I articulate my own ideas, orally or written. He taught me how to better structure my papers, state the theorems, and present more effectively. I could never thank him enough.

I am grateful to Patrick Loiseau for working closely with me on parts of this dissertation. His critical comments and his strong mathematical background proved indispensable in helping me learn how to properly write rigorous proofs. I enjoyed our collaboration and our fruitful discussions, and I would like to thank him for his time and feedback both during his visits in California and our Skype calls.

I thank Luca de Alfaro for serving on my orals committee and for always providing to-the-point feedback over the years. I also want to thank Dan Friedman for serving on my orals and my reading committee. Dan was among the first teachers I have had on Game Theory and I thank him for his generosity with his time and ideas on parts of my work. I am also grateful to Peyman Milanfar who served as the chair in my preliminary examination and Alyson Fletcher who served on my orals committee and my reading committee.
I feel very privileged to have started my academic research career with an advisor such as Constantinos Papadias during my Masters thesis in Greece. He was the first person who believed in me, who treated me as his peer throughout our discussions and motivated me with his positive attitude and optimism to pursue a PhD in the US. He is still a great mentor and I would not be here writing this dissertation if I had not met him. I will be eternally grateful to him for his advice, trust, and support.

Being a PhD student has been a great experience, due to my lab mates at UCSC and at the NetEcon Group at UC Berkeley. Bita, Michael, Amar, Patrick and Braden, thank you for making my life at UCSC more enjoyable. Jean Walrand, Galina Schwartz, Stephan Adams, and Vijay Kamble, thank you for your feedback throughout the various talks and for providing me with a stimulating environment. It was great spending most of my Fridays at Berkeley. The School of Engineering at UCSC has proved a great host for me for the last years. I thank Carol, Tracie and Emily for making my life easier.

While a PhD student, I spent a summer interning at the Algorithms team of Stitch Fix. I thank my managers, Bhaskar Rao and Jeff Magnusson for mentoring me and helping me getting onboard with my first internship in a tech company and Eric Colson for providing me with this amazing opportunity to get hands on experience outside the research world.

I want to express my gratitude to my friends in Greece for keeping our long distance relationship vivid: Stamatia, Eleni, Fotis, Theodosis, Sarantis, Artemis, George, Dimitra, thank you for making my life in Greece as enjoyable as it was before I left.

To my Bay Area friends, Vicky and Yiota, Arlette and Elena, Dido and Magdalini, Maria and Winnie, Panayiotis and Spiros, Panagiotis and Christos, Michalis and Elizabetta,
George and Dimitris, Pigi and Ioannis, Dimitris and Alkis, thank you for bearing with me the stress that a life of a PhD student entails and supporting me in various aspects of my life.

I would also like to thank the baristas at Cafe Venetia and Coupa Cafe in Palo Alto, and Brewed Awakening in Berkeley, for providing me with the right resources, i.e., coffee, pastries, Internet and electricity to fulfill most of the work in this thesis.

I want to thank my parents, Konstantinos and Anastasia, and my sister, Marianthi, for their love and support through the years. Also, I would like to thank my grandmother Lemonyia who at her 82 years old was the first family member who encouraged me to study abroad and leave Greece. She is still counting the years for me to go back. I would like to thank my aunt Georgia for teaching me how to cook greek meals over the phone and sending me precious (full of goodies) packages. My parents-in-law Eleni and Ioannis Venetis have always been there for me providing me with balance and showering me with love as if I were their daughter. Gloria, Vassilios, Margarita and Nick were the closest family I’ve had in the US and I thank them for their love and support throughout the years.

Lastly but not least, I would like to thank my husband, Petros Venetis, for being a star worth following. He has always been there for me, from sharing notes in our undergrad studies in Greece to sharing our first PhD years together in California. Petros helped me fight my weaknesses and become a better researcher and person. And foremost I thank him for continuously pushing me to excel. I appreciate every moment of his companionship.
Chapter 1

Introduction

“Statistical analysis. The minimum number of actions it will take to win the war, but the maximum number we can take before the Germans get suspicious.”

Alan Turing

Today’s interconnected world brings opportunity but also risks. Recent breach incidents such as the ones involving Anthem, Home Depot, Target and Sony Pictures have raised the interest on cybersecurity on a national and worldwide level: More than 100 million users have seen their private data leaked or compromised. The more companies and nations invest in improving their defenses, the more intelligent and sophisticated the attackers become to evade detection and stay stealthy.

Standard classification algorithms developed by the machine learning and statistics communities have been used to perform this task relatively successfully for some time.
However, it then became apparent that attackers can become aware of the classification algorithms being used to detect their attacks and are able to react so that the classification algorithm becomes less efficient to prevent attacks. The role of Game Theory becomes more and more important in such a setting, inherently capturing the different and sometimes opposing tractions between defenders and attackers. It provides us with the mathematical tools to study the strategic behavior of the attackers who seek to evade detection by classifiers.

The goal of this thesis is:

\textit{to develop game-theoretic models capable of capturing the interactions between a defender and an attacker in a strategic classification game and use these models to: (a) predict the equilibrium attack and defense strategies of the adversary and the defender respectively, (b) qualitatively and quantitatively explore the impact of the underlying parameters on the equilibrium strategies and payoffs, and (c) investigate investment incentives of the defenders to increase the quality of their classification algorithms in both the presence and the absence of competition.}

In this thesis, we use a game-theoretic approach to strategic classification. Chapters 2-3 focus on modeling the interactions of a defender and attacker as a game in the realistic case where the objectives of both are not opposite. Chapters 4-5 add another dimension in the classification problem. They allow competition between two defenders that compete to acquire customers while classifying traffic as fraudulent or legitimate. We study the investment equilibrium strategies between two competitors and explore interesting dilemmas that they might face. The rest of this chapter gives a high level overview of the two major parts of this dissertation.

**Part I** Adversarial Classification (Section 1.1).

**Part II** Click Fraud Classification (Section 1.2).
Lastly, we summarize our research contributions (Section 1.3). Note that related work will be included in each individual chapter.

1.1 Overview: Adversarial Classification (Part I)

Part I begins our game-theoretic study of strategic classification by studying an adversarial classification game. *Adversarial classification games* are played between a classifier that attempts to distinguish whether some activity is coming from a strategic adversary (positive class) or a benign user (negative class), and a strategic adversary that attempts to make the classifier classify positive instances as negative.

In Chapter 2, we investigate attacker classification games in which intelligent adversaries have an incentive to manipulate their attacks to get misclassified. The classification is based on a feature vector that might be generated by the attacker or by the non-attacker (benign user). Game theory inherently succeeds in capturing the strategic interaction between the players: the defender needs to balance the expected cost of missed detections and false alarms, while the attacker trades off the potential increased reward of a more aggressive attack vector with his increased risk of being detected.

We completely characterize all Nash equilibrium strategies of the players in polynomial time; such description cannot be derived by a solution with a linear programming tool. We further analyze the impact of the non-attacker on the attacker’s equilibrium strategy and conclude that the attacker’s equilibrium strategy is a truncated version of the non-attacker’s distribution. We also investigate whether a strategic defender should invest in capturing more data about the behavior of the attackers. We conclude by presenting
examples on which our theory / results can be applied, that vary from spam detection to any problem in which someone needs to differentiate two entities A and B, when one of the two is strategic.

Chapter 3 evaluates an instance of the game developed in the previous chapter, but drops the assumption that the benefit from detection is constant. We focus on a simple one-shot game between an attacker and a defender. We first develop parameterized families of payoff functions for both players and analyze the Nash equilibria of the noncooperative nonzero-sum game. Based on certain assumptions on the nature of these payoff functions, we analyze the equilibrium strategies in the classification game between a defender and an attacker. Using the results of the previous section, the defender strategically selects her classification policy: a threshold on the number of file server attacks. The non-attacker naively attacks (with a known distribution) his main target: the mail server. The attacker strategically selects the number of attacks on his main target: the file server. We give two examples of the general model, one that involves forensics on the side of the defender and one that does not. Finally, we evaluate how investments in forensics and data logging could improve the Nash equilibrium payoff of the defender.

1.2 Overview: Click Fraud Classification (Part II)

While adversarial classification is one of the most important applications that would benefit from combining machine learning and game theory, defenders and attackers do not live in an isolated world. People daily form their opinions on which companies they deem more secure; from their email or network providers to their healthcare providers and
their banks to business owners who care about which ad networks deliver better return on investment from their advertising campaigns. Industry is well aware of that, that is why companies invest in their security monitoring systems. Nobody wants the bad publicity that comes from a security breach or unsuccessful campaigns and thus the number of security start ups with security as a service as their selling point have multiplied just in the last couple of years.

Part II expands our study for adversarial classification to include a third entity: revenue generating users. Inspired by the most prominent manifestation of fraud we see daily online, we study the click fraud problem in advertising markets. Ad networks use revenue sharing and effective filtering of fraudulent clicks to attract publishers. The adversary is a malicious publisher who inflates the click rates to gain more revenue. The game is played though between ad networks who — competing for legitimate publishers — need to maintain a certain level of quality of their clicks for which the advertisers are charged. (We define the terms publishers, revenue sharing and effective filtering more explicitly in the two chapters.)

In Chapter 4, we develop a simple Hotelling competition-based game-theoretic model to study the effect of competition along these dimensions. We compute the Nash equilibrium strategies for two ad networks who compete for publishers. We then investigate how the preferences of the publishers and the quality of the ad networks affect the market share and the strategies chosen at equilibrium. Our results show that the more asymmetric in quality the ad networks are, the more asymmetric their equilibrium prices will be. Another finding of our work is that as the publishers become more heterogeneous (differentiated in their preferences), the competition in prices softens.
In Chapter 5, we focus again on the click fraud problem, in the presence of advertisers who cannot perfectly readjust their bids to maintain a certain return of investment. We compute the Nash equilibrium filtering strategies for two ad networks that compete for publishers in a single-shot game and conclude that maximum filtering is no longer a dominant strategy. There is a tradeoff between increasing the misdetection rate versus the false alarms that depends on the bidding of the advertisers.

We then study a two-stage game in which the ad networks first decide on their investment levels to improve their classification algorithms. By using the Capital Asset Pricing Model, we establish a model in which the optimal investment policies for the ad networks are derived. We are primarily interested in exploring the negative and positive externalities that the investment decisions of ad networks would trigger: positive externalities to the publishers and negative externalities to their competitors. Increasing the investment, an ad network would improve the accuracy of the classification algorithms, thus increasing the advertisers’ bids. Do the ad networks over invest or underinvest comparing to the social optimum? Which force of the two externalities dominates the other? How does the cost for investment and the underlying parameters (revenue share, fraction of fraudulent clicks, platform homogeneity) affect the resulting investment strategies? The answers to these questions are very important to ad networks and their advertisers, as they can determine how much should be charged for advertisements and the future of pay per click advertising market. Our results suggest that as the networks’ revenue share decreases and as platform homogeneity decreases, there is a tendency for the ad networks to underinvest.

At the end of each of the aforementioned chapters, we present other works related
to the chapter topic. Finally, in Chapter 6 we present our closing remarks and discuss potential future work.

1.3 Research Contributions

In summary, the high level contributions of this thesis are the following:

1. **Part I** Adversarial Classification

   • We give a complete characterization of the Nash equilibria of the adversarial classification game and find that the defender must randomize the classification rule.

   • If the classification is based on several features, the defender should apply a threshold on the attacker’s benefit from attacks. This is in contrast with known algorithms such as logistic regression which have a predefined shape of the boundary independently of the attacker’s goal.

   • We analyze the investment tradeoffs that a strategic defender is facing, when she needs to decide to acquire more data about the attacker and conclude that it is not always beneficial to invest in more sensors.

2. **Part II** Click Fraud Classification

   • We present a model to capture the incentives of ad networks to fight click fraud.

   • Our results show what assumptions about the advertisers’ bidding favor maximum filtering as a dominant strategy for ad networks.
• We explore the negative and positive externalities that would force ad networks to over invest or under invest compared with the social optimum.
Part I

Adversarial Classification
Chapter 2

A Game Theoretic Analysis of Adversarial Classification

Classification is one of the most useful tools from machine learning. In its simplest instance, a classification algorithm trains a model from a dataset of labeled data samples of two different classes (generally called class 0 and class 1) and then uses this model to predict the class of a new data sample. Many classification algorithms such as SVM, logistic regression, or Naive Bayes have been developed over the past decades and have been successfully used in a large number of applications ranging from computer vision to biology or marketing.

One of the most prominent applications of classification, however, has been security. Typically, in security problems, a defender tries to detect attacks, that is to classify usage into normal usage (class 0) or attacks (class 1). In this framework, attacks can range from spams received at a user’s inbox (class 1) that must be distinguished from regular
emails (class 0) to more serious attacks such as DoS or malicious infiltrations on a server (class 1) that must be distinguished from random failures (class 0).

Indeed, a strategic attacker who is aware of the classification algorithm being used to detect his attacks can modify the distribution of his attacks (i.e., of the class 1 data samples) either to evade detection or to alter the algorithm’s training. The data distribution is therefore no longer exogenous and standard algorithms can perform very poorly. Some studies in the subfield of adversarial machine learning have studied the vulnerability of classification algorithms to malicious attackers modifying their attacks in reaction to the classification algorithm and sometimes proposed more robust algorithms (see Section 2.5). These studies, however, are restricted to worst-case analysis where the attacker’s objective is the opposite of the defender’s objective, i.e., the attacker aims to minimize the efficiency of the classification algorithm to detect attacks. In reality, attackers and defenders have their own costs and benefits and their objectives are not directly opposite. The worst-case assumption is overly pessimistic and algorithms derived from this assumption are suboptimal.

In this chapter, we use a game-theoretic approach to model the interaction between a defender using a classification algorithm to detect attacks and an attacker shaping his attacks to avoid detection in a realistic case where the objectives of the two players are not opposite. Specifically, we consider a game between an attacker and a defender, where the attacker chooses his attack pattern and the defender chooses a classification strategy. The attacker’s objective balances the benefit from attacks and the cost of being detected; the defender’s objective balances the benefit of a correct attack detection and the cost of
false alarm. This cost of false alarm makes the objective of the defender different from the opposite of the attacker’s objective and can be very significant. Note that our game is framed in the context of security where a defender needs to classify between an attack and a normal/benign usage, but it applies to any situation where there is a need to classify between two different classes of activities or users: class 0 is a normal user (non-attacker) while class 1 is a strategic and intelligent player who is trying to avoid classification, while achieving his goal.

**Contributions.** Our contribution can be summarized as follows. We give a complete analysis of the game’s Nash equilibria. Our analysis reveals several important and intuitive messages on classification in the presence of an attacker.

1. First, we find that the defender must randomize the classification rule.

2. Second, if the classification is based on several features (e.g., the number of attacks on each target if there are several targets), the defender should apply a threshold on the attacker’s benefit from attacks. This is in contrast with known algorithms such as logistic regression which have a predefined shape of the boundary independently of the attacker’s goal.

3. Third, the attacker should mimic the distribution of the normal behavior but only on a subset of the support with patterns that yield the highest payoff, and potentially add a peak of attack distribution on the pattern yielding the highest payoff.

4. Finally, our results allow us to analyze the investment tradeoffs that a strategic defender is facing, when she needs to decide to acquire more data about the attacker.
(e.g., investing in a new sensor). We provide the tool to help the defender decide whether she should invest or not, and what the lower and upper bounds of her investment should be (depending on her assumption / uncertainty on the various network parameters.)

Overall, our analysis strengthens our understanding on the equilibrium strategies of the players.

In Section 2.1 we present the classification we studied. In Section 2.2 we justify the selection of threshold strategies for the defender and the proportionality of the attacker’s equilibrium strategy to the non-attacker’s distribution. Section 2.3 provides a Nash equilibrium analysis that gives insights on the structure of the players’ equilibrium strategies (Theorem 2). Section 2.4 contains the simulation results, Section 2.5 summarizes related work, and we conclude in Section 2.6.

2.1 The classification game

We first summarize the main notational conventions used throughout the chapter.

**Notational Conventions:**

All vectors are assumed to be column vectors and are denoted by bold lowercase letters (e.g., $\mathbf{\alpha}$, $\mathbf{\beta}$). The inner product of two vectors $\mathbf{\alpha}$, $\mathbf{\beta}$ of size $n$ is denoted by $\mathbf{\alpha} \cdot \mathbf{\beta} = \sum_{i=1}^{n} \alpha_i \beta_i$.

For matrices we use capital greek letters (e.g., $\Lambda$). We denote matrix elements in row $i$ and column $j$ by $\Lambda(i,j)$. We use the prime sign (') for transpose of matrices and vectors.

We use “min $\mathbf{\alpha}$” to denote the minimum element of a vector $\mathbf{\alpha}$, and “minimize” when we minimize a specific expression over some constraints. The indicator function is denoted by
$1_{\text{cond}}$; it is equal to 1 if “cond” holds and is equal to 0 otherwise. The column vector of ones of length $N$ is denoted by $1_N$ and the matrix of ones of dimensions $N \times M$ is denoted by $1_{N \times M}$. The norm of a vector $\mathbf{x}$ of length $N$, denoted by $\|\mathbf{x}\|$, always refers to the $L_1$-norm, i.e., $\|\mathbf{x}\| = |x_1| + |x_2| + \ldots + |x_N|$, while the cardinality of a set $S$ is denoted by $|S|$. The probability given to strategy $s$ is denoted by $\alpha_s$.

We consider a strategic situation between a defender and an agent, that may either by an attacker with probability $p$ or a non-attacker, such as a normal user, with probability $1 - p$.

The defender seeks to classify the agent as class 0 (normal user) or class 1 (attacker). The strategic attacker exploits the uncertainty of the defender (about the type of the agent) by attacking in such a way to evade classification. For instance, in spam classification, the spammer might change the frequency of words included in an email to evade mail spam filters. In other settings, the owner of fraudulent twitter accounts might need to acquire more followers or publish more posts, so that he will get misclassified as a normal user [TMG+13, SWE+13].

The agent selects an “attack” vector $v$ in $\mathcal{V}$, where $\mathcal{V}$ is the set of all possible attack vectors. Note, that although we use the word “attack”, the vector of the attacker could be any feature upon which he will be classified: Each attack vector could be a set of features, like average number of friends, number of retweets, and others (in social networks fraud), the number of initiated connections (portscanner detection [JPBB04a]), a path on a graph among nodes in a network or number of accesses to different targets (intruder detection). Therefore, the setting we will describe is much broader than the security scenario presented
The results can be applied in any setting in which a defender needs to detect a malicious user, who is willing to evade classification by gaming.

The defender observes the attack behavior of the agent and selects a classification rule \( c \) in \( \mathcal{C} \), where \( \mathcal{C} \subseteq 2^{\mathcal{V}} \) is the set of all possible classification rules. We define the notion of the classification rule, called classifier, as follows.

**Definition 1 (Classifier).** A classifier \( c \) is a function \( c : \mathcal{V} \rightarrow \{0,1\} \), with

\[
c(v) = \begin{cases} 
0 & \text{for "attacker" classification} \\
1 & \text{for "non-attacker" classification.}
\end{cases}
\]

If the agent is an attacker, the choice \( v \) is strategic — the attacker seeks to maximize the payoff function

\[
U^A(v, c) = R(v) - c_d I_{c(v)=1}, \tag{2.1}
\]

where \( R : \mathcal{V} \rightarrow \mathbb{R}_+ \) is the reward function of attack vectors \( v \), and \( c_d \) is the cost in case of detection. We refer to \( R(v) \) as the “reward” (to the attacker) for the attack vector \( v \), which is granted to the attacker even in case of detection. In contrast, his “payoff” is the reward minus the cost if detected.

If the agent is a non-attacker, he picks \( v \in \mathcal{V} \) according to a distribution \( P_N(\cdot) \) over \( \mathcal{V} \), known to both players. \( \mathcal{V} \) is assumed to be finite. Since \( \mathcal{V} \) is finite by assumption, then \( \mathcal{C} \) is also a finite set.

The defender’s payoff has two additive components. The first component captures the expected loss to an attacker. We assume that this part equals what is gained by the

\footnote{For example in Twitter Spam, \( R(v) \) could be the cost of a fraudulent account with \( v \) being the number of followers.}
attacker. Then, since the defender interacts with an attacker with chance \( p \), the expected loss is \(-pU^A(v, c)\). The second component captures the expected loss due to false alarms. Since the non-attacker is present with chance \( 1 - p \), the expected false alarm cost is \( 1 - p \) times the chance that a non-attacker would pick a \( v \) that gets classified as an attacker. Finally, the whole payoff function is scaled by the constant \( 1/p \) for the convenience of having the term \( U^A(v, c) \) appear unscaled in the payoff. The resulting payoff function is

\[
U^D(v, c) = -U^A(v, c) - \frac{1-p}{p} c_{fa} \sum_{v' \in V} P_N(v') \mathbb{1}_{c(v')=1}, \tag{2.2}
\]

where \( c_{fa} \) is a constant that captures the cost of false alarms. Note that scaling the payoff of one player does not change the set of Nash equilibrium strategies. Moreover, the results presented below are still valid if the defender bears a different cost from each attack vector than the attacker, e.g., strategy \( v \) yields a reward \( R(v) \) to the attacker and a cost \( D(v) \neq R(v) \) to the defender. In that case, the payoff function of the defender is

\[
\hat{U}^D(v, c) = U^D(v, c) + R(v) - D(v)
\]

and adding constants to the rows of the column player’s (defender) payoff matrix produces the same best responses: given any \( v \in V \), maximizing \( \hat{U}^D \) is equivalent to maximizing \( U^D \). For simplicity of the exposition, we assume that the reward/cost function of the attack vector is the same for both players. We summarize the game we consider in the following definition.

**Definition 2** (Classification game \( \mathcal{G} \)). The classification game \( \mathcal{G} = (V, C, p, c_d, c_{fa}, P_N) \) is the game with two players, the attacker and the defender, with

- \( V \) : the strategy space of the attacker;
- \( C \subseteq 2^V \) : the strategy space of the defender;
• \( p \in [0,1] \): probability that the agent is an attacker;

• \( c_d \in \mathbb{R}_+ \): cost of detection;

• \( c_{fa} \in \mathbb{R}_+ \): cost of false alarm;

• \( P_N : \mathcal{V} \rightarrow [0,1] \): probability measure that describes the non-attacker’s distribution on \( \mathcal{V} \);

and objectives given by (2.1) and (2.2).

2.2 Justification of threshold strategies

In this section, we show that in equilibrium, the defender’s strategy space can be reduced to threshold classifiers on the attacker reward. We are interested in mixed strategy Nash equilibria — equilibria in which the attacker randomizes across multiple attack vectors with a probability distribution \( \alpha \) on \( \mathcal{V} \) and the defender randomizes across multiple classifiers with a probability distribution \( \beta \) on \( \mathcal{C} \). A pure strategy is a special case of mixed strategies in which that particular pure strategy is selected with probability 1 and every other strategy with probability 0.\(^2\) The expected attacker and defender payoffs are then given by

\[
U_A(\alpha, \beta) = \sum_{v \in \mathcal{V}} \sum_{c \in \mathcal{C}} \alpha_v U^A(v, c) \beta_c, \tag{2.3}
\]

\[
U_D(\alpha, \beta) = \sum_{v \in \mathcal{V}} \sum_{c \in \mathcal{C}} \alpha_v U^D(v, c) \beta_c. \tag{2.4}
\]

\(^2\)For most instances of interest no pure-strategy equilibrium exists. If one player chooses deterministically a pure strategy, the opponent would switch to a strategy to either evade detection completely or to guarantee detecting the attacker.
Recall the definition of Nash equilibrium [FT91]:

**Definition 3** (Nash equilibrium). The pair of probability measures \((\alpha, \beta)\) on \(V\) and \(C\) respectively is a Nash equilibrium (NE) of game \(G\) if each player’s mixed strategy is a best response to the other player’s mixed strategy, i.e,

\[
U^A(\alpha, \beta) \geq U^A(\hat{\alpha}, \beta), \quad (2.5)
\]

for every probability distribution \(\hat{\alpha}\) over \(V\), and

\[
U^D(\alpha, \beta) \geq U^D(\alpha, \hat{\beta}) \quad (2.6)
\]

for every probability distribution \(\hat{\beta}\) over \(C\).

We define the notion of best-response equivalent games in the same way as in [Ros74]:

**Definition 4.** Two games are **best-response equivalent** if the sets of best response strategies of a player in both games coincide for any strategy of the other player.

Note that for best-response equivalent games, the strategy spaces need to be the same for both players. We now define the reduced strategy space for the defender that consists of threshold rules on all possible attack rewards.

**Definition 5** (Set of threshold classifiers).

\[
C^T = \{ c \in C : c(v) = 1_{R(v) \geq t}, \forall v \in V \text{ for some } t \in \mathbb{R} \}.
\]

We also define the probability of detection function, which corresponds to the probability that the attacker is detected given his chosen attack vector \(v\) and the defender’s strategy \(\beta\).
Definition 6 (Probability of detection function). The probability of detection for an attack vector $v$ and defender’s strategy $\beta$ is defined as

$$\pi_d^\beta(v) = \sum_{c \in C} \beta_c 1_{c(v) = 1}, \quad \forall v \in V.$$ (2.7)

2.2.1 Defender’s reduced strategy space

Threshold strategies are simple and intuitive, but their optimality is not guaranteed a priori. In this section, we show that it is optimal for the defender to use threshold strategies in the classification game. This result is formally stated in the following theorem.

Theorem 1. For any NE $(\alpha, \beta)$ of $G = (V, C, p, c_d, c_{fa}, P_N)$, there exists a NE of $G^T = (V, C^T, p, c_d, c_{fa}, P_N)$ with the same $\alpha$ and equilibrium payoff pair and same $\pi_d$ in the support of the non-attacker’s distribution.

In other words, the defender compares what the attack reward would have been from the observed attack vector to a threshold instead of computing a mapping from any possible combination of attack vectors to a detection probability.

The proof of Theorem 1 is provided at the end of this section. We establish a series of lemmas that give information about the game structure.

We first show that both players’ expected payoffs depend on the defender’s strategy $\beta$ (probability measure among classifiers) only through the probability of detection function of each attack vector $v$.

Lemma 1. For any strategy profile $(\alpha, \beta)$ of $G = (V, C, p, c_d, c_{fa}, P_N)$, the expected payoffs of the players depend on $\beta$ only through the probability of detection function $\pi_d^\beta(\cdot)$:
\[ U^A(\alpha, \beta) = \sum_{v \in V} \left( \alpha_v R(v) - c_d \alpha_v \pi_d^\beta(v) \right), \quad (2.8) \]

\[ U^D(\alpha, \beta) = -U^A(\alpha, \beta) - \frac{1-p}{p} cf_o \sum_{v' \in V} \left( P_N(v') \pi_d^\beta(v') \right). \quad (2.9) \]

**Proof.** For the attacker, from (2.3) we derive

\[ U^A(\alpha, \beta) = \sum_{v \in V} \sum_{c \in C} \alpha_v U^A(v, c) \beta_c \]

\[ = \sum_{v \in V} \sum_{c \in C} \alpha_v \left( R(v) - c_d \sum_{c \in C} \beta_c \mathbb{1}_{c(v)=1} \right) \beta_c \]

\[ = \sum_{v \in V} \left( \alpha_v R(v) - c_d \sum_{c \in C} \beta_c \mathbb{1}_{c(v)=1} \right) \]

\[ = \sum_{v \in V} \left( \alpha_v R(v) - c_d \alpha_v \pi_d^\beta(v) \right), \]

where \( \pi_d^\beta(v) \) is given by (2.7). Similarly, for the defender, (2.4) yields

\[ U^D(\alpha, \beta) = \sum_{v \in V} \sum_{c \in C} \alpha_v U^D(v, c) \beta_c \]

\[ = -\sum_{v \in V} \sum_{c \in C} \alpha_v U^A(v, c) \beta_c \]

\[ - \sum_{c \in C} \beta_c \sum_{v' \in V} \frac{1-p}{p} c_{fa} P_N(v') \mathbb{1}_{c(v')=1} \]

\[ = -U^A(\alpha, \beta) - \sum_{v' \in V} \frac{1-p}{p} c_{fa} P_N(v') \pi_d^\beta(v'). \]

By abuse of notation, we denote the probability of detection function by \( \pi_d \), instead of \( \pi_d^\beta \), when it brings no ambiguity.
Lemma 2. For any function $f : V \rightarrow [0, 1]$, there exists a probability measure $\beta$ over $C = 2^{[V]}$ s.t. $f(v) = \pi^\beta_d(v), \forall v \in V$.

Proof. Let $f$ be an arbitrary function from $V \rightarrow [0, 1]$. Without loss of generality, we reindex strategies $v$ such that $f$ is non-decreasing, i.e., $\forall v_i, v_j \in V$, with $i < j$, $f(v_i) \leq f(v_j)$. Starting with the attack vector $v_1$ with the lowest value of $f$, we assign positive probability $\beta_{c_1} = \pi^\beta_d(v_1)$ to the classifier $c_1 \in C$ with $c_1(v) = 1, \forall v \in V$. We then assign $\beta_{c_2} = f(v_2) - f(v_1)$ to the classifier $c_2$ with $c_2(v) = 1, \forall v \in V \setminus \{v_1\}$. We continue this process until we reach the last vector $v_{|V|}$. We assign probability $f(v_{|V|}) - f(v_{|V| - 1})$ to the classifier that classifies only $v_{|V|}$ as coming from an attacker. The remaining weight $1 - f(v_{|V|})$ (if positive) is given to the classifier that never classifies the agent as an attacker. The strategy $\beta$ derived with the above procedure is guaranteed by construction to have elements in $[0, 1]$ with unit sum. Moreover $\pi^\beta_d(v_1) = \beta_{c_1} = f(v_1)$ since $v_1$ is classified as coming from an attacker only by $c_1$, $\pi^\beta_d(v_2) = \beta_{c_1} + \beta_{c_2} = f(v_2)$ since $v_2$ is classified as coming from an attacker only by $c_1$ and $c_2$ and so on until $\pi^\beta_d(v_{|V|}) = \sum_{i=1}^{[V]} \beta_{c_i} = f(v_{|V|})$, in which case $v_{|V|}$ is classified as coming from an attacker by all classifiers $c_1$ through $c_{v_{|V|}}$. Thus we have constructed a valid probability measure $\beta$ over $C$, with $\pi^\beta_d(v) = f(v), \forall v \in V$.

Without loss of generality, we now rank the attack vectors in non-decreasing attacker reward, i.e., $R(v_i) \leq R(v_{i+1}), \forall i \in \{1, \ldots, |V| - 1\}$. The following lemma establishes that under some assumptions about the non-attacker’s distribution, in any NE: (1) the defender is never selecting a threshold on a specific reward if the attacker’s strategy never yields that specific reward, and (2) smaller attacker rewards have a higher probability to never be selected in either players’ support.
Lemma 3. If $\left(\alpha, \beta\right)$ is a NE of $G = \left(V, C, p, c_d, c_{fa}, P_N\right)$ that yields a probability of detection function $\pi_d$, then $\forall v \in V$ such that $P_N(v) > 0$, there exist at most three possible cases for the players’ strategies, as depicted in Figure 2.1:

- **Case 0:** $\alpha_v = 0$, $\pi_d(v) = 0$,
- **Case 1:** $\alpha_v > 0$, $\pi_d(v) = 0$, and
- **Case 2:** $\alpha_v > 0$, $\pi_d(v) > 0$.

Furthermore $R(v_0) \leq R(v_1) < R(v_2)$, for strategies $v_0, v_1, v_2$ in Cases 0, 1, and 2 respectively.

Proof. Suppose that $\left(\alpha, \beta\right)$ is a NE and that there exists $v^* \in V$ with $P_N(v^*) > 0$ such that $\alpha_{v^*} = 0$ and $\frac{\beta_d}{\pi_d}(v^*) > 0$. Let $\hat{\beta}$ be a mixed strategy of the defender assigning zero detection probability on $v^*$ and leaving the probability of detection unchanged for other attack vectors, i.e., such that $\frac{\hat{\beta}_d}{\pi_d}(v^*) = 0$ and $\frac{\hat{\beta}_d}{\pi_d}(v) = \frac{\beta_d}{\pi_d}(v)$, for all $v \neq v^*$. By Lemma 2, strategy $\hat{\beta}$ exists. By Lemma 1, we have

$$U^D(\alpha, \hat{\beta}) = U^D(\alpha, \beta) + \frac{1-p}{p} c_{fa} P_N(v^*) \frac{\beta_d}{\pi_d}(v^*) > U^D(\alpha, \beta),$$

which contradicts the fact that $\left(\alpha, \beta\right)$ is a NE.
We now show that $R(v_0) \leq R(v_1) < R(v_2), \forall v_0, v_1, v_2$ in Cases 0, 1, and 2 respectively. Since both pure strategies $v_1, v_2$ are included in the attacker’s equilibrium mixed strategy, $U^A(v_1, \beta) = U^A(v_2, \beta)$. Since $\pi_d(v_1) = 0$ and $\pi_d(v_2) > 0$,

$$R(v_1) - c_d \cdot 0 = R(v_2) - c_d \pi_d(v_2) \implies R(v_1) < R(v_2).$$

Moreover, since $\alpha_{v_0} = 0, \alpha_{v_1} > 0$, $U^A(v_0, \beta) \leq U^A(v_1, \beta)$. Since $\pi_d(v_0) = \pi_d(v_1) = 0$, this implies $R(v_0) - c_d \cdot 0 \leq R(v_1) - c_d \cdot 0$, hence $R(v_0) \leq R(v_1)$. 

We thus expect that in NE, the defender will randomize among classifiers that detect the most rewarding attack vectors, and the attacker will randomize among the most rewarding attack vectors. Finally, if the attacker does not use a certain attack vector in NE, the defender never classifies the agent as attacker upon seeing this vector. This is illustrated in Section 2.3 using numerical experiments. We can also show the following corollary.

**Corollary 1.** If $(\alpha, \beta)$ is a NE of $\mathcal{G} = (V, C, p, c_d, c_{fa}, P_N)$, for all $v_i, v_j$ in Case 1, $R(v_i) = R(v_j)$.

**Proof.** Suppose that there exist $v_1, v_2$ in Case 1, such that $R(v_1) \neq R(v_2)$. Since $v_1, v_2$ are in Case 1, $\alpha_{v_1} > 0, \alpha_{v_2} > 0$ and

$$\pi_d(v_1) = \pi_d(v_2) = 0.$$ 

(2.10)

Since the attacker mixes among both pure strategies $v_1, v_2$, these give the same expected
utility to the attacker:

\[ U^A(v_1, \beta) = U^A(v_2, \beta) \]

\[ \Rightarrow R(v_1) - c_d \cdot 0 = R(v_2) - c_d \cdot 0 \quad \text{(using (2.10))} \]

\[ \Rightarrow R(v_1) = R(v_2). \]

Contradiction, since we assumed that \( R(v_1) \neq R(v_2) \).

What we show next is that, under certain assumptions on the non-attacker’s behavior, vectors of higher attacker reward have higher or equal probability of getting detected than vectors with smaller reward.

**Lemma 4.** Suppose that \( P_N(v) > 0, \forall v \in V \). In any NE \((\alpha, \beta)\) of \( \mathcal{G} = (V, C, p, c_d, c_{fa}, P_N) \), the probability of detection function \( \pi_d(v) \) is non-decreasing in the attack reward \( R(v) \), i.e.,

\[ \forall v_1, v_2 \in V, R(v_1) \leq R(v_2) \Rightarrow \pi_d(v_1) \leq \pi_d(v_2). \]

**Proof.** If both \( v_1, v_2 \) are in either Cases 0 or 1, then by Lemma 3 (which holds since since \( P_N(v) > 0 \)), \( \pi_d(v_1) = \pi_d(v_2) = 0 \) so that \( \pi_d(v_1) \leq \pi_d(v_2) \) holds. Similarly there is no violation if \( v_1, v_2 \) are in Cases 0 or 1 and 2 respectively, since \( \pi_d(v_1) = 0 \) and \( \pi_d(v_2) > 0 \).

The only remaining case is when both \( v_1 \) and \( v_2 \) are in Case 2. Suppose that \( \pi_d \) is decreasing in the attack reward in Case 2, i.e., there exist \( v_1, v_2 \) in Case 2, with \( R(v_1) \leq R(v_2) \), \( P_N(v_1) > 0, P_N(v_2) > 0 \), s.t. \( \pi_d(v_1) > \pi_d(v_2) \). From Lemma 3, \( \alpha_{v_1} > 0, \alpha_{v_2} > 0 \).

Since the attacker mixes among both \( v_1, v_2 \), \( U^A(v_1, \beta) = U^A(v_2, \beta) \). But, from (2.8) we have

\[ U^A(v_2, \beta) = U^A(v_1, \beta) + R(v_2) - R(v_1) + c_d (\pi_d(v_1) - \pi_d(v_2)), \]
and since $R(v_2) \geq R(v_1)$ and $\pi_d(v_1) > \pi_d(v_2)$ we get $U^A(v_2, \beta) > U^A(v_1, \beta)$. Contradiction.

The first step in the proof of Lemma 2 was to reindex attack vectors so that they have non-decreasing detection probability. By Lemma 4, vectors ranked in non-decreasing reward already satisfy this property. We can thus skip the step of reindexing and describe the nature of classifiers $c \in \mathcal{C}$ that are given positive weight. Classifier $c_1$ detects all attack vectors and is equivalent to a threshold classifier on the reward of the vector with the smallest detection probability or the least reward that can be achieved. Classifier $c_2$ detects all vectors except the one with the smallest detection probability and is equivalent to a threshold classifier on the second smallest attack reward, and so on until we reach classifier $c_{|\mathcal{V}|}$ that detects all attack vectors (threshold on the highest reward that can be achieved).

The remaining weight (if any) is given to classifier $c_{|\mathcal{V}|+1}$ that always classifies the agent as a non-attacker, which is equivalent to a threshold on the highest attacker reward. The above procedure leads to the following corollary of Lemma 2:

**Corollary 2.** For any NE $(\alpha, \beta)$ of $G = (\mathcal{V}, \mathcal{C}, p, c_d, c_{fa}, P_N)$ that results in a non-decreasing probability of detection $\pi_d^\beta$ there exists a NE $(\alpha, \hat{\beta})$ of $G$, where $\hat{\beta}_c = 0, \forall c \in \mathcal{C} - \mathcal{C}^T$ and $\pi^\beta_d(v) = \pi^\beta_d(v)$, $\forall v \in \mathcal{V}$.

We now formalize the proof for Theorem 1.

**Proof of Theorem 1.** Suppose that $(\alpha, \beta)$ is a NE of $G = (\mathcal{V}, \mathcal{C}, p, c_d, c_{fa}, P_N)$, that results in some $\pi_d^\beta$. We consider two cases.

1. $P_N(v) > 0$, for all $v \in \mathcal{V}$: By Lemma 4, $\pi_d^\beta$ is non-decreasing and Corollary 2 holds. Thus
(\(\alpha, \hat{\beta}\)) is a NE of \(\mathcal{G}\) where \(\hat{\beta}\) is some probability measure over \(\mathcal{C}\) with positive weight only on \(\mathcal{C}^T\). Therefore \((\alpha, \beta^T)\) is also NE of \(\mathcal{G}^T = (\mathcal{V}, \mathcal{C}^T, p, c_d, c_{fa}, P_N)\), where \(\beta^T\) is a probability measure over \(\mathcal{C}^T\) with \(\beta^T_c = \hat{\beta}_c\), for all \(c \in \mathcal{C}^T\) and \(\pi^T_d(v) = \pi^T_d(\hat{\beta}(v))\), for all \(v \in \mathcal{V}\).

2. There exists \(v^* \in \mathcal{V}\) with \(P_N(v^*) = 0\). We first show that if \(\alpha_{v^*} > 0\), \(\pi_d(v^*) = 1\). Suppose \(\pi_d(v^*) < 1\). Consider \(\hat{\beta}\) that results in \(\pi^T_d(v^*) = 1\) and \(\pi^T_d(v) = \pi^T_d(\beta(v))\), for all \(v \neq v^*\). By Lemma 2, such \(\pi^T_d\) exists. By Lemma 1,

\[
U^D(\alpha, \hat{\beta}) = U^D(\alpha, \beta) + c_d \left(1 - \pi^T_d(v^*)\right)
\]

\[
> U^D(\alpha, \beta),
\]

which contradicts the fact that \((\alpha, \beta)\) is a NE. Let us further distinguish the following subcases:

(a) There exists \(\hat{v} \in \mathcal{V}\), with \(R(\hat{v}) > R(v^*)\): We show by contradiction that \(\alpha_{v^*} = 0\).

Suppose that \(\alpha_{v^*} > 0\). By the previous analysis, \(\pi_d(v^*) = 1\). By Lemma 1 we have

\[
U^A(\hat{v}, \beta) = U^A(v^*, \beta) + R(\hat{v}) - R(v^*) + c_d \cdot (1 - \pi_d(\hat{v}))
\]

\[
> U^A(v^*, \beta),
\]

since \(R(\hat{v}) > R(v^*)\) and \(\pi_d(\hat{v}) \leq 1\). Contradiction. The defender assigns some positive weight to the threshold classifier \(c^*\), where \(c^*(v) = 1\) if \(R(v) \geq R(v^*)\) and 0 otherwise. If \(\beta_{c^*} = 0\) then the attacker would switch some weight to \(v^*\) to avoid detection. Consequently \(\pi_d(v^*) > 0\), such that \(\pi_d(v^*) = \pi_d(v^*_{next})\), where \(v^*_{next}\) is the next rewarding strategy after \(v^*\) when these are sorted in increasing reward. Otherwise, if \(\pi_d(v^*) < \pi_d(v^*_{next})\) the Nash equilibrium assumption would be violated.
Figure 2.2: The attacker’s reduced strategy space is the set of all images of function $R$.

(the attacker would unilaterally deviate to the strategy with the smallest probability of detection). Thus the result of the theorem is not contradicted.

(b) For all $v \in \mathcal{V}, R(v^*) > R(v)$. If $\alpha_{v^*} > 0$, by the previous analysis $\pi_d(v^*) = 1 \geq \pi_d(v), \forall v \in \mathcal{V}$. The probability of detection function is non-decreasing and Corollary 2 holds, yielding a distribution $\beta^T$. If $\alpha_{v^*} = 0$, we can construct a Nash equilibrium with resulting $\pi_d(v^*) = 1$ (in a similar way as in 2a but now there is no “next” rewarding strategy, hence $\beta_{c^*}$ is selected such as $\pi_d(v^*) = 1$).

In both cases, $\alpha$ remains the same in $G^T$. By construction $\beta^T$ leads to the same $\pi_d$ as $\beta$ and by Lemma 1 to the same players expected payoffs. □

2.2.2 Reduced attacker’s strategy space and equilibrium structure

In this section we look into the attacker’s strategy in equilibrium. We also show that we can reduce the attacker’s NE strategy space to strategies that yield distinct attack reward. In the following lemma, we show that the attacker mixes among certain strategies $v \in \mathcal{V}$ proportionally to the non-attacker’s distribution.

**Lemma 5.** If $(\alpha, \beta)$ is a NE of $\mathcal{G} = (\mathcal{V}, \mathcal{C}, p, c_d, c_f, P_N)$, then for all $v \in \mathcal{V}$ such that
0 < \pi_d(v) < 1,
\alpha_v = \frac{1 - p c_{fa}}{p c_d} P_N(v). \quad (2.11)

\textbf{Proof.} Consider } v_i \in \mathcal{V} \text{ with } \pi_d(v_i) \in (0, 1). \text{ Since } \pi_d(v_i) \neq 0, \text{ there exists } c_i \in \mathcal{C} \text{ s.t. } c_i(v_i) = 1 \text{ with } \beta_{c_i} > 0. \text{ Since } \pi_d(v_i) \neq 1, \text{ there exists } c^* \in \mathcal{C} \text{ s.t. } c^*(v) = 0 \text{ with } \beta_{c^*} > 0. \text{ Without loss of generality suppose that } c_i(v) = 0, \forall v \in \mathcal{V} - \{v_i\} \text{ and } c^*(v_i) = 0, \forall v \in \mathcal{V}. \text{ Since } \beta_{c_i} > 0, \beta_{c^*} > 0, U^D(\alpha, c_i) = U^D(\alpha, c^*), \text{ which yields }
\sum_{v \in \mathcal{V}} \alpha_v R(v) + c_d \alpha_v - \frac{1 - p c_{fa}}{p} P_N(v) = - \sum_{v \in \mathcal{V}} \alpha_v R(v),
\text{ which immediately gives the result.} \quad \square

Thus, for attack vectors with some uncertainty of getting detected, the attacker mixes proportionally to the non-attacker’s distribution.

The attacker’s strategy space could be complex. For instance, consider a game in which there are M different targets to attack and the attacker generates attack sequences on these targets of length N. Counting the no attack strategy, there are (M + 1)^N permutations of attack sequences. Given that the defender’s classification rule is a threshold on the attack reward, permutations of attacks that yield the same attack reward will have the same probability of getting detected, and can be collapsed into a single strategy. For instance, if only the total number of times each target is hit (and not the sequence of the attacks) matters for the reward of the attacker, there are \binom{N + M + 1}{N} combinations of attack rewards. Hence, we can exploit the defender’s equilibrium threshold rule and reduce the cardinality (and complexity) of the attacker’s strategy space. As N or M increase, the benefits from this reduction become more profound.
The intuition behind the reduction in the attacker’s strategy space comes from the following observation: If the attacker includes in his equilibrium support one attack vector of a certain reward, he should include all vectors of the same reward as all of them will have the same probability of detection. The attacker’s equilibrium weight on each one is proportional to the non-attacker’s distribution (see Lemma 5). Since the defender’s classification is based on the reward of the attack vectors (and not on the actual vector), a game in which the attacker mixes on attack rewards (instead of attack vectors) does not influence the defender’s equilibrium strategy. Furthermore, such a game does not give any more or less freedom to the attacker, but reduces the complexity (cardinality) of the strategy space of the attacker.

Essentially, we can think of the new attacker’s strategy space as the set of all images of the reward function $R : \mathcal{V} \to \mathbb{R}_+$, as depicted in Figure 2.2, and defined as follows.

**Definition 7 (Reduced strategy space of the attacker).**

$$\mathcal{V}^R = \{ r \in \mathbb{R}_+ : r = R(v), \text{ for some } v \in \mathcal{V} \}.$$  \hfill (2.12)

The non-attacker’s probability measure can be similarly reduced to a probability measure that describes the non-attacker’s distribution on $\mathcal{V}^R$ with

$$P^R_N(r) = \sum_{v' \in \mathcal{V}} P_N(v') \mathbb{1}_{R(v')=r}.$$  \hfill (2.13)

We show the following proposition that establishes the intuition described above.

**Proposition 1.** If $(\alpha, \beta)$ is a NE of $G^T = (\mathcal{V}, \mathcal{C}^T, p, c_d, c_{fa}, P_N)$, then $(\alpha^*, \beta)$ is a NE of
\( G^{R,T} = (V^R, C^T, p, c_d, c_f_a, P^R_N) \) with the same equilibrium payoff pair where

\[
\alpha^*_r = \sum_{v_i \in V, R(v_i) = r_i} \alpha_{v_i}, \forall r_i \in V^R.
\]

**Proof.** Suppose \((\alpha, \beta)\) is a NE of \(G^T\). If \(|V| = |V^R|\), all attack vectors in \(V\) yield distinct rewards and \(\alpha^*_r = \alpha_{v_i}, \forall r_i \in V^R\), with \(r_i = R(v_i)\).

Suppose that there exist \(v_1, v_2 \in V\) with \(v_1 \neq v_2\) and \(r = R(v_1) = R(v_2)\) and all other vectors yield a distinct reward\(^3\). We show that if we collapse vectors \(v_1, v_2\) to a single strategy \(r\) with \(\alpha^*_r = \alpha_{v_1} + \alpha_{v_2}\), we get the same expected utilities for the two players. Indeed,

\[
U^A(\alpha, \beta) = \sum_{i=3}^{\lvert V \rvert} \left( \alpha_{v_i} R(v_i) - c_d \alpha_{v_i} \pi_d(v_i) \right)
\]

\[
+ \sum_{i=1}^{2} \left( \alpha_{v_i} R(v_i) - c_d \alpha_{v_i} \pi_d(v_i) \right).
\]

(2.14)

The defender classifies each attack vector according to a threshold \(t\) on the reward, thus

\[
1_{c(v_1)} = 1_{c(v_2)} = 1_{r \geq t}, \forall c \in C^T, \text{ and } \pi_d(v_1) = \pi_d(v_2) = \pi_d(r).\]

Thus,

\[
\sum_{i=1}^{2} \left( \alpha_{v_i} R(v_i) - c_d \alpha_{v_i} \pi_d(v_i) \right) = \sum_{i=1}^{2} \left( \alpha_{v_i} R(v) - c_d \alpha_{v_i} \pi_d(v) \right)
\]

\[
= \alpha^*_r r - c_d \alpha^*_r \pi_d(r),
\]

(2.15)

where \(\alpha^*_r = \alpha_{v_1} + \alpha_{v_2}\). Substituting the second term in (2.14) from (2.15) yields the same equilibrium attacker’s utility and \(U^A(\alpha, \beta) = U^A(\alpha^*, \beta)\). Similarly we show that the defender’s equilibrium utility is the same if we collapse equally rewarding attack vectors into a single strategy.

\(^3\)We can generalize this procedure for more than two equally rewarding vectors and for multiple groups of distinct rewards.
It is easier to compute \((\alpha^*, \beta)\) in \(G^{R,T}\), in which the cost matrix of the attacker consists of non-identical rows. By Proposition 1, \(\beta\) in \(G^T\) is the same as in \(G^{R,T}\). Theorem 2 in Section 2.3 gives \((\alpha^*, \beta)\). Given \(\alpha^*\) on \(\mathcal{V}^R\) we compute the attacker’s strategy \(\alpha\) over \(\mathcal{V}\), as follows: For \(r_i \in \mathcal{V}^R\) with \(\pi_d(r_i) \in (0,1)\), \(\alpha_{v_i}\) is given by (2.11) \(\forall v_i \in \mathcal{V}\) with \(R(v_i) = r\). For \(r_i\) with \(\pi_d(r_i) \in \{0,1\}\), the attacker is not restricted in the scaled version of the non attacker’s distribution. Any possible combination of weights is possible, as long as \(\sum_{v_i \in \mathcal{V}, R(v_i) = r_i} \alpha_{v_i} = \alpha^*_{r_i}\).

Therefore, although we reduce the attacker’s strategy space, we provide a roadmap to get all NE of \(G^T\) as well. Note that if all attack strategies in \(\mathcal{V}\) yield distinct attack reward, then \(|\mathcal{V}| = |\mathcal{V}^R|\) and there is no reduction in the attacker’s strategy space.

### 2.3 Nash Equilibrium Analysis

Our goal in this section is to characterize the structure of the NE of the classification game. It is known that every finite game (finite number of players with finite number of strategies for each player) has a mixed-strategy NE [Nas50]. Our game is finite, thus it admits a NE in mixed strategies. However, finding the Nash equilibrium has a high computational complexity in the general case [CDT09].

We consider the game \(G^{R,T} = (\mathcal{V}^R, \mathcal{C}^T, p, c_d, c_f, P^R_N)\), in which the attacker’s strategy space consists of distinct attack rewards \(r \in \mathcal{V}^R\), the defender’s strategy space consists of threshold classifiers \(c \in \mathcal{C}^T\), and \(P^R_N\) is the non-attacker’s probability measure on \(\mathcal{V}^R\) given by (2.13). We denote by \(r_i, i \in \{1, \ldots, |\mathcal{V}^R|\}\) the elements of \(\mathcal{V}^R\) and, without
loss of generality, we assume that they are ranked in increasing order, i.e., $r_i < r_{i+1}$ for all $i \in \{1, \ldots, |\mathcal{V}^R| - 1\}$. Similarly, classifier $c_i$ corresponds to a threshold classifier with threshold equal to the attacker reward $r_i$. By definition, $C^T$ includes the “always classify as non-attacker” strategy, which is denoted by $c_{|\mathcal{V}^R|+1}$, thus $|C^T| = |\mathcal{V}^R| + 1$. The last classifier is a threshold on the reward $r_{|\mathcal{V}|} + \delta$.

We can express the payoff functions of the players in compact form as matrix equations. We define $\tilde{\Lambda}$ to be the cost matrix of the attacker, with $\tilde{\Lambda}(i, j) = c_d I_{r_i \geq r_j} - r_i$:

$$
\tilde{\Lambda} = c_d \begin{pmatrix}
1 & 0 & \cdots & \cdots & 0 \\
\vdots & 1 & \ddots & \ddots & \vdots \\
\vdots & \ddots & \ddots & \ddots & \vdots \\
\vdots & \ddots & \ddots & 0 & \vdots \\
1 & \cdots & \cdots & 1 & 0
\end{pmatrix} - \begin{pmatrix}
r_1 \\
r_2 \\
\vdots \\
r_{|\mathcal{V}^R|-1} \\
r_{|\mathcal{V}|}
\end{pmatrix} \cdot 1'_{|\mathcal{V}^R|+1}.
$$

Hence, there are $|\mathcal{V}^R|$ rows (strategies) for the attacker, all ranked in increasing reward. Certain computations are simplified by using a matrix with only positive entries. Therefore, we define $\Lambda = \tilde{\Lambda} + (r_{|\mathcal{V}^R|} + \epsilon) \cdot 1'_{|\mathcal{V}^R| \times |\mathcal{V}^R|+1}$, where $\epsilon > 0$. Adding a constant to the players payoff does not affect their best responses, and hence does not change the equilibrium strategies. Thus, from here on, we will use matrix $\Lambda$ to define the players’ payoffs.

In the following, the pair $(\alpha, \beta)$ will denote the probability measures of the attacker and defender on $\mathcal{V}^R, C^T$ respectively. The attacker’s expected cost is given by $\alpha'\Lambda\beta$. The

\footnote{We have assumed detection when $r \geq t$, thus the “always classify as attacker” strategy is associated with a threshold on the minimum reward ($r_1$) that can be achieved. For symmetry, we also allow the “always classify as non-attacker” strategy which would be associated with a threshold on the reward $r_{|\mathcal{V}^R|} + \delta$, $\delta > 0$.}
defender’s expected payoff is given by $\alpha'\Lambda^\text{eq}\beta$, with

$$\Lambda^\text{eq} = \Lambda - 1_{|\mathcal{V}_R|} \cdot \mu',$$

(2.16)

where $\mu$ represents the false alarm penalty vector for the defender with elements $\mu_i = 1 - \frac{p}{c_{\text{fa}}} \sum_{k \geq i} P_N^R(r_k)$. We assume that $\mu$ is a strictly decreasing vector (component-wise): $\mu_i > \mu_{i+1}, \forall i \in \{1, \ldots, |\mathcal{V}_R|\}$, i.e., $P_N^R(r_i) > 0, \forall i \in \{1, \ldots, |\mathcal{V}_R|\}$. This assumption is equivalent to assuming that the non-attacker gives some positive weight to all strategies $r \in \mathcal{V}_R$. Even if this property does not hold, we can still describe how both players behave, as shown in Theorem 1 in Section 2.2.

It is easy to show that $G^{R,T} = (\mathcal{V}_R, \mathcal{C}_T, p, c_d, c_{\text{fa}}, P_N^R)$ is best-response equivalent to a zero-sum game, in which the attacker’s cost is given by $\alpha'\Lambda^\text{eq}\beta$: The two games have the same players with the same strategy spaces. The vector $\mu$ depends only on the non-attacker’s complementary cumulative probability distribution and is fixed. Adding constants to the columns of the cost matrix of the attacker (row player) in the original classification game yields the cost matrix of the defender of the new game without changing the Nash equilibria of the game. Indeed, the defender’s payoff matrix is unchanged, and, for any given $\beta$, minimizing $\alpha'\Lambda\beta$ and minimizing $\alpha'\Lambda^\text{eq}\beta$ with $\Lambda^\text{eq}(i,j) = \Lambda(i,j) - \mu_j$ give the same minimizing strategy for the attacker. Thus the two games generate the same sets of best response functions hence they are best-response equivalent (see Definition 4) and have the same equilibrium strategies.

Note that the best-response equivalence of our game to a zero-sum game guarantees that in all NE the defender’s expected payoff will be the same (and equal to the value of the zero-sum game), but the attacker’s payoff is not always the same in each equilibrium of
the original nonzero-sum game. Indeed, equilibria of our original game could give different payoffs to the attacker after transforming his cost matrix \( \Lambda^{\text{eq}} \) (adding constants to the columns) back to \( \Lambda \).

There exist known polynomial algorithms to compute the Nash equilibria in zero-sum games via a transformation to a linear program [Kar84]. These algorithms, though, do not provide structure on the selected equilibria strategies. In the remaining of this section, we aim to give more intuition on the players’ NE strategies than the solution derived via a linear programming toolbox. Our results can be summarized in the following theorem.

**Theorem 2.** Algorithm 1 finds all NE of the classification game \( G_{R,T} \). Moreover, if \((\alpha, \beta)\) is a NE, then, there exists \( k \in \{1, \ldots, |V_R^l|\} \) such that

\[
\beta = (0, \ldots, 0, \beta_k, \ldots, \beta_{|V_R^l|}, \beta_{|V_R|+1}),
\]

\[
\alpha = (0, \ldots, 0, \alpha_k, \ldots, \alpha_{|V_R^l|}),
\]

where

\[
\beta_i = \frac{r_i - r_{i-1}}{c_d}, \quad \forall i \in \{k+1, \ldots, |V_R^l|\}, \quad (2.17)
\]

\[
\alpha_i = \frac{1 - p c_f a_c}{p} P_N(r_i), \quad \forall i \in \{k+1, \ldots, |V_R^l| - 1\}, \quad (2.18)
\]

and \( \beta_k, \beta_{|V_R^l|+1} \geq 0 \) and \( \alpha_k, \alpha_{|V_R^l|} \geq 0 \) are such that

(i) \( \beta_k > 0, \beta_{|V_R^l|+1} = 0 \), and

\( \alpha_k \) satisfies (2.18), \( \alpha_{|V_R^l|} = 1 - \sum_{i=k}^{|V_R^l|-1} \alpha_i \); or

(ii) \( \beta_k = 0, \beta_{|V_R^l|+1} > 0 \), and

\( \alpha_k = 1 - \sum_{i=k+1}^{|V_R^l|} \alpha_i \), \( \alpha_{|V_R^l|} \) satisfies (2.18); or
(iii) \( \beta_k = 0, \beta_{|V_R|+1} = 0, \) and
\[
\alpha_k \in [0, 1 - \sum_{i=k+1}^{|V_R|-1} \alpha_i], \quad \alpha_{|V_R|} = 1 - \sum_{i=k}^{|V_R|-1} \alpha_i; \quad \text{or}
\]

(iv) \( \beta_k \in [0, \frac{r_k - r_{k-1}}{c_d}], \beta_{|V_R|+1} = 1 - \sum_{i=k}^{|V_R|} \beta_i, \) and
\[
\alpha_k, \alpha_{|V_R|} \text{ satisfy (2.18)}.
\]

**Remark 1.** For those familiar with Linear Programming, the result in Theorem 2 is in accordance with the relationship between degeneracy and multiplicity of the primal and the dual optimal solutions \cite{Lue03}. We refer to a degenerate optimal solution when more than \( n \) constraints are tight at an extreme point of size \( n \). Consider the defender’s LP to be the primal LP. If there exists one non degenerate optimal solution, the optimal solution to the dual problem (the attacker’s LP) is unique. If the primal has multiple solutions with at least one non degenerate solution, the optimal solution to the dual is unique. If the primal optimal solution is unique and degenerate, the dual has multiple optimal solutions.

The rest of this section is dedicated to proving Theorem 2. As a direct consequence of best-response equivalence of our classification game to a zero-sum game, we establish the following result.

**Corollary 3.** In NE, the defender’s strategy \( \beta \) maximizes \( \min[\Lambda \beta] - \mu' \beta \); and the attacker has cost \( \delta = \min[\Lambda \beta] \).

**Proof.** If \((\alpha, \beta)\) is a NE of \( G^{R,T} = (V^R, C^T, p, c_d, c_{fa}, P^R_N) \), with attacker’s cost matrix \( \Lambda \), then \((\alpha, \beta)\) is a NE of the zero-sum, best-response equivalent game with cost matrix \( \Lambda_{eq} \). Therefore, \( \beta \) maximizes \( \min_\alpha \alpha' \Lambda_{eq} \beta = \min[\Lambda \beta] - \mu' \beta \). For a given classifier strategy \( \beta \), the minimum attacker cost is achieved by putting positive probability only on strategies
corresponding to the minimum entries of the vector $\Lambda \beta$. Thus the attacker’s optimal cost is $\delta = \min \left[ \Lambda \beta \right]$.

Thus in NE, the defender maximizes her payoff $\min \left[ \Lambda \beta \right] - \mu' \beta$, or equivalently she solves the following linear program (LP):

\[
\begin{align*}
\text{maximize} & \quad -\mu' \beta + z \\
\text{subject to} & \quad z1_{|V_R|} \leq \Lambda \beta \\
& \quad 1'_{|V_R| + 1} \cdot \beta = 1, \beta \geq 0.
\end{align*}
\] (2.19)

We now define the main entities used throughout the section.

**Definition 8.** A **polyhedron** is the solution set of a finite number of linear inequality constraints. An inequality constraint is **tight** if it holds as an equality; otherwise, it is **loose**. A point $x = (x_1, \ldots, x_{|V_R| + 1})$ of a polyhedron is said to be **extreme** if there is no $x'$ whose set of tight constraints is a strict superset of the set of tight constraints of $x$. For an $n$-dimensional linear program, a point is called a **basic solution**, if $n$ linearly independent constraints are tight for that point. A **feasible solution** to a linear program is a solution that satisfies all constraints. A point is a **basic feasible solution**, iff it is a basic solution that is also feasible. Two distinct basic feasible solutions to a set of $n$ linear constraints are **adjacent** if we can find $n - 1$ linear independent constraints that are tight at both of them. We say that a point $x$ of a polyhedron **corresponds** to strategy $\beta$ (or strategy $\beta$ corresponds to $x$), if $\beta = x/\|x\|$.

Extreme point and basic feasible solution are equivalent terms [Lue03, Chapter 2.5] and we will use them interchangeably.
2.3.1 Form of optimal extreme points

In this section, we gain intuition on the structure of the defender’s NE strategy $\beta$. We first show the following lemma.

**Lemma 6.** Any NE strategy $\beta$ of the defender corresponds to an extreme point or a convex combination of extreme points of the polyhedron defined by

\[ P : \Lambda x \geq 1_{|\mathcal{V}_R|}, x_{|\mathcal{V}_R|+1} \geq 0. \]  

(2.20)

*Proof.* From the Fundamental Theorem of Linear Programming, if there exists an optimal solution to an LP, it will occur at an extreme point or a convex combination of extreme points of the polyhedron defined by the constraints. The defender’s LP admits an optimal solution, otherwise Nash’s theorem of NE existence would be violated.

Let $[\beta, z]$ be an extreme point of the polyhedron defined by the constraints of (3.2). We define $x \triangleq \beta/z$. Note that $z = \min[\Lambda \beta] > 0$, since $\Lambda$ is positive, hence $x$ is finite. Substituting for $x$, the constraints in (3.2) become $\Lambda x \geq 1_{|\mathcal{V}_R|}, x \geq 0$. Multiplying with a scalar the objective and feasible region results in a change of parameters (from $\beta$ to $x$) and does not change the problem or the feasible region, thus $x$ is an extreme point of the polyhedron defined by $\Lambda x \geq 1_{|\mathcal{V}_R|}, x \geq 0$. This concludes the proof of Lemma 6. \qed

We call the first type “inequality” constraints and the second type “positivity” constraints. There are $|\mathcal{V}_R|$ inequality constraints and $|\mathcal{V}_R| + 1$ positivity constraints.
Writing down the inequality constraints, we get

\[ c_d x_1 + (r_{|V_R|} - r_1 + \epsilon) \| x \| \geq 1 \]
\[ c_d (x_1 + x_2) + (r_{|V_R|} - r_2 + \epsilon) \| x \| \geq 1 \]
\[ \vdots \]
\[ c_d (x_1 + x_2 + \ldots + x_{|V_R|}) + \epsilon \| x \| \geq 1. \]

Searching for an extreme point of the polyhedron \( P \) is computationally straightforward and there are known algorithms that provide polynomial complexity. In the remaining of this section, our goal is to provide an algorithm that can run faster (though still polynomially) and in parallel to provide intuition on the structure of the extreme points that reveals interesting properties of the Nash equilibria. Essentially, we eliminate suboptimal non-extreme points, so that we reduce the search space.

Combining properties of basic feasible solutions of an LP and properties of the structure of the defender’s LP, we show the following lemma.

**Lemma 7.** There exists exactly one contiguous block (of indices) of inequality constraints that are tight at an extreme point \( x \) that corresponds to a defender NE strategy \( \beta \).

**Proof.** Consider an extreme point \( x \) that corresponds to defender equilibrium strategy \( \beta \) with \( \beta = x/\|x\| \). There exists at least one inequality constraint that is tight at \( x \). Otherwise, \( |V^R| + 1 \) positivity constraints are tight at \( x \) and \( \beta = 0 \) is invalid. Suppose that there exist more than one tight inequality constraints \( x \). Let \( s, f \) be the indices of the first and last tight inequality constraints respectively, and \( l \) be the index of the first loose inequality between \( s \) and \( f \). Subtracting the tight inequality constraint \( l - 1 \) from the loose inequality
constraint \( l \), yields \( x_l > \frac{r_l - r_{l-1}}{c_d} \| x \| > 0 \). We make the following transformation

\[
\hat{x}_i = \begin{cases} 
  x_i, & \text{for } i \in \{0, \ldots, l - 1\} \cup \{l + 2, \ldots, |V^R| + 1\} \\
  x_i - \epsilon_1, & \text{for } i = l \\
  x_i + \epsilon_1, & \text{for } i = l + 1,
\end{cases}
\]

where \( \epsilon_1 > 0 \) is such that \( \hat{x}_l = \frac{r_l - r_{l-1}}{c_d} \| x \| \). Then \( \| \hat{x} \| = \| x \| \). The defender’s payoff for the new strategy \( \hat{\beta} \) is given by:

\[
\min[\Lambda\hat{\beta}] - \mu' \hat{\beta} = \min[\Lambda\beta] - \mu' \beta
\]

\[
> \min[\Lambda\beta] - \mu' \beta,
\]

where the first equality holds because \( \min[\Lambda\hat{\beta}] = \min[\Lambda\beta] = \frac{1}{\| x \|} = \frac{1}{\| \hat{x} \|} \) and the second inequality holds because \( \mu'(\beta - \hat{\beta}) = \mu_l(\beta_l - \hat{\beta}_l) + \mu_{l+1}(\beta_{l+1} - \hat{\beta}_{l+1}) = \mu_l \cdot \epsilon_1 + \mu_{l+1}(-\epsilon_1) > 0 \) and \( \mu \) is a strictly decreasing vector. Thus, the existence of a loose inequality constraint leads to contradiction. This concludes the proof of Lemma 7.

We define \( s, f \) to be the indices of the \textbf{starting} and \textbf{finishing} inequality constraints that are tight. We now formalize the result that describes the optimal extreme points of the defender’s LP.

**Lemma 8.** For extreme points \( x \) of the polyhedron \( P \) that correspond to defender NE strategies, there exists at most one extreme point of Type I and at most two adjacent extreme points of Type II:

**Type I:** \( x = (0, \ldots, 0, x_s, x_s > 0, \ldots, x_{|V^R|} > 0, 0, 0)' \),

**Type II:** \( x = (0, \ldots, 0, x_{s+1} > 0, \ldots, x_{|V^R|} > 0, x_{|V^R|+1} \geq 0)' \),
for some $s_1, s_2 \in \{1, \ldots, |\mathcal{V}^R|\}$.

**Proof.** Suppose $x$ is an extreme point of the polyhedron $P$ that corresponds to a defender NE strategy $\beta$. By Lemma 7, inequality constraints $s$ through $f$ are tight at $x$.

We first show that $f = |\mathcal{V}^R|$. Suppose that $f < |\mathcal{V}^R|$, then any inequality constraint $i$ is loose for $i \in \{f + 1, \ldots, |\mathcal{V}^R|\}$. We make the following transformation with $\|\hat{x}\| = \|x\|$:

$$
\hat{x}_i = \begin{cases} 
  x_i & \text{for } i \in \{0, \ldots, f\} \cup \{f + 3, \ldots, |\mathcal{V}^R|\}, \\
  \frac{r_{f+1} - r_f}{c_d} \|x\| & \text{for } i = f + 1, \\
  x_{f+2} + x_{f+1} - \frac{r_{f+1} - r_f}{c_d} \|x\| & \text{for } i = f + 2.
\end{cases}
$$

The defender’s payoff for the new strategy $\hat{\beta}$ is given by:

$$
\min[\Lambda \hat{\beta}] - \mu' \hat{\beta} = \min[\Lambda \beta] - \mu' \beta
$$

for the same arguments as in the proof of Lemma 7. Contradiction.

We now show that $x_i = 0, \forall i \in \{1, \ldots, s - 1\}$. Suppose that there exists $i \in \{1, \ldots, s - 1\}$ s.t. $x_i > 0$. We make the following transformation: We reduce $x_i$ to $\hat{x}_i$ and increase $x_{i+1}$ by the same amount until either $\hat{x}_i = 0$ or the inequality constraint $i$ is tight. All inequality constraints $j \in \{1, \ldots, |\mathcal{V}^R|\}, j \neq i$ and all positivity constraints $j \in \{1, \ldots, |\mathcal{V}^R|\}, j \neq i, i + 1$ remain intact, but at least one of the inequality constraints or the positivity constraints with index $i$ is tight at $\hat{x}$. Contradiction with the assumption that $x$ was an extreme point. If $x_{i+1} = 0$, then by shifting weight from $x_i$ to $x_{i+1}$, we get the same number of tight inequality constraints but the new point $\hat{x}$ results in a smaller
false alarm penalty cost, since \(\mu_i \cdot x_i > \mu_i \cdot \hat{x}_i + \mu_{i+1} \cdot (x_i - \hat{x}_i)\), and hence to a higher defender payoff. Thus the assumption that \(x\) corresponds to a NE defender strategy \(\beta\) is contradicted. Therefore all extreme points \(x\) that correspond to defender strategy \(\beta\), are vectors of the form

\[
(0, \ldots, 0, x_s, \ldots, x_{|V_R|+1}).
\]

We now show that \(x_i > 0, \forall i \in \{s + 1, \ldots, |V_R|\}\). Subtracting any subsequent tight inequality constraints with indices \(i - 1, i\) with \(i \in \{s + 1, \ldots, |V_R|\}\) yields \(x_i = \frac{r_i - r_{i-1}}{c_d} \|x\| > 0\).

We now show that \(x_s > 0, x_{|V_R|+1} > 0\) cannot happen simultaneously. Suppose that \(x_s > 0\) and \(x_{|V_R|+1} > 0\). Then, we make the following transformation:

\[
\hat{x}_i = \begin{cases} 
\gamma x_i & \forall i \in \{1, \ldots, s - 1\} \cup \{s + 1, \ldots, |V_R|\} \\
0 & \text{for } i = s \\
\gamma (x_s + x_i) & \text{for } i = |V_R| + 1,
\end{cases}
\]

with \(\gamma = \frac{1}{1 - c_dx_s} = \frac{1}{\|x\| [r_{|V_R|} - r_s + \epsilon]} > 1\). The definition of \(\gamma\) is such that the \(s^{th}\) inequality constraint is still tight after the transformation: \(\gamma \left(c_d0 + [r_{|V_R|} - r_s + \epsilon] \|x\|\right) = 1\). The loose inequality constraints before \(s\) become looser and the previously tight inequality constraints \(s + 1\) through \(|V_R|\) remain intact. With the above transformation, we get an extra tight constraint \((x_s = 0)\), thus the previous point was not extreme. Contradiction.

Thus there exist two forms of extreme points \(x^1, x^2\) that may be optimal: each one may satisfy a different block of contiguous tight inequality constraints with starting
indices $s_1$, $s_2$ respectively.

Type I: $x^1 = \left(0, \ldots, 0, x_{s_1}^1 \geq 0, x_{s_1+1}, \ldots, x_{|V_R|}, 0 \right)'$,

Type II: $x^2 = \left(0, \ldots, 0, x_{s_2+1}, \ldots, x_{|V_R|}, x_{|V_R|+1}^2 \geq 0 \right)'$,

where $x_{s_1}^1 = \|x^1\| - \sum_{j=s+1}^{\|V_R\|} x_j^1 \geq 0$, and $x_{|V_R|+1}^2 = \|x^2\| - \sum_{j=s}^{\|V_R\|} x_j^2 \geq 0$.

We now show that there exists a unique starting index $s_1$ such that an extreme point $x$ of Type I is optimal. Suppose there exist two extreme points $x^{1a}, x^{2b}$ of Type I that are both optimal, i.e., they correspond to defender NE strategies $\beta^{1a}, \beta^{1b}$ with

$$\min[\Lambda\beta^{1a}] - \mu'\beta^{1a} = \min[\Lambda\beta^{1b}] - \mu'\beta^{1b}.$$ 

Without loss of generality, suppose that $s_{1b} > s_{1a}$. The inequality constraint with index $|V_R|$ is tight at both extreme points, therefore $\|x^1\| = \|x^2\| = \frac{1}{c_d + \epsilon}$ which yields that $\min[\Lambda\beta^{1a}] = \min[\Lambda\beta^{1b}] = c_d + \epsilon$.

The defender’s payoff for NE strategy $\beta^{1b}$ is given by:

$$\min[\Lambda\beta^{1b}] - \mu'\beta^{1b} = \min[\Lambda\beta^{1a}] - \mu'\beta^{1b}$$

$$> \min[\Lambda\beta^{1a}] - \mu'\beta^{1a},$$

where the last inequality is a result of the strictly decreasing false alarm penalty vector $\mu$. Contradiction. Thus $s_1$ is unique.

We now show that there exist at most two adjacent optimal extreme points of Type II. Suppose that there exist three optimal extreme points $x^1, x^2, x^3$ of Type II that correspond to defender NE strategies $\beta^1, \beta^2, \beta^3$ respectively. Then these should be adjacent,
otherwise there exists a unique optimal extreme point [BT97]. Without loss of generality, suppose that \( s_2, s_2 + 1, s_2 + 2 \) are the starting indices of the block of inequality constraints that are tight at \( x^1, x^2, x^3 \) respectively. Since they are of Type II,

\[
\beta^i_j = 0, \quad \text{for } 1 \leq i \leq 3, \quad 0 \leq j \leq s_2 + i - 1.
\] (2.22)

Since all three extreme points are optimal,

\[
\min[\Lambda \beta^1] - \mu^1 \beta^1 = \min[\Lambda \beta^2] - \mu^1 \beta^2 = \min[\Lambda \beta^2] - \mu^3 \beta^3.
\] (2.23)

The first inequality constraint that is tight for each extreme point yields:

\[
\min[\Lambda \beta^i] = r_{|V^R|} - r_{s_i + \epsilon}, \quad \text{for } i \in \{1, 2, 3\},
\]

and since the block of inequality constraints \( s_2 + 2 \) through \( |V^R| \) that are tight at all extreme points is common, we get:

\[
\beta^i_j = \frac{r_j - r_{j-1}}{c_d}, \quad \text{for } 1 \leq i \leq 3, \quad s_2 + i - 1 \leq j \leq |V^R|.
\]

Therefore, as a result of the first and second equality in (2.23) respectively:

\[
c_d = \mu_{s_2 + 1} - \mu_{|V^R| + 1}, \quad \text{(2.24)}
\]

\[
c_d = \mu_{s_2 + 2} - \mu_{|V^R| + 1}. \quad \text{(2.25)}
\]

Solving (2.24), (2.25) simultaneously yields \( \mu_{s_2 + 1} = \mu_{s_2 + 2} \), which contradicts the assumption that \( \mu \) is a strictly decreasing vector. This concludes the proof of Lemma 8.

\[\square\]

**2.3.2 Form of players’ equilibrium strategy**

Having described the form of the optimal extreme points of the polyhedron \( P \) that defines the constraints of the defender’s optimization problem, we now provide the proof for our main result in this section.
Proof of Theorem 2.

Part I (Defender’s NE strategy): By Lemma 6, the defender’s equilibrium strategy $\beta$ corresponds to an extreme point or a convex combination of extreme points of the polyhedron $P$ defined by (3.3). There exist three possible scenarios:

1. The optimal solution is of Type I. Then by Lemma 8 there exists a unique index $s_1$, with $\beta_{s_1} > 0$ and $\beta_{|V^R|+1} = 0$. Therefore the defender’s strategy has the form in 2 with $k = s_1$.

2. There exist at most two adjacent optimal extreme points of Type II. If there is a unique optimal extreme point of Type II, then $s_2 = k$ and $\beta_k = 0$. Otherwise, $\beta$ is a convex combination of the two adjacent optimal extreme points and $\beta_{s_2} = 0, k = s_2 + 1$.

3. There exists one optimal extreme point of Type I and one optimal extreme point of Type II. These extreme points should be adjacent, therefore $s_1 = s_2 = k$. By Lemma 8 $\beta$ is a convex combination of these extreme points, which yields $\beta_k \geq 0, \beta_{|V^R|+1} \geq 0$. In case $\beta_k = \beta_{|V^R|+1} = 0$, we say that the two types of extreme points coincide (two different bases yield the same optimal extreme point).

In all the above cases, subtracting any subsequent tight inequality constraints with indices $i-1, i$ with $i \in \{k + 1, \ldots, |V^R|\}$ yields $\beta_i = \frac{r_i - r_{i-1}}{c_d}$, and since $\beta$ is a probability measure, $\beta_k + \beta_{|V^R|+1} = 1 - \sum_{j=k+1}^{|V^R|} \beta_j$.

Part II (Attacker’s NE strategy): We have shown that the classification game is best-response equivalent to a zero-sum game in which the attacker minimizes $\alpha' \Lambda^{eq} \beta$ and the defender maximizes $\alpha' \Lambda^{eq} \beta$. Therefore the attacker’s dual is the defender’s LP [BV04,
Chapter 5.2. By complementary slackness [NS96, Chapter 6.2.1] the slack constraints in the dual (loose inequality constraints in the polyhedron $\Delta x \geq 1$) must be associated with strategies played with zero probability in the primal. Thus $\alpha_i = 0, \forall i \in \{1, \ldots, s - 1\}$.

By Lemma 5 and Proposition 1, for attacker strategies $r \in \mathcal{V}^R$ with $\pi_d(r) \in (0, 1)$

$$\alpha_{r_i} = \frac{1 - p}{p} c_{fa} P_N^R(r_i),$$

(2.26)

where $P_N^R(r_i)$ is given by (2.13). For $\beta$ with:

1. $\beta_k > 0, \beta_{|\mathcal{V}^R|+1} > 0 : \pi_d(r_k) \in (0, 1)$ and $\pi_d(r_{|\mathcal{V}^R|+1}) \in (0, 1)$, therefore (2.26) holds for $i = k, |\mathcal{V}^R| + 1$.

2. $\beta_k > 0, \beta_{|\mathcal{V}^R|+1} = 0 : \pi_d(r_k) \in (0, 1)$, therefore (2.26) holds for $i = |\mathcal{V}^R|$ and $\alpha_k = 1 - \sum_{j=k+1}^{||\mathcal{V}^R||} \alpha_j$.

3. $\beta_k = 0, \beta_{|\mathcal{V}^R|+1} > 0 : \pi_d(r_{|\mathcal{V}^R|+1}) \in (0, 1)$, therefore (2.26) holds for $i = k$ and $\alpha_{|\mathcal{V}^R|} = 1 - \sum_{j=k}^{||\mathcal{V}^R||-1} \alpha_j$.

4. $\beta_k = \beta_{|\mathcal{V}^R|+1} = 0 :$ Any $\alpha_k, \alpha_{|\mathcal{V}^R|}$ s.t. $\alpha_k + \alpha_{|\mathcal{V}^R|} = \sum_{j=k+1}^{||\mathcal{V}^R||} \alpha_j$ are NE. Indeed, when $\beta_k = 0$ then

$$c_d = r_{|\mathcal{V}^R|} - r_k.$$  

(2.27)

The pure strategies $k, |\mathcal{V}^R|$ yield

$$U^A(r_k, \beta) = r_k - c_d \cdot \pi_d(r_k) = r_k$$

$$U^A(r_{|\mathcal{V}^R|}, \beta) = r_{|\mathcal{V}^R|} - c_d \cdot \pi_d(r_{|\mathcal{V}^R|}) = r_{|\mathcal{V}^R|} - c_d,$$

and from (2.27), we get $U^A(r_k, \beta) = U^A(r_{|\mathcal{V}^R|}, \beta)$. Thus both pure strategies give the same attacker payoff and could be included in the support of the attacker, with any
weight such that $\alpha$ is a valid probability measure. (Note that the weights given to strategies with indices $k, |V^R|$ do not impact the equilibrium strategy of the defender since $\beta_k = 0$ and $\beta_{|V^R|}$ is fixed.)

2.4 Numerical Results

In this section, we provide a series of different scenarios on which we applied our model. We investigate the Nash equilibrium results and give insights on the behavior of the players in NE. We explore the role of the noise (non-attacker) and specifically how the strategic attacker exploits the knowledge he has for the non-attacker to confuse the defender.

2.4.1 Single target

We first consider $G^{R,T} = (V^R, C^T, p, c_d, c_{fa}, P^N)$, in which there is a single target of interest. The attacker’s strategy space consists of $N + 1$ attack rewards $r_0, \ldots, r_N$, with $r_i = i \cdot c_a$ representing the attack reward when the target is compromised $i$ times. The defender’s strategy space $C^T$ consists of all threshold classifiers on $r_i$. The attacker bears a cost of $c_d$ upon detection. The false alarm penalty for the defender when he mistakenly classifies the non-attacker as an attacker is expressed by the factor $c_{fa}$.

The non-attacker, which one can think of like a normal user (or noise), accesses the target $i$ times with binomial distribution: attack reward $r_i$ is the outcome of $i$ successes
Algorithm 1: How to compute the NE \((\alpha, \beta)\)

1. for type = 1, 2 do
2.   construct \(\beta\) for \(s \in \{1, \ldots, |V^R|\}\) using Algorithm 2;
3.   find \((\beta_{1,2}, s^*_1, 2)\) that maximize \(U_{1,2}^D\);
4. if \(U_1^D > U_2^D\) then
   5.     \(\beta \leftarrow \text{compute-}\beta(s^*_1, 1)\); \(\alpha \leftarrow \text{compute-a}(s^*_1)\);
6. if \(U_1^D < U_2^D\) then
   7.     if \(s_2\) is unique then
      8.       \(\beta \leftarrow \text{compute-}\beta(s^*_2, 2)\); \(\alpha \leftarrow \text{compute-a}(s^*_2)\);
   9.     else
      10.    \(\beta_{2a} \leftarrow \text{compute-}\beta(s^*_2a, 2)\); \(\beta_{2b} \leftarrow \text{compute-}\beta(s^*_2b, 2)\);
      11.    \(\beta\) is a conv. combination of \(\beta_{2a}, \beta_{2b}\); \(\alpha \leftarrow \text{compute-a}(max(s^*_2a, s^*_2b))\)
12. if \(U_1^D = U_2^D\) then
   13.    \(\beta_1 \leftarrow \text{compute-}\beta(s^*_1, 1)\); \(\beta_2 \leftarrow \text{compute-}\beta(s^*_1, 2)\);
   14.    if \(\beta_1 \neq \beta_2\) then
      15.      \(\beta\) is a conv. combination of \(\beta_1, \beta_2\).
   16.    else
      17.      \(\beta = \beta_1\)
   18.    \(\alpha \leftarrow \text{compute-a}(s^*_1)\)
Algorithm 2: Compute-\(\beta(s, \text{type})\)

1 \(\beta_0\) to \(\beta_{s-1}\) ← 0;

2 for \(i ← s + 1\) to \(|V^R|\) do

3 \(\beta_i ← \frac{r_i - r_{i-1}}{c_d}\);

4 \(\beta_s ← 1_{\text{type}=1}(1 - \sum_{i=s+1}^{\beta_s})\);

5 \(\beta_{|V^R|+1} ← 1_{\text{type}=2}(1 - \sum_{i=s+1}^{\beta_{|V^R|+1}})\);

6 \(U^D_{\text{type}} ← \min[\Lambda\beta] - \mu'\beta\);

Algorithm 3: Compute-\(\alpha(k)\)

1 \(\alpha_0\) to \(\alpha_{s-1}\) ← 0;

2 for \(i ← k + 1\) to \(|V^R| - 1\) do

3 \(\alpha_i ← 1 - p\,\frac{c_{fa}}{c_d} P^R_N(r_i)\);

4 if \(\beta_k > 0\) then

5 \(\alpha_k ← 1 - p\,\frac{c_{fa}}{c_d} P^R_N(r_k)\);

6 else if \(\beta_{|V^R|+1} > 0\) then

7 \(\alpha_{|V^R|} ← 1 - p\,\frac{c_{fa}}{c_d} P^R_N(r_{|V^R|})\);

8 \(\alpha_k ← 1 - \sum_{i=k+1}^{\beta_{|V^R|+1}} \alpha_i\);

9 else if \(\beta_k = \beta_{|V^R|+1} = 0\) then

10 any \(\alpha_k \in [0, 1 - \sum_{i=k+1}^{\beta_{|V^R|+1}} \alpha_i]\)

11 \(\alpha_{|V^R|} ← 1 - \sum_{i=k}^{\beta_{|V^R|}-1} \alpha_i\);
over $N$ trials with probability of success $\theta_0$. His behavior results in a distribution

$$P^R_N = \binom{N}{k} \theta_0^k (1 - \theta_0)^{N-k}. \quad (2.28)$$

Figure 2.3: The attacker uses a truncated version of the distribution of the non-attacker in NE.

Figure 2.3 shows that in equilibrium, in the support of the defender, the attacker uses a truncated, scaled version of the distribution of the non-attacker (as shown in Lemma 5). The game parameters are $c_a = 1$, $c_d = 120$, $p = 0.2$, $c_{fa} = 140$, $\Psi^R = \{r_0, \ldots, r_i, \ldots, r_{100}\}$, with $r_i = ic_a$, and $P^R_N(r_i)$ given by (2.28) with $\theta_0 = 0.3$. Note that this equilibrium is calculated by using the results from Section 2.3. The defender uses a set of contiguous strategies (thresholds on the attack reward). Furthermore, the weight given to each strategy is constant in the interior of the support and equal to $\beta_i = \frac{r_i - r_{i-1}}{c_d} = \frac{c_a}{c_d} = \frac{1}{120}$. One can use this analysis to show how the equilibrium payoff of the players changes with respect to the various cost parameters and the probability distribution of the non-attacker across the attacker’s strategy space.

In Figure 2.4 we see the impact of the reward from each target attack vector.
As each attack becomes more and more valuable, the cost of detection which is constant becomes less important than the attack reward. The attacker favors the most rewarding attack vectors (with higher \(c_a\)) versus the cost of getting detected, inflicting a higher cost to the defender. Hence, the attacker’s payoff is increasing while the defender’s payoff is decreasing as \(c_a\) increases.

![Graph](image)

Figure 2.4: As each attack becomes more and more valuable, the attacker’s payoff is increasing while the defender’s payoff is decreasing.

In Figure 2.5 we observe the impact of the false alarm penalty. When the false alarm penalty increases, the defender is more conservative and selects more often thresholds on rewards less frequently generated by the non-attacker (which in our case are the higher thresholds). The attacker exploits this space and becomes more aggressive without risking getting detected.

As we observe in Figure 2.5, the attacker’s expected payoff has a staircase nature. This is an effect of the discrete optimal values of the starting index \(s\) of the contiguous block of tight inequality constraints: for some region of parameters, the same \(s\) is optimal. For the same optimal \(s\), the support of the defender’s strategy stays the same and the detection benefit, hence the attacker’s equilibrium payoff is the same. The defender’s staircase nature
is slightly affected by the false alarm penalty factor in her payoff function.

2.4.2 Investment strategies of the defender

The feature vector of the defender, upon which the classification is based, consists of multiple numerical features. For example, the feature vector of a classification algorithm of Twitter that needs to classify accounts as legitimate (class 1) or fraudulent (class 0), might consist of features such as number of followers, average number of retweets per tweet, number of mentions, and others.

In the experiments presented above, the classification was based on a single feature (or else, multiple features combined to a single one). In a security environment, one important decision of the defender (or network administrator) is to decide whether an additional feature (and a sensor to acquire data about this second feature) would help in the attacker classification.

For the purposes of the following experiment, we assume that the defender al-
ready has access to information about what type of servers in her network the attacker compromises. A second source of information is inspired by the literature of detecting portscanners [JPBB04b]. Based on empirical data, Jung et. al. found a distinction between benign and malicious portscanners in terms of the proportions of the connections that were successfully established. They proposed a new metric, called “inactive-pct”, as the ratio of number of hosts to which failed connections are made versus the number of hosts to which successful connections are made. Benign users (e.g. web search engines) have low “inactive-pct” (< 80%), as a larger fraction of connections will be successfully established. On the contrary, malicious portscanners will initiate a lot of connections, to detect vulnerabilities, but then terminate them immediately, as it is costly otherwise to maintain them.

Based on the above analysis, we assume that a second feature exists about whether the “inactive-pct” is low (< 80%) or high (> 80%). Depending on the underlying parameters, the extra information coming from the second sensor might benefit the defender or not. Our analysis provides the tool to the defender to establish a priori whether to invest and if yes, how much. We investigate the following four scenarios.

1. In scenario 1, the defender can only observe how many times an agent accesses a target of interest over a window of $N = 2$ time slots. At each time slot the agent can attack the target at most once. Each access to the target yields an attack reward of $c_a$. There are thus three attack vectors of the attacker, giving reward $r_0 = 0$, $r_1 = c_a$, $r_2 = 2c_a$. The attacker can still access the remaining targets / ports of the network, and either maintain a low or a high “inactive-pct”, which is unidentifiable by the defender (common
information). A strategic attacker, would therefore never select to attack with a low “inactive-pct” and his dominant strategy would always be to attack with a high “inactive-pct”.

2. In scenario 2, we assume that the equilibrium strategy of the attacker is fixed in the previous scenario. The defender on the other hand has now access to a second sensor, based on which he can compute the “inactive-pct” induced. In other terms, the attacker does not know that the defender uses this second feature in his feature vector and does not change his strategy. We expect a dip in the attacker’s payoff.

3. In scenario 3, we assume that the defender thinks that the attacker’s distribution is going to be fixed, but in reality, the attacker adapts his strategy. In other terms, the defender uses a single feature in the training phase, but the attacker knows that in real classification he is going to get detected based on both features.

4. In scenario 4 we assume that the attacker has figured out that he is classified based on the two features described before, and adapts his strategy.

Our goal is to compare the defender’s equilibrium payoff in the above scenarios, and see under what circumstances the defender’s value (expected payoff in equilibrium) will be greater.

As we see in Figure 2.6, when the attacker does not know that the defender classifies him based on two features (scenario 2), the defender’s payoff increases from the scenario 1 in which the defender only classified the agent based on a single feature. When the attacker finds out that he is getting classified on multiple features, then the defender’s
Figure 2.6: Scenario 1: Defender does not differentiate between low and high “inactive-pct”. Scenario 2: Attacker mistakenly believes he is only classified based on a single feature. Scenario 3: Defender mistakenly believes that attacker believes he is classified based on a single feature. Scenario 4: both players know that two sensors (features) determine classification.

value decreases from scenario 2, since the attacker is smart enough to realize why he got detected. Comparing scenarios 1 and 4, we see that in this experiment, the defender’s payoff in equilibrium increases when she has access to two features. The parameters are $c_a = 1, c_{low} = 2$, (benefit to the attacker by using a low “inactive-pct”), $c_{high} = 4.1$ (benefit to the attacker by using a high “inactive-pct”), $c_d = 1, p = 0.2, N = 2, \theta_0 = 0.3$ (non-attacker’s frequency to attack the target at each time slot), $\theta_{low} = 0.8$ (non-attacker’s probability to have a low “inactive-pct”).

2.5 Related Work

The intersection of game theory and security has inspired a lot of research [RES+10], [MZA13], [Lun93], [LW02], [Chr11], [AB10]. Sommer and Paxson investigated the intruder detection challenge from a machine learning perspective [SP10]. A recent study by Vorobeychik and Li [VL14] recognizes the inefficiency of deterministic classifiers for the detection
of intruders: They reach a similar result with ours in a specific scenario / attack model (uniform randomization on two different classifiers). Their analysis is based on machine learning tools; and their uniform randomization result is not a Nash equilibrium. Our analysis is based on game theoretic tools, and we consider a more general form of games (a special case of nonzero-sum games).

The classification game we investigate is similar in nature with the inspection game, a multi-stage game between a customs inspector and a smuggler, proposed and studied by Dresher [Dre62] and Maschler [Mas66]. In [ASZ02], Avenhaus et. al. find the equilibrium of the general nonzero-sum game by using an auxiliary zero-sum game in which the inspectee chooses a violation procedure and the inspector chooses a statistical test with a given false alarm probability. We do not separate the general nonzero-sum game to two games but show that the equivalence to a zero-sum game and provide structure to the equilibrium strategies of a single-shot simultaneous-move game.

Chen and Leneutre [CL09] address the intrusion detection problem in heterogeneous networks consisting of nodes with different non-correlated security assets, in the same way that different targets are of different value in our system. The main difference with our work is that their detection probability is the same for any attack and defense strategies, yielding the utility function to be the sum of the utilities on each target. On the contrary, we involve different detection and false alarm rates, which makes the set-up more realistic. We also assume asymmetric information, since the defender is not aware of the attacker’s type.

Lye and Wing [LW02] have also investigated a security problem with multiple tar-
gets. They analyze a stochastic game between an attacker and administrator and compute the best response strategies of the players using a non-linear program, while we analyze the Nash equilibria of a classification game.

The authors in [BT13] also use a game theoretic approach to solve the problem of source identification. They consider though a zero-sum game and derive an asymptotic Nash equilibrium with numerical computations, while we analyze the interesting case of a non-competitive game and derive closed-form expressions for the equilibria.

Another work related to the present work is [SLL12], where the authors are using game theory to solve for the optimum defense strategies, given a certain attack. Game theory is used to derive a best response for the defender given a fixed attack distribution, and does not characterize the Nash equilibria of the game, as the attacker could as well deviate. On a similar setting, the authors in [DDM+04] address the problem of classifying a malicious intruder in the presence of an innocent user. The malicious intruder can perturb his behavior to confuse the classifier and evade detection. But, their work focuses only on one iteration of the game and how each player can once adjust his strategy to optimize his expected payoff, given the optimal strategy of the other player. On the contrary, we provide an algorithm to compute the Nash equilibria of the game.

2.6 Concluding Remarks

We investigated a classification game, in which a defender seeks to classify an agent as a strategic attacker or a non-strategic non-attacker. We showed that the strategy space in NE can be reduced significantly for both players: The attacker mixes among attack vectors
that give a certain attack reward. The defender selects a distribution on thresholds on the
attack reward. By taking advantage of the structure of the payoff formulation, we provided
an algorithm to compute the players NE strategies in polynomial time. Furthermore, we
developed intuition on the impact of the various game parameters on the NE strategies.

Our results exhibit a correlation between the behavior of a benign user (non-
attacker) and the attacker’s behavior that suggests how the attacker mixes his strategy to
resemble the norm and remain stealthy. Finally, we explore the incentives of a strategic
defender to invest in a second sensor capturing more data about the underlying agent: the
decision of whether to invest or not is not a simple yes/no answer. Overall, our results show
how to combine features in a strategic setting (by looking at the attack reward they give),
differentiating our work from known classification algorithms like logistic regression.

Note that due to the special structure of the payoffs, our game is best-response
equivalent to a zero-sum game for which the complexity is reduced. Surprisingly, although
this equivalence seems straightforward, the literature in security games is largely looking
at zero-sum games. Thus models that better capture the realistic scenarios in which the
defender and attacker have different tradeoffs in their objective functions become now com-
putationally equivalent to zero-sum games.

Our analysis identifies the need for more sophisticated countermeasure defenses.
One can develop classifiers that successfully identify fraudulent activity but the implementa-
tion of these classifiers is as important as the task of classification. A strategic defender will
(and should) expect that a strategic adversary will be able to detect over time what classi-
fiers were used, and try to circumvent them, as was shown in the experiment in [TMG+13].
Two weeks after Twitter has suspended several million fraudulent accounts based on certain features / classifiers the number of automatically classified as fraudulent accounts fell from 95% to 67%, and the underground market for twitter accounts re-ignited. Randomization and game theory could help solve this problem, as the strategic interactions of defender and attacker are inherently accommodated and reflected in their payoff functions.
Chapter 3

Classification game with generalized payoff functions

The classification game presented in Chapter 2 has a specific form for the payoff functions of the players. For example, in the event that the attacker is detected, he loses a fixed cost captured by the parameter $c_d$. In this chapter we generalize the payoff functions of the players, motivated by the need to capture more realistic scenarios in which the attacker has a detection cost that depends on his attack strategy.

We provide a polynomial time algorithm to derive the equilibrium strategies of the two players. Furthermore, we propose two characteristic examples, in which the defender has made different security investments in forensic tools. The analysis of these models provide us with a qualitative and quantitative view on how changes on the network parameters affect the strategies of the players at equilibrium.

Summary of contributions.
• We propose a generic game-theoretic model to analyze the interactions between two adversaries: a defender and a malicious attacker when a non-strategic user (non-attacker) is present (Section 3.1).

• We show under which assumptions on the nature of the payoff functions of the players the results derived in Chapter 2 apply (Section 3.2).

• We develop two models for an adversarial classification game (Sections 3.3.1 and 3.3.2) that rely on different assumptions on the defender’s forensic tools.

• By comparing the above two models, we extract key insights on the expected gain from the defender’s investment in forensic capabilities. This is an example of how our methodology can be used to evaluate how changes in the strategic situation affect the equilibrium payoffs of the players. We also investigate the impact of the different network parameters on the resulting NE strategies (Section 3.4).

We give a summary of the related work in Section 3.5 and we conclude the chapter with our remarks in Section 3.6.

3.1 Game Model

The network we consider consists of a defender and two servers that she monitors for potential attacks: a File Server (FS) with sensitive data and a Mail Server (MS) with contents of inferior importance. The defender observes the number of hits from an attacker to each server for a fixed classification window of $N$ time slots. The attacker may be a spy
or a spammer with probabilities $p$ and $1 - p$ respectively\footnote{Note that in this chapter we are using the terms spy instead of attacker and spammer instead of non-attacker. In this context, the behavior of the spammer is not targeted but results in various FS and MS hits, while the spy is intelligent and strategic and seeks to remain stealthy.}. The defender is a strategic player that seeks to correctly classify the potential intruder by selecting a threshold\footnote{Note that in this chapter, we assume that the defender is constrained into selecting threshold rules. As a sanity check we can show that the assumptions on the nature of the payoff functions are in accordance with this result.} $T$. When she observes $T$ or more hits on the FS, she classifies the attacker as spy; otherwise as spammer. The spy is also a strategic player that selects the number of FS attacks $H$ he will perform. He seeks to attack the FS as frequently as possible, while evading detection. The spammer is a non-strategic player that mostly attacks the MS and adds noise to the network. He also attacks the FS $Z$ times ($Z$ follows a known distribution). For example, the spammer can be modeled to follow the binomial distribution, with a small probability $\theta_0$ to attack the FS at each time slot.

Our solution captures a more general setting than the one presented above. We only require that the attacker has some cost function if he gets detected or missed. We describe the model around the example scenario in which there are two servers, one of which is of primary interest to the strategic attacker (the file server) in order to be more concrete. However, the model we develop is quite general and applicable to many settings in which there is a target of special interest to a strategic attacker but who is incentivized to mix his attack across other targets to make classification more difficult.

### 3.1.1 Spy’s cost function

The spy cost depends on the defender’s classification decision and the number of FS hits. We denote the spy cost function when the spy is detected (i.e., $T \leq H$) by $D(H)$.
and when the spy is not detected (i.e., $T > H$) by $M(H)$. Thus, the overall spy cost function is expressed as follows:

$$J_A(T, H) = D(H) \cdot \mathbb{1}_{T \leq H} + M(H) \cdot \mathbb{1}_{T > H},$$

or by making the appropriate simplifications

$$J_A(T, H) = [D(H) - M(H)] \cdot \mathbb{1}_{T \leq H} + M(H).$$

### 3.1.2 Defender’s payoff function

We now describe how the defender’s expected payoff function is constructed. We distinguish two cases:

- With probability $p$ the defender faces a spy. If the defender correctly classifies the intruder as a spy (i.e., $T \leq H$), she gains $D(H)$. If the defender misclassifies the spy (i.e., $T > H$), she gains $M(H)$.

- With probability $1-p$ the defender faces a spammer. If the defender correctly classifies the intruder as spammer (i.e., $T \geq Z$), she does not benefit. The defender incorrectly classifies the spammer with probability $\phi(T) = \Pr\{Z \geq T\}$ and in this case there is a false alarm penalty $c_{fa}$.

Combining these two scenarios, the defender’s expected payoff is

$$\tilde{U}_D(T, H) = p \cdot [D(H) \cdot \mathbb{1}_{T \leq H} + M(H) \cdot \mathbb{1}_{T > H}] - (1 - p) \cdot c_{fa} \cdot \phi(T). \quad (3.1)$$

Multiplying with the constant $1/p$ we get

$$U_D(T, H) = D(H) \cdot \mathbb{1}_{T \leq H} + M(H) \cdot \mathbb{1}_{T > H} - \mu(T),$$
where $\mu(T) = \frac{1 - p}{p} \cdot c_{fa} \cdot \phi(T)$. Function $\phi(T)$ is decreasing on $T$, and we assume that it is strictly decreasing: $\Pr\{Z \geq T\} > \Pr\{Z \geq T + 1\}$. Let us note here that multiplying a payoff constant with a constant factor does not change the best responses, hence the Nash equilibria will be the same.

3.1.3 Players’ interactions

For a classification window of $N$ time slots, the spy has $N + 1$ available actions (attack the file server $H \in \{0, \ldots, N\}$ times). The defender has $N + 2$ available actions (select $T \in \{0, \ldots, N + 1\}$ as the classification threshold).

We model our problem as a nonzero-sum game. However, the defender’s payoff is different from the spy’s cost function in only one term $\mu(T)$ that depends only on the defender’s strategy ($U_{D}(T, H) = J_{A}(T, H) - \mu(T)$). These games are known as almost zero-sum games and are best-response equivalent to zero-sum games, as shown in Chapter 2.

Let $\tilde{A}$ be a $(N + 1) \times (N + 2)$ matrix representing the spy’s (pure) strategies’ cost. We express the cost matrix of the attacker as

$$\tilde{A} = \begin{bmatrix}
\delta(0) & 0 \\
\vdots & \ddots & \ddots & \vdots \\
0 & \delta(N) \\
\end{bmatrix}
\begin{pmatrix}
1 & 0 & \cdots & \cdots & 0 & 0 \\
\vdots & 1 & \ddots & \ddots & \vdots & \vdots \\
\vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\
0 & \cdots & \cdots & 0 & \vdots \\
1 & \cdots & \cdots & 1 & 0 \\
\end{pmatrix}
\begin{pmatrix}
M(0) & \cdots & M(0) \\
M(1) & \cdots & M(1) \\
\vdots & \ddots & \vdots \\
M(N - 1) & \cdots & M(N - 1) \\
M(N) & \cdots & M(N) \\
\end{pmatrix},$$

where $\delta(H) = D(H) - M(H)$. Each row $i$ of $\tilde{A}$ corresponds to one of the $N + 1$ possible spy strategies. For instance, row “0” corresponds to spy attacking the FS 0 times (or $H = 0$),
row “1” corresponds to spy selecting $H = 1$ and so on. Each column of $\tilde{\Lambda}$ corresponds to one of the $N+2$ possible defender strategies.

Let $\tilde{\Lambda}$ be defined as above, and $\alpha, \beta$, be the spy and defender distributions respectively. The attacker cost can be written as $\alpha' \tilde{\Lambda} \beta$ and the defender payoff can be written as $\alpha' \tilde{\Lambda} \beta - \mu' \beta$, where $\mu$ is a strictly decreasing vector (component-wise) with $\mu_i$ be the $i^{th}$ component of vector $\mu$. Certain computations are simplified by using a matrix with only positive entries. We define

$$\Lambda = \tilde{\Lambda} + K \cdot 1_{(N+1) \times (N+2)},$$

where $K > 0$ is such that every matrix element is positive. Since $\alpha$ and $\beta$ must each sum to 1, the expressions $\alpha' \Lambda \beta$ and $\alpha' \Lambda \beta - \mu' \beta$ are respectively the attacker cost and defender payoff shifted by a constant. Adding a constant to the players’ payoff does not affect their best responses, thus from here on we will consider these expressions to be the payoff functions of each player.

### 3.2 Game Theoretic Analysis

The game described in the previous section is best-response equivalent to a zero-sum game. Indeed, vector $\mu$ depends only on the spammer’s complementary cumulative probability distribution and is fixed. Adding constants to the columns of the cost matrix of the attacker (row player) in the original classification game yields the cost matrix of the defender of the new game without changing the Nash equilibria of the game. Indeed, the defender’s payoff matrix is unchanged, and, for any given $\beta$, minimizing $\alpha' \Lambda \beta$ and minimizing $\alpha' \Lambda^{eq} \beta$ with $\Lambda^{eq}(i,j) = \Lambda(i,j) - \mu_j$ give the same minimizing strategy for the
attacker. Thus the two games generate the same sets of best response functions hence they are best-response equivalent and have the same equilibrium strategies.

As a direct consequence of best-response equivalence of our classification game to a zero-sum game, we establish the following result.

**Corollary 4.** In NE, the defender’s strategy $\beta$ maximizes $\min[\Lambda \beta] - \mu'\beta$; and the attacker has cost $\delta = \min[\Lambda \beta]$.

Thus in NE, the defender maximizes her payoff $\min[\Lambda \beta] - \mu'\beta$, or equivalently she solves the following linear program (LP):

$$\begin{align*}
\text{maximize} & \quad -\mu'\beta + z \\
\text{subject to} & \quad z1_N \leq \Lambda \beta \\
& \quad 1'_{N+1} \cdot \beta = 1, \quad \beta \geq 0.
\end{align*}$$

(3.2)

Similarly with Section 2.3 in the previous chapter, we can show that the best-response equivalence of the classification game to a zero-sum game leads to the following result.

**Lemma 9.** Any NE strategy $\beta$ of the defender corresponds to an extreme point or a convex combination of extreme points of the polyhedron defined by

$$P : \Lambda x \geq 1_{N+1}, x_{N+2} \geq 0.$$  

(3.3)

We call the first type “inequality” constraints and the second type “positivity” constraints. There are $N + 1$ inequality constraints and $N + 2$ positivity constraints. The difference in this chapter is the assumption on the form of the cost matrix $\Lambda$. Writing down the inequality constraints, we get:
Our goal is to eliminate nonextreme and other points that are not selected by a
defender in NE, so that we reduce the number of points we have to check. Depending on
the nature of the attacker’s cost functions $\delta$ and $M$, we are able to compute analytically
the defender’s NE strategies in polynomial time. We will consider the following conditions
for the subsequent analysis.

**Condition 1 (Discrete convexity).**

$\Delta_1 \delta(s + 1) \geq \Delta_1 \delta(s)$

$\Delta_1 M(s + 1) \geq \Delta_1 M(s)$,

$\forall s \in \{0, \ldots, N - 1\}, \text{where } \Delta_k g(i) = g(i + k) - g(i)$.

Condition 1 suggests that the difference between the cost of the spy upon detection
and his cost upon misdetection is non decreasing with respect to $H$. It also suggests that
the marginal cost for the spy when he is not detected is smaller for smaller values of $H$. We
use this condition to prove that the inequalities are violated, unless there is a contiguous
block of tight inequalities (see Lemma 12).
Condition 2:

1. $D(H)$ is monotone with respect to the number of attacks to the FS $H$.

2. $M(H)$ is a decreasing function with respect to $H$.

We use condition 2 to show that the contiguous block of inequality constraints that are tight at an extreme point extends to the last $N^{th}$ inequality constraint. Furthermore, we can see that the maximization of the defender’s payoff reduces to searching for a single parameter $\beta_{N+1}$. Theorem 3 summarizes our results on the computation of Nash equilibria for the attacker classification game with generalized payoff functions.

**Theorem 3.** Under condition 1, the defender NE strategy $\beta$ satisfies a contiguous block of tight inequality constraints (indexed $s$ through $f$). Under condition 2, the contiguous block will finish at index $f = N$, or we only have pure NE. When $f = N$, we search amongst different $\beta_{N+1}$ for the defender strategies $\beta$ that maximize the defender’s equilibrium payoff. The remaining elements of $\beta$ result from solving the fewest tight inequalities (maximum allowed integer $s$).

We now develop a series of lemmas that lead to Theorem 3.

**Lemma 10.** Two points $x_1$ and $x_2$ on the polyhedron, with $\|x_1\| = \|x_2\|$, correspond to defender strategies $\beta_1$ and $\beta_2$ respectively with detection cost $\min[\Lambda_1\beta_1] = \min[\Lambda_2\beta_2]$ against a best responding attacker.

**Proof.** We showed in Lemma 9 that a defender NE strategy $\beta$ corresponds to one of the extreme points of a polyhedron defined by $\Lambda x \geq 1_{N+1}$, $x \geq 0$, with $\|x\| = 1/z = 1/\min[\Lambda \beta]$.  

67
Thus, for the same the norm $\|x\|$, we get the same detection cost against a best responding attacker, i.e., $\min[\Lambda \beta_1] = \min[\Lambda \beta_2]$.

**Lemma 11.** If $\|x_1\| = \|x_2\|$ and $\mu'_x x_1 < \mu'_x x_2$, then $x_1$ corresponds to a defender strategy $\beta_1$ with a higher defender equilibrium payoff, i.e., $\min[\Lambda \beta_1] - \mu'_x \beta_1 > \min[\Lambda \beta_2] - \mu'_x \beta_2$.

**Proof.** The defender’s payoff for equilibrium strategy $\beta$ is given by the formula $\min[\Lambda \beta] - \mu'_x \beta$. We get

$$\min[\Lambda \beta_1] - \mu'_x \beta_1 - (\min[\Lambda \beta_2] - \mu'_x \beta_2) = \mu'_x \beta_2 - \mu'_x \beta_1$$

(3.4)

$$> 0,$$

(3.5)

where (3.4) results from Lemma 10 (since $\|x_1\| = \|x_2\|$, $\min[\Lambda \beta_1] = \min[\Lambda \beta_2]$), and (3.5) follows the assumption $\mu'_x x_1 < \mu'_x x_2$. The point $x_1$ corresponds to a defender strategy $\beta_1$ with a smaller false alarm cost, i.e., $\mu'_x \beta_2 > \mu'_x \beta_1$.

**Lemma 12.** Under condition 1, an extreme point $x$ corresponding to a defender NE strategy $\beta$ satisfies exactly one contiguous set (of indices) of tight inequalities.

**Proof.** An extreme point $x$ satisfies at least one tight inequality. If none of the inequalities are tight, we scale the vector $x$ down until one inequality becomes tight. The new vector’s set of tight inequalities is a strict superset of those of the original vector, thus the point with no tight inequalities is not extreme. Let there be two tight inequalities with indices $s$ and $s+k$ and let all their intermediate inequalities be loose. There exist two possible cases:

1. $\exists i \in \{1, \ldots, k - 1\}$, with $x_{s+i} > 0$. We make the following transformation that increases the defender’s NE payoff. We reduce $x_{s+i}$ by a small amount $\epsilon > 0$ and
increase $x_{s+i+1}$ by the same amount, maintaining the same norm $\|x\|$. All the inequalities before and after the one with index $(s+i)$ are intact, while the previously loose inequality $(s+i)$ is now tight. For the new vector $\hat{x}$ it holds that $\mu'\hat{x} < \mu'x$, since $\mu$ is a vector with decreasing values and we have shifted some weight from $x_{s+i}$ to $x_{s+i+1}$. By Lemma 11, the new point corresponds to a defender NE strategy with a higher NE payoff. We continue the above procedure until there are no loose inequalities between the initial tight ones, or until $x_{s+i} = 0, \forall i \in \{1, \ldots, k-1\}$.

2. $x_{s+i} = 0, \forall i \in \{1, \ldots, k-1\}$. Subtracting the first tight inequality (of index $s$) from any loose inequality of index $s+i$, with $i \in \{1, \ldots, k-1\}$, we get

$$
\Delta_1 \delta(s) \cdot (x_0 + \ldots + x_s) + \Delta_1 M(s) \|x\| > 0
$$

$$
\vdots
$$

$$
\Delta_{k-1} \delta(s) \cdot (x_0 + \ldots + x_s) + \Delta_{k-1} M(s) \|x\| > 0.
$$

Similarly, subtracting the last tight inequality $(s+k)$ from all the loose inequalities of index $s+i, \forall i \in \{1, \ldots, k-1\}$, we get

$$
\Delta_{k-1} \delta(s+1) \cdot (x_0 + \ldots + x_s) + \Delta_{k-1} M(s+1) \|x\| < -\delta(s+k)x_{s+k}
$$

$$
\vdots
$$

$$
\Delta_1 \delta(s+k-1) \cdot (x_0 + \ldots + x_s) + \Delta_1 M(s+k-1) \|x\| < -\delta(s+k)x_{s+k}.
$$

Under condition 1, the set of equations (3.6) and (3.7) cannot be satisfied simultaneously. Indeed, the last equation of (3.7) gives $0 > \Delta_1 \delta(s+k-1) \cdot (x_0 + \ldots + x_s) + \Delta_1 M(s+k-1) \|x\| > \Delta_1 \delta(s) \cdot (x_0 + \ldots + x_s) + \Delta_1 M(s) \|x\|$, which contradicts the first equation of (3.6).
Let $s$, $f$ be the indices of the first and last tight inequalities (of the contiguous block of tight ones) respectively.

**Lemma 13.** Under condition 1, an extreme point $x$ that corresponds to a defender NE strategy $\beta$ has zeros before $s$ and after $f + 1$, i.e.,

$$x_i = 0, \forall i \in \{0, \ldots, s - 1\} \cup \{f + 2, \ldots, N + 1\}.$$

**Proof.** We first show that $x_i = 0, \forall i < s$. If $\exists i \in \{0, \ldots, s - 1\}$, s.t. $x_i > 0$, we reduce $x_i$ to $\hat{x}_i$ until either $\hat{x}_i = 0$ or $i$th inequality is tight, and increase $x_{i+1}$ by the same amount. We maintain $\|x\|$ constant, and in case that $x_{i+1} > 0$ we get one more tight constraint. Thus the original point is not extreme, as we can find another point whose tight constraints is a strict superset of those of the original. In case that $x_{i+1} = 0$, the new $\hat{x}$ corresponds to a defender NE strategy with a higher defender NE payoff. We now show that $x_i = 0, \forall i > f + 1$. If $\exists i \in \{f + 2, \ldots, N + 1\}$, s.t. $x_i > 0$, we reduce $x_i$ until $\hat{x}_i = 0$ and increase $x_{f+1}$ by the same amount. We again keep the norm $\|x\|$ constant but $\hat{x}$ has one more tight constraint, thus $x$ was not extreme. \qed

**Lemma 14.** In any Nash equilibrium, under conditions 1 and 2,

1. $f = N$, when $D$ is non increasing.

2. $f = N$ or $s = f$, when $D$ is increasing.
Proof. Suppose that \( f < N \). Then the inequality of index \((f + 1)\) exists, is loose and all positivity constraints are satisfied. Subtracting the tight inequality of index \( f \) from the loose inequality of index \((f + 1)\), we get

\[
x_{f+1} > \frac{[D(f) - D(f + 1)] \cdot \|x\|}{\delta(f)} \triangleq C.
\]  

(3.8)

1. If \( D \) is non increasing, since \( \delta \) is positive, \( C \geq 0 \) and \( x_{f+1} > 0 \). We consider the following transformation

\[
\hat{x}_i = \begin{cases} 
 x_i & \text{for } i \in \{0, \ldots, f\} \cup \{f + 3, \ldots, N + 1\} \\
 C & \text{for } i = f + 1 \\
 x_{f+1} - C & \text{for } i = f + 2.
\end{cases}
\]

(3.9)

With the above transformation we get

\[
\mu' (\hat{x} - x) = \mu_{f+1} \cdot (\hat{x}_{f+1} - x_{f+1}) + \mu_{f+2} \cdot (\hat{x}_{f+2} - x_{f+2})
\]

\[
= \mu_{f+1} \cdot (C - x_{f+1}) + \mu_{f+2} \cdot (x_{f+1} - C - 0)
\]

\[
= (x_{f+1} - C) \cdot (\mu_{f+2} - \mu_{f+1})
\]

\[
< 0,
\]

since \( x_{f+2} = 0, x_{f+1} > C \), and \( \mu \) is a strictly decreasing vector \( (\mu_{f+2} < \mu_{f+1}) \). Hence, for the new point \( \hat{x} \), \( \|\hat{x}\| = \|x\| \), but \( \mu' \hat{x} < \mu' x \). By Lemma 11 point \( \hat{x} \) corresponds to a defender NE strategy with a higher defender NE payoff. We can continue making the above transformation until \( f = N \).

2. If \( D \) is increasing, then \( C < 0 \) and \( x_{f+1} \geq 0 \). If \( x_{f+1} > 0 \), while \( f < N \) we can shift a small amount \( \epsilon \) from \( x_{f+1} \) to \( x_{f+2} \), keeping the same norm but getting a higher
defender NE payoff. We keep making the above transformation until \( f = N \). If \( x_{f+1} = 0 \), then \( \|x\| = x_s + \ldots + x_f \). Subtracting the two tight inequalities (s) and (f) and since \( D(H) \) is an increasing function,

\[
x_s = \frac{[D(f) - M(s)] \cdot \|x\|}{\delta(s)} > \frac{[D(s) - M(s)] \cdot \|x\|}{\delta(s)} = \frac{\delta(s) \cdot \|x\|}{\delta(s)} = \|x\|,
\]

or \( x_s > \|x\| \). Contradiction, unless \( s = f \).

Lemma 15. Amongst different defender mixed strategies \( \beta \) with the same component \( \beta_{N+1} \), the detection cost against a best responding attacker is the same, under conditions 1 and 2.

Proof. By Lemma 14, under conditions 1 and 2, \( f = N \) or we have pure strategies NE. By Lemma 10, the points with the same norm \( \|x\| \) correspond to defender strategies with the same detection cost (\( \min[\Lambda \beta] = 1/\|x\| \)). Scaling the last tight inequality \( N \) with the norm and since \( \beta \) is a distribution, we get \( \delta(N) (1 - \beta_{N+1}) + M(N) = \frac{1}{\|x\|} \). Thus for the same \( \beta_{N+1} \), the norm is the same, which results in the same detection cost against a best responding attacker.

Lemma 16. Under conditions 1 and 2, amongst defender mixed strategies with different \( s \) and same \( \beta_{N+1} \), the higher defender NE payoff is maximum when \( s \) is maximal.

Proof. Let \( \beta, \hat{\beta} \) be two different defender NE strategies with \( \beta_{N+1} = \hat{\beta}_{N+1} \). By Lemma 15, since \( \beta_{N+1} \) is the same for both vectors, the cost of detection is the same. Let \( \hat{s} = s - 1 \). We will show that the false alarm penalty for the largest index \( s \) is larger, i.e., \( \mu' \cdot (\beta_s + \ldots + \|x\|, \)
Subtracting the tight inequalities \( N \) and \((N - 1), \) results in \( \beta_N = \hat{\beta}_N. \) Similarly, iteratively subtracting the tight inequalities \((s + k)\) and \((s + k + 1), \)
\[ \forall k \in \{1, \ldots, N - s - 1\} \text{ results in } \beta_{s+k} = \hat{\beta}_{s+k}. \] By Lemma 13, \( \beta_{s-1} = \ldots = \beta_0 = 0 \) and \( \hat{\beta}_{s-2} = \ldots = \hat{\beta}_0 = 0. \) Thus the two different NE strategies differ only in \( \beta_{s-1}, \) and \( \beta_s. \) The remaining weight is the same for both vectors \( (\beta_{s-1} + \beta_s = \hat{\beta}_{s-1} + \hat{\beta}_s = 1 - \sum_{i=s+1}^{N+1} \beta_i). \) In the case of the vector \( \hat{\beta}, \) this weight is divided into two different components \( (\hat{\beta}_{s-1} \text{ and } \hat{\beta}_s) \) while in the case of \( \beta \) it is all assigned into the component with index \( s. \) Since \( \mu_s < \mu_{s-1}, \) the vector \( \beta \) with the largest index of the first tight inequality \( s \) will provide a smaller false alarm cost, and hence a higher defender NE payoff.

We now give the proof of the main theorem in this chapter.

Proof of Theorem 3. Depending on the nature of the cost functions, there are two potential constructions for the defender NE strategy \( \beta. \) We select the one that yields the highest defender NE payoff.

1. Mixed strategies NE with \( f = N. \) By Lemma 16, defender strategies \( \beta \) with the same \( \beta_{N+1} \) yield the highest defender NE payoff when \( s \) is maximal. Thus, we need to find the largest possible \( s \) such that the inequalities 0 through \((s - 1)\) are loose and \( x_0 = \ldots = x_{s-1} = 0. \) Since we are in mixed NE strategies, there exist at least two tight inequalities. Starting from the last tight inequality \( N \) and subtracting the next tight inequality \( N - 1, \) we compute \( \beta_N. \) In general, subtracting the \( i, \) \((i - 1)\) inequalities, we compute \( \beta_i = \frac{D(i) - D(i+1) [\delta(i) - \delta(i-1)]}{\delta(i-1)} \sum_{i=s+1}^{N+1} \beta_i}. \) In every step we check whether the previous inequality \((s - 1)\) can be loose. If this is possible, then we assign all the remaining weight to \( \beta_s \) \( (\beta_s = 1 - \sum_{i=s+1}^{N+1} \beta_i). \) Since the block of tight
inequalities that ranges from $s$ through $N$ (integers) is unique, only a certain number of selections on $\beta_{N+1}$ will produce valid vectors $\beta$ (with unit norm and nonnegative weights). Thus we need to solve the following equations

$$
\delta(s - 1) \cdot 0 + M(s - 1) > 1/\|x\|
$$

$$
\delta(s) \cdot \beta_s + M(s) = 1/\|x\|
$$

$$
\vdots
$$

$$
\delta(N) \cdot (1 - \beta_{N+1}) + M(N) = 1/\|x\|.
$$

Subtracting the tight inequality $N$ from the $(s-1)$ loose inequality we get $M(s-1) > \delta(N) \cdot (1 - \beta_{N+1}) + M(N)$. Solving for the integer $s$, we compute the increments of $\beta_{N+1}$ that give a valid distribution $\beta$.

2. Pure NE with $s = f$. This case implies that when $D$ is an increasing function, a pure defender strategy maximizes the defender NE payoff. For each selection of $s$ in $\{0, \ldots, N\}$, we compute the defender’s NE payoff of the resulting strategy $\beta (\beta_s = 1)$, and select the strategy that maximizes the defender’s NE payoff.

   \[\square\]

For the attacker, the same properties as shown in Chapter 2 apply for $\beta$ the resulting equilibrium defender strategy derived by Theorem 3. For completeness, Algorithm 4 gives the attacker’s NE strategy form.

### 3.3 Evaluation with model examples

In this section, we present two characteristic examples of the above general problem and evaluate them in terms of the expected defender NE payoff.
Algorithm 4: Compute-a(s)

1 $\alpha_0$ to $\alpha_{s-1} \leftarrow 0$;

2 for $i \leftarrow s + 1$ to $N - 1$ do

3 \hspace{1em} $\alpha_i \leftarrow \frac{\mu_i}{c_d}$;

4 if $\beta_s > 0$ then

5 \hspace{1em} $\alpha_s \leftarrow \frac{\mu_s}{c_d}$;

6 else if $\beta_{N+1} > 0$ then

7 \hspace{1em} $\alpha_N \leftarrow \frac{\mu_N}{c_d}$;

8 \hspace{1em} $\alpha_s \leftarrow 1 - \sum_{i=s+1}^{N} \alpha_i$;

9 else if $\beta_s = \beta_{N+1} = 0$ then

10 \hspace{1em} any $\alpha_s \in [0, 1 - \sum_{i=s+1}^{N-1} \alpha_i]$;

11 $\alpha_N \leftarrow 1 - \sum_{i=s}^{N-1} \alpha_i$;
3.3.1 Example Model 1

In the first model, the spy’s cost function in case of detection is \( D(H) = c_d - H \cdot c_a \).

There is a constant cost \( c_d \) associated with the detection and a benefit proportional to the number of attacks \( H \). In case of missed detection, the spy gets the benefit from the attacks, without suffering from the detection cost, thus \( M(H) = -H \cdot c_a \), where \( c_a \) is the cost associated with a single FS attack. The spy cost is

\[
J_A(T, H) = c_d \cdot 1_{T \leq H} - c_a \cdot H.
\]

The defender’s expected reward function depends on the true type of the attacker and following the general model analysis and is given by

\[
U_D(T, H) = J_A(T, H) - \mu(T),
\]

where \( \mu(T) = \frac{1-p}{p} \cdot c_a \cdot \phi(T) \). All lemmas that were proved in Section 3.2 hold since conditions 1 and 2 hold. Note that \( M(H) = -H \cdot c_a \) is a decreasing function with respect to \( H \) and \( \delta(H) = c_d \) is constant. Thus there is a contiguous block of tight inequalities starting at index \( s \) and finishing at index \( N \) with \( x_i = 0, \forall i \in \{0, \ldots, s - 1\} \), or we have pure NE.

Furthermore, the defender’s NE strategy exists amongst the two forms in Table 3.1 (convex optimization of solutions of these two forms). This is a special case of the game defined in Chapter 2. Since the cost from each attack on the FS hit is constant and equal \( c_a \), a threshold on the attack reward (measured by the quantity \( H \cdot c_a \) is a direct equivalent to a threshold on the number of FS hits).
Table 3.1: Defender’s strategy in NE ($\beta_m = c_a/c_d$)

<table>
<thead>
<tr>
<th>#</th>
<th>$\ldots$</th>
<th>$\beta_s$</th>
<th>$\beta_{s+1}$</th>
<th>$\ldots$</th>
<th>$\beta_N$</th>
<th>$\beta_{N+1}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>0</td>
<td>0</td>
<td>$\beta_m$</td>
<td>$\beta_m$</td>
<td>$\beta_m$</td>
<td>$1 - (N-s)\beta_m$</td>
</tr>
<tr>
<td>2.</td>
<td>0</td>
<td>$1 - (N-s)\beta_m$</td>
<td>$\beta_m$</td>
<td>$\beta_m$</td>
<td>$\beta_m$</td>
<td>0</td>
</tr>
</tbody>
</table>

3.3.2 Example Model 2

In this second variation of the model, we assume that the defender maintains some logs on the type of occurred attacks. When a spy is detected, the defender has the appropriate tools to investigate the attacker’s behavior. This way, the defender has the opportunity to learn about the spy’s true intentions (which specific target/information he seeks to extract from the file server), his location or identity and his future attack pattern, in case he is not immediately expelled.

Each of the $H$ FS hits now gives the spy a benefit of $c_a$ only if he evades detection. In case he is correctly identified, each FS attack yields a cost of $c_a$ for the spy, as they reveal the intentions of the spy. Thus $D(H) = c_d + H \cdot c_a$, and $M(H) = -H \cdot c_a$, giving the spy a cost function of

$$J_A(T, H) = (c_d + 2c_a \cdot H) \cdot 1_{T \leq H} - c_a \cdot H.$$  

Following the analysis for the general model, the defender payoff function is

$$U_D(T, H) = J_A(T, H) - \mu(T).$$

All lemmas that were proved in Section 3.2 hold since conditions 1 and 2 hold. Note that $M(H) = -c_a \cdot H$ is a decreasing function with respect to $H$ and $\delta(H) = D(H) - M(H) = c_d + 2Hc_a$ is increasing with respect to $H$. Thus there is a contiguous block of tight inequalities starting at index $s$ and finishing at index $N$ with $x_i = 0$, $\forall i \in \{0, \ldots, s-1\}$,
or we have pure NE.

**Defender’s strategy in example Model 2.** After subtracting the two tight inequalities $N$ and $N - 1$, we get $\beta_{N+1} \geq 1/2$, because the existence of a tight inequality with index $N - 1$ also suggests that $\beta_{N-1} \geq 0$. Thus in either case, it must be that $\beta_{N+1} \geq 1/2$. The upper bound for $\beta$ is 1. But since the index of the first tight inequality is an integer, only certain values of $\beta_{N+1}$ result in an optimal $s$, which is also an integer.

By Theorem 3, given a certain $\beta_{N+1}$ for the defender NE strategy, we need to find the largest possible $s$ such that inequality $s$ is tight and $(s - 1)^{th}$ is loose, with $\beta_0 = \ldots = \beta_{s-1} = 0$. Subtracting the tight inequality $N$ from the loose inequality $s - 1$, we get $s \leq \frac{(c_a-Nc_a)+(c_d+2Nc_a)\beta_{N+1}-c_d}{c_a}$. Since $s$ must be an integer, $\beta_{N+1} = \frac{(N-1+k)c_a+c_d}{c_d+2Nc_a}$, with $k$ integer. Thus the search over the optimal $\beta_{N+1}$ has linear complexity with respect to $N$, with $\beta^{\text{min}}_{N+1} = 1/2$, $\beta^{\text{max}}_{N+1} = 1$ and $\text{step} = \frac{(N-1+k)c_a+c_d}{c_d+2Nc_a}$. Alternatively, solving for the integer $k$, we get $k^{\text{min}} = 1 - N - c_d/c_a$ and $k^{\text{max}} = N + 1$.

### 3.4 Parameter effects in the Game

The two previously presented models have an essential difference: While in the first model, the spy benefits from the FS attacks regardless of the defender’s classification decision, in the second model, the spy benefits from the FS attacks only when he is misclassified. The assumption under the second model is that the defender has invested in forensic techniques, and is able to identify, preserve and analyze attacks within the network. This way, each FS attack reveals information about the identity and the intention of the attacker.

Computer forensics is a costly investment, thus the defender needs to decide under
which circumstances she should develop such tools. By comparing the two above models, and essentially the defender’s expected payoff in NE, we extract key insights on the expected gain from the forensics. The crucial parameters are prevalent in both models, like \( p \), \( c_{fa} \), \( \theta_0 \), \( c_a \), \( c_d \). The critical difference is that the spy’s cost function in case of detection \( D(H) \) is decreasing in the first model and increasing in the second model, with respect to \( H \).

Some network parameters are correlated in sets of combinations, in the sense that a change in any element on the set alters the Nash equilibrium and payoffs of the players toward the same direction. For instance, looking at the defender’s payoff function for the first model, we observe that a change in \( p \) or \( c_{fa} \) affects the false alarm penalty, thus changing \( p \) and keeping \( c_{fa} \) constant will provide us with the same implications as if we kept \( p \) the same and changed \( c_a \). Thus it makes sense to investigate the impact of only a subset of the parameters.

### 3.4.1 Effect of the probability of the spy \( p \) and false alarm cost \( c_{fa} \)

We expect that when \( p \) is small, the defender will suffer a small cost from potential FS attacks. As \( p \) increases it becomes more difficult to distinguish between spy and spammer and the defender’s payoff will be decreasing. When \( p \) becomes larger, the defender will classify him correctly and receive a higher payoff.

Indeed, in Figure 3.1(a) we observe two areas of different behavior. When \( p < 0.5 \), the defender payoff function is decreasing whereas it is increasing as the spy’s probability reaches \( p = 0.5 \). We also observe that as \( p \) increases, the spy’s attack policy becomes more aggressive and the defender reduces her threshold to catch the more-probable-to-exist spy. In Figure 3.1(b) we note that as \( p \) increases, the benefit of investing in forensics (and
employ Model 2) is an increasing function on $p$. We note here that the depicted values for the defender payoff are the unscaled and unshifted initial payoffs, as expressed in (3.1).

### 3.4.2 Effect of the detection cost $c_d$, the classification window $N$ and single FS attack cost $c_a$

In Model 1, when the cost of detection $c_d$ is small compared with the maximum achievable gain from the FS attacks ($N \cdot c_a$), the spy does not care about getting detected and is attacking with his maximum allowed strength ($N$ times). On the contrary, in Model 2, where the spy suffers a cost proportional to his attack aggressiveness in case of detection, the spy is more conservative with his attacks. This difference is depicted in Figure 3.2(a).

As we can see, in Model 1, the cost of detection is so small, than the attacker always attacks $N$ times. On the other hand, the defender selects a threshold equal to the pure strategy of the spy and detects him. If the defender selected $T = N+1$ or $T = N-1$ instead of $T = N$ as his classification threshold, she would miss the spy and would have smaller payoff due to the increased false alarm, respectively. In the second model, though, the spy takes into consideration the potential benefit his FS attacks would give the defender. The spy is less aggressive, and attacks fewer times. Other parameters of the game are $N = 5$, $c_d = 1$, $c_a = 1$, $p = 0.1$, $c_{fa} = 10$, and $\theta_0 = 0.1$. We also note here that the spy’s strategy is a weighted truncated binomial distribution. Every defender’s strategy in his NE support gives the defender the same payoff. Thus the difference in the false alarm penalty for the different thresholds matches the difference in the misdetection cost. For instance

$$\Pr\{H = 3\} = \frac{(1-p)\cdot c_{fa} \cdot [\phi(3) - \phi(4)]}{c_d + 2.3 \cdot c_{fa}} = 0.1041.$$  

When $c_a$ is small, (or else when $c_d$ is most important than $N \cdot c_a$), we observe that
(a) NE defender payoff first decreases and then increases as \( p \) increases.

(b) The defender’s equilibrium payoff is higher in Model 2.

Figure 3.1: As \( p \) increases, the NE defender gain with Model 2 increases.
Figure 3.2: For a small detection cost, the spy is more conservative in Model 2.

the two models result in the same strategies for the two players (Figure 3.2(b)). Indeed, when the spy expects not to reveal a lot of information to the defender if he gets detected,
he will act as if there was not risk (as in Model 1). Thus, when the defender expects to lose little from the FS hits, she will avoid investing in forensics to learn more about the intentions of the spy.

3.4.3 Effect of the spammer’s distribution parameter $\theta_0$

In these two models we have assumed a specific distribution on the FS attacks for the naive player, i.e., the spammer. Each time slot (period) of the available $N$ time slots, the spammer attacks the FS with a frequency of $\theta_0$. In the case that $\theta_0$ is small (the spammer is mostly interested in attacking the MS instead of FS) the task of the defender to differentiate between the two types of attackers becomes easier.

On the contrary, if the spammer is attacking with a high $\theta_0$ each period, then the defender is hurt from the false alarms, since she will be confused from the large number of FS hits and will classify the attacker as spy. We can see this difference in the defender NE payoff as $\theta_0$ increases.

In Figure 3.3 we see the effect of the spammer’s strategy, essentially $\theta_0$, on the two players’ NE strategies. In both models, as $\theta_0$ increases, the spy becomes more aggressive (to imitate the spammer’s behavior). As $\theta_0$ increases, the spammer attacks the FS more frequently, and it is more difficult for the defender to distinguish the two types of attacker. The spy then exploits this uncertainty to increase his payoff (by attacking more times). When $\theta_0$ is small, the defender sets the threshold low for spy classification. As $\theta_0$ increases, his false alarm penalty gets smaller and the defender assigns a larger weight to the “always classify as a spammer” strategy.

In Figure 3.4 we see the effect of the $\theta_0$ on the defender’s payoff for the two models,
Figure 3.3: Effect of $\theta_0$ on the players’ NE strategies ($N = 5, c_a = 1, c_d = c_{fa} = 10, p = 0.3$). The bar left and right of numeral represents the defender and the spy respectively.

for various values of the prior probability of the spy $p$. In Model 1, depicted in Figure 3.4(a), 3.4(b), we observe that as $\theta_0$ increases, the defender’s NE payoff decreases for any value of $p$, because higher $\theta_0$ signifies a higher false alarm penalty for the defender. In contrast, the second model depicted in Figure 3.4(b), the above rule applies only for the ranges of $\theta_0$ below $\theta_0 = 0.5$. For $p > 0.5$, the defender will always select the same pure strategy, that yields the same payoff.
Figure 3.4: The defender’s NE payoff decreases as $\theta_0$ increases for all values of $p$ for Model 1, but only for $p < 0.5$ for Model 2.

3.5 Related work

Cybersecurity is important to businesses and individuals. The number of cyber attacks and threats has increased during the past few years, resulting in lost productivity, reduced revenue, and bad reputation for the associated businesses. Different kinds of attacks (e.g., internal unintentional actions and external malicious ones) should be treated differently and organizations need the security intelligence to respond to all threats rapidly. Since only less than half of organizations are currently pursuing security issues, there is still room for improvement. Our work contributes to the understanding of the interaction between network operators and potential attackers.

In almost every network security situation, the administrator of a network (defender) has limited resources in energy and time. The defender needs to distinguish between different types of attackers (attacker or non-attacker) and decide whether to take actions or not. For example, an attack on a mail server by a non-attacker (causing at most network
congestion) should be treated differently than an attack on a file server (possibly involving identity theft). Therefore, the defender should employ various statistical tests, based on the observed number of file and mail server attacks and decide upon the true type of the attacker. Knowing that a defender is trying to classify attackers, the strategic attacker is likely to change the way he attacks in order to make it more difficult to be classified as an attacker. In this chapter, we analyzes a simple model of such a classification game and extracted key insights.

There exists a growing body of works on the topic of intrusion detection. In a series of papers [AB03, AB04], Alpcan and Başar present a game-theoretic analysis of a security game between an attacker and an intrusion detection system in different scenarios, both in finite and continuous-kernel versions. Our game-theoretic framework focuses on attacker classification, rather than intrusion detection. In the presence of a non-strategic player who is represented with a fixed and known probability distribution, the defender’s task of distinguishing the true type of the attacker becomes more challenging. It is also interesting to see how the non-strategic non-attacker influences the attacker’s strategy.

Bao, Kreidl, and Musacchio [BKM11] also consider an intruder classification game, in which the sequence of attacks is taken into account. While their model has many similarities with ours, we focus on less complex (but still realistic) payoff functions that allow us to go one step further than simulations and analyze the structure of the Nash equilibria.

Gueye, Walrand, and Anantharam [Gue11, Ass11] have investigated the structure of the Nash equilibria in a network topology game where two adversaries select which links to attack and defend. They consider a special case of nonzero-sum games, in which the
different term in the players’ payoffs is controlled only by the one player. In these games, one player optimizes his payoff against the other who has optimized his payoff as well. Such games are easier to analyze than general nonzero-sum games, and they give interesting insights on the strategies of the two players. Our work is using a similar payoff formulation in a different setting: the defender selects a threshold on file server attacks (not a set of links to defend) and there are two different types of attackers.

To the best of our knowledge, [DDM⁺04] is the most relevant work to ours. The authors address the problem of classifying a malicious intruder in the presence of an innocent user. The malicious intruder can perturb his behavior to confuse the classifier and evade detection. But, their work focuses only on one iteration of the game and how each player can once adjust his strategy to optimize his expected payoff, given the optimal strategy of the other player. On the contrary, we provide an algorithm to find the Nash equilibria of the game.

3.6 Conclusions

In this chapter we investigated the adversarial classification game presented in Chapter 2 from a different angle. By formulating a simple one-shot game between an attacker and a defender, we showed which results in Chapter 2 hold instantly based on the formulation of the game and which depend on the payoff functions of the players. Our experimental results coincide with the theoretically expected ones: The structure of the cost matrix of the attacker leads to only two forms of defender’s strategies in NE (or a convex combination of them). There is a relationship between the non-attacker’s distribution and
the attacker’s NE strategy. Furthermore, the defender NE strategy includes a contiguous set of thresholds that always include large values. If the parameters of the game satisfy a certain condition, the defender is uniformly randomizing among a set of thresholds. We investigated two specific game models: Model 1 is a simpler game, where the spy benefits from his attacks, regardless of the defender’s classification decision. In Model 2, the defender is equipped with forensic tools and the spy only benefits from his attacks upon a misclassification. By analyzing these two games, we extracted insights about when the defender should invest in forensics and how the strategies of the two players in NE are affected by the various control parameters of the game.
Part II

Click Fraud Classification
The online advertising market typically involves different classes of players: publishers, ad networks, and advertisers. Advertisers have products to advertise and design the ads. Publishers own websites that receive traffic and at which the ads can be placed. The ad networks act as intermediaries between publishers and advertisers: they match the publishers with different ads, and charge the advertisers for a certain portion of clicks, the ones the ad networks deem as “valid”.

Click fraud (or click spam) has been a serious problem in the online advertising market. By click spam, we define the act of clicking on an ad without any interest to see the ad. When such clicks are counted as “valid” by the ad network, the advertiser pays for a useless click, and the publisher is rewarded for generating it with a certain fraction of the generated revenue called revenue share. Thus there exist incentives for publishers to inflate...
click numbers. Ad networks have an incentive to identify and filter out fraudulent clicks in order to deliver a more valuable service to advertisers, who are their customers. On the other hand, ad networks do receive revenue for fraudulent clicks, which creates an incentive in the opposite direction to fight fraud less.

We are interested in a two-dimensional competition between ad networks — they simultaneously compete to maximize their revenue on filtering aggressiveness and revenue share given to the publishers. Our model is admittedly simplified, yet still captures those aspects that are interesting when fighting click fraud. To the best of our knowledge, we are the first to investigate the Nash equilibria in games of such a setting. Will one network choose to be tolerant with filtering and compensate by giving a bigger share to publishers, while the other network is more aggressive and gives a smaller share? Will the networks even care fighting click fraud? How are the preferences of the publishers affect the decisions of the ad networks? In which direction of the two aforementioned ones will the competition be fiercer?

To address the above questions, our paper is organized as follows. Section 4.1 describes the underlying economic model. Section 4.2 provides the Nash equilibrium analysis. Section 4.3 presents numerical experiments that show how the quality of the ad networks and the distribution on the publishers’ preferences affect the revenue sharing strategies of the two ad networks. In Section 4.4 we point to related work in click fraud and cross-platform competition. Finally, we conclude with the main results and insights in Section 4.5. (This chapter draws on material from Dritsoula and Musacchio [DM13].)
Figure 4.1: Publishers’ preferences are uniformly distributed between AN$_1$ and AN$_2$.

4.1 Economic Model

We consider a one-shot game between two ad networks, called AN$_1$ and AN$_2$. The two ad networks compete to receive clicks (display ads) from the publishers. The publishers are uniformly distributed along a line of length 1 between the two ad networks, as shown in Figure 4.1. Preferences are driven by anticipated click volume. A publisher could believe that one ad network would be better at placing relevant ads for the type of content the publisher offers and the demographics of the visitors on the publisher’s site. In this context, we assume that AN$_1$ is “preferred” by some publishers and AN$_2$ is “preferred” by others.

Each ad network, AN$_i$, simultaneously decides how aggressively to filter out invalid clicks, and what fraction of the revenue the publishers will get. After both ad networks announce their decisions:

1. Publishers choose between the ad networks, according to their preferences and the revenue they get.

2. Ad networks mark a fraction of clicks as valid.

3. Advertisers adjust their bids in ad auctions to realize a fixed return on investment – based on the anticipated ratio of truly valid clicks to clicks that are marked valid by the ad network.
4. Advertisers pay for the clicks marked as valid.

Our goal is to compute how aggressive ad networks will be, what fraction of their revenue will be distributed to the publishers at equilibrium, and how the market of publishers will react.

4.1.1 Ad networks

As in [MWGM08], we assume that ad networks can identify fraudulent (invalid) clicks with a receiver operating characteristic (ROC) curve of the form shown in Figure 4.2. Ad network $i$ is endowed with a type $\alpha_i$ that characterizes the ROC curve of the click fraud filtering technology used. Each ad network’s inherent type is the effectiveness $\alpha_i \in [0, 1]$ of their filtering, while their strategic decisions involve the aggressiveness $x_i \in [0, 1]$ and the revenue share $h_i \in [0, 1]$. We define aggressiveness $x_i$ to be the fraction of valid clicks classified as invalid. An ad network that is more aggressively identifying fraud would choose a higher value $x_i$. Given $x_i$ and $\alpha_i$, each ad network marks a fraction $(1 - x_i)$ of valid clicks as valid, and a fraction $(1 - x_i^{\alpha_i})$ of invalid clicks as valid, as shown in Table 4.1. If $\alpha_1 < \alpha_2$, AN$_1$ is more effective.

$$
\begin{array}{|c|c|c|c|}
\hline
\text{Variable} & \text{Classification} & \text{Truth} & \text{Result} \\
\hline
x_i & \text{"invalid"} & \text{valid} & \text{False positive} \\
1 - x_i & \text{"valid"} & \text{valid} & \text{True negative} \\
x_i^{\alpha_i} & \text{"invalid"} & \text{invalid} & \text{True positive} \\
1 - x_i^{\alpha_i} & \text{"valid"} & \text{invalid} & \text{False negative} \\
\hline
\end{array}
$$

Table 4.1: Ad networks make mistakes when filtering out invalid clicks.
Figure 4.2: ROC curve: If the ad networks are willing to tolerate a false positive rate of $x_i$, they can achieve a true positive rate of $x_i^{\alpha_i}$.

The goal of the ad networks is to maximize their revenue

$$U_{AN}^i(x_i, h_i) = (1 - h_i) \cdot N_{ri} \cdot c,$$

where $h_i$ is the revenue share given to the publishers, $c$ is the price per each ad click, and $N_{ri}$ is the number of clicks marked by ad network $i$ as valid or “real” (the ones the advertiser is charged for). $N_{ri}$ is a function of both the quality $\alpha_i$ of the classification algorithms and the aggressiveness $x_i$ selected by the ad network and is given by

$$N_{ri} = (1 - x_i) \cdot r \cdot V_i + (1 - x_i^{\alpha_i}) \cdot (1 - r) \cdot V_i,$$

where $r$ is the fraction of total clicks that are real or valid, and $V_i$ is the volume of the clicks received by each network. The volume $V_i$ depends on how the market of publishers is split.

4.1.2 Publishers

Depending on the quality of traffic and clicks generated, publishers can either be classified as good (legitimate) or bad (fraudulent). The information is asymmetric: publishers know if they are good or bad, but the ad networks do not. Therefore ad networks need to develop classification algorithms. Also, bad publishers that get discovered as being
bad can easily change identities.

We assume that all clicks generated on good publishers’ websites are valid, and that all clicks generated on bad publishers’ websites are invalid. The assumption that good publishers have only good clicks is extreme, but it is a convenient way to model the fact that good publishers will have a much larger fraction of good clicks than fraudulent ones. It would be cumbersome to add more parameters, like the fraction of bad clicks for good publishers and the fraction of bad clicks for bad publishers for instance.

We consider that publishers have different preferences for ad networks. Publishers are uniformly distributed along a line $\theta \in [0, 1]$. The point of division between the regions served by the two ad networks (denoted by $\theta^*$) is determined by the condition that at this position the publishers are indifferent between AN$_1$ and AN$_2$. Equating the delivered publishers’ revenues we have

$$h_1 \Phi_1(x_1)[(1-\theta^*)(1-g) + g] = h_2 \Phi_2(x_2)[\theta^*(1-g) + g]$$

(4.1)

where $\Phi_i(x_i) = \frac{r \cdot (1-x_i)}{r \cdot (1-x_i) + (1-r) \cdot (1-x_i^{\alpha_i})}$ is the fraction of charged clicks that are valid, which also depends on the quality of each network $\alpha_i$.

The parameter $g \in [0, 1]$ is the degree of platform homogeneity. It reflects the importance of the preferences of the publishers with respect to the prices. When $g$ is small $(g = 0)$, preferences are more important, while when $g$ is large $(g = 1)$, prices have a greater impact on the decision of the publishers. In the subsequent analysis (Section 4.2), we investigate both extreme cases and highlight how a different model about the publishers affects the equilibrium strategies of the ad networks. Solving Eq. (4.1) for $\theta^*$ we find
\[
\theta^* = \frac{h_1\Phi_1(x_1) - gh_2\Phi_2(x_2)}{(1-g)[h_1\Phi_1(x_1) + h_2\Phi_2(x_2)]}. 
\tag{4.2}
\]

4.1.3 Advertisers

As in [MWGM08], we assume that the number of advertisers is sufficiently large and covers the number of ad positions on the publishers’ websites. This is a realistic assumption, as the advertisers are actually competing to display their ads through auctions. The advertisers adjust their bids to maintain a certain return on investment. It is arguable they would have such a strategy since they would want to invest in online advertising up until the point its return is comparable to that achieved from other forms of advertising. Depending on the quality of the clicks they pay for, they adjust their bids to account for clicks of inferior quality by a factor of
\[
\frac{r}{r(1-x_i) + (1-r)(1-x_i^{\alpha_i})}. 
\]

We avoid dealing with the auctions mechanism details and focus on the implications of the revenue sharing and aggressiveness levels selected by the ad networks. Therefore, we assume that the advertisers’ bid is the price per click \(c\), multiplied by the adjustment factor.

4.2 Equilibria

Following the previous analysis, the profits of the ad networks are
\[
J_1(x_1, h_1) = (1 - h_1) \cdot r \cdot c \cdot V \left( \frac{1}{2} - \frac{\theta^*}{2} \right), \tag{4.3}
\]
\[
J_2(x_2, h_2) = (1 - h_2) \cdot r \cdot c \cdot V \cdot \left( \frac{\theta^*}{2} - \frac{\theta^*}{2} \right), \tag{4.4}
\]
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha_i$</td>
<td>quality of ad network $i$</td>
</tr>
<tr>
<td>$h_i$</td>
<td>revenue share given to publisher $i$</td>
</tr>
<tr>
<td>$x_i$</td>
<td>aggressiveness of ad network $i$</td>
</tr>
<tr>
<td>$r$</td>
<td>fraction of clicks that are valid (“real”)</td>
</tr>
<tr>
<td>$c$</td>
<td>price per click</td>
</tr>
<tr>
<td>$V_i$</td>
<td>volume of clicks for ad network $i$</td>
</tr>
<tr>
<td>$N_i^r$</td>
<td>number of ad network $i$’s clicks that are valid</td>
</tr>
<tr>
<td>$g$</td>
<td>degree of platform homogeneity</td>
</tr>
<tr>
<td>$\theta^*$</td>
<td>point of market segmentation</td>
</tr>
<tr>
<td>$\Phi_i(x_i, h_i)$</td>
<td>fraction of charged clicks that are valid</td>
</tr>
</tbody>
</table>

Table 4.2: List of variables introduced in Section 4.1.

where $\theta^*$ is given by Eq. (4.2).

Each ad network $i$ needs to select $x_i \in [0, 1]$, and $h_i \in [0, 1]$ to maximize the revenues given by Equations (4.3) and (4.4). We first analyze the case when the publishers are heterogeneous: preferences are more important than immediate revenues.

### 4.2.1 Heterogeneous publishers ($g = 0$).

In this case, the market share is highly determined by the preferences of the publishers. For example, when a publisher is located near (strongly prefers) AN$_1$, no matter what revenue share is given, AN$_2$ will never win over the entire market.

**Lemma 17.** *The ad networks’ payoff functions $J_i(x_i, h_i)$ are concave with respect to $x_i$, $h_i$, for $i = 1, 2$.***

**Proof sketch.** Differentiating twice Eq. (4.3), we get

$$
\frac{\partial^2 J_1}{\partial x_1^2} = -(1 - h_1)rcV \left[ \left( \frac{\partial \theta^*}{\partial x_1} \right)^2 + \theta^* \frac{\partial^2 \theta^*}{\partial x_1^2} \right].
$$
We need to show that $\frac{\partial^2 \theta^*}{\partial x_1^2} > 0$. Indeed, differentiating twice Eq. (4.3), we show that $\frac{\partial^2 \theta^*}{\partial x_1^2} > 0$. We thus show the concavity of $J_1(x_1, h_1)$ with respect to $x_1$. Similarly, we show the concavity of $J_1(x_1, h_1)$ with respect to $h_1$, and of $J_2(x_2, h_2)$ with respect to $x_2, h_2$.

**Theorem 4.** The levels of aggressiveness chosen by the ad networks at equilibrium are $x_1^* = 1$ and $x_2^* = 1$.

**Proof.** The conditions $\frac{\partial^2 J_i(x_i, h_i)}{\partial x_i^2} < 0$, sufficient for a maximum of each of the functions $J_i(x_i, h_i)$, $i = 1, 2$, are satisfied (Lemma 17). Solving $\partial J_1/\partial x_1 = 0$ for $x_1$ results to $\Phi_1(x_1)\partial \Phi_1(x_1)/\partial x_1$, or $(1 - \alpha_1)x_1^{\alpha_1} + \alpha_1x_1^{\alpha_1-1} - 1 = 0$. The unique solution to the previous equation for $x_1 \in [0, 1]$ is $x_1^* = 1$. We can similarly prove that $x_2^* = 1$.

**4.2.2 General model for publishers ($g > 0$).**

In this case, publishers are distributed between the two ad networks, not only according to their preferences, but according to the revenue share they get as well. We can similarly prove that $x_1^* = 1$ and $x_2^* = 1$.

**Lemma 18.** The payoff function of each ad network is quasi-concave with respect to the revenue share $h_i$, for $i = 1, 2$.

**Proof.** When $g > 0$, there exists a possibility that the ad network with the higher revenue share will win over all the publishers. We have already established that before such a point occurs, the payoff functions will be concave. After this inflection point, the dominant ad network will have already won over all the publishers. Since the competitor is already out of
the game, there is no benefit for the winning ad network to increase the revenue share. Thus the payoff function will be non increasing with respect to $h_i$. Overall, the payoff functions will be quasi-concave.

**Theorem 5.** *The game between the ad networks has a Nash equilibrium in pure strategies.*

*Proof sketch.* The payoff functions of the ad networks are continuous and quasi-concave in a convex compact set. Thus, there exists a Nash equilibrium in pure strategies.

### 4.3 Experiments

In this section, we gain some insights into the Nash equilibria of the game through numerical experiments. We have shown in Section 4.2 that both ad networks will select an aggressiveness level of $x = 1$ and will compete in prices (revenue share given to the publishers). Depending on the quality $\alpha_i$ of the classification algorithms of each network $i$, the estimated fraud intensity $r_i$, and the homogeneity $g$ of the publishers, the two players adjust the revenue shares $h$ they give out.

The first experiment studies the impact of the networks’ efficiency in classifying valid clicks on the prices they give to the publishers. When the ad networks are of the same quality, the ad networks’ NE prices are symmetric, as shown in Figure 4.3. On the contrary, when the ad networks are asymmetric, we observe that the inferior network (AN$_2$ in our case) selects to give more to the publishers, as seen in Figure 4.4.

We also explore the role of the publishers’ homogeneity $g$ in determining the prices in equilibrium. As $g$ increases, the publishers are more homogeneous, and the gap between the ad networks’ equilibrium prices increases (Figure 4.4). When $g > 0$, there is a chance for
Figure 4.3: When symmetric, the ad networks select symmetric equilibrium prices.

one ad network to get the entire market of publishers. Thus, we observe a fiercer competition on the revenue shares. The invalid fraction of clicks is \( r = 0.3 \), and the qualities of the ad networks are \( \alpha_1 = 0.2 \) and \( \alpha_2 = 0.7 \).

### 4.4 Related Work

Work has been done to gauge the degree of click fraud. For example, Dave et al. [DGZ12] have provided a systematic methodology to estimate and measure the click spam in ad networks. Their analysis shows that click spam is a serious problem, that tends to grow as the mobile advertising market develops.

Mungamuru, Weis, and Molina [MWGM08] have studied the effects of click fraud in the online advertising market by modeling the incentives of the different actors. The main result of their analysis is that ad networks have a net incentive to fight fraud, despite getting
revenue from fraudulent clicks that are billed to the advertisers. They have considered a market of advertisers, publishers, and ad networks, and have concluded that the ad network can gain a market advantage by aggressively combating fraud. Their analysis, though, is a one-step best response analysis and does not result in the computation of Nash equilibria. Our work investigates both the quality of classification algorithms and the revenue share as strategies for the ad networks, who compete for publishers, and we derive the Nash equilibria.

Hotelling has argued in his seminal work [Hot29] that in reality, duopoly is not fragile: a small price advantage by one firm does not capture the whole market. He showed that “some buy from one seller, some from another, in spite of moderate differences of price.” In our paper, we consider a similar “location” model, at which the publishers’ preferences are distributed uniformly on a line between two ad networks.

Kim [Kim07] shows that in the context of several applications of contemporary importance, the dispersion of consumers’ relative preferences between competing firms results
in softening market competition, and studies how the intensity of competition influences the
effects of firms’ strategies. We also establish a similar result: when the publishers become
less homogeneous in their preferences, the competition in prices becomes less fierce. Re-
searchers use the Hotelling model within models of network platform competitions [MK09],
[SJGH10].

Perlof and Salop [PS85] have showed that as users’ preferences become more in-
tense, equilibrium price increases. Similarly, changes in the utilities of the publishers by a
different multiplicative factor in our model, led to different equilibrium prices.

Comparing to other economic results on competition of identical or differentiated
products that take as given the differentiation between the products (horizontal and/or ver-
tical) and examine how price competition takes place under network effects [BS95], we study
the competition between the two ad networks in both price and degree of differentiation.

Our work is similar in spirit with the work of Musacchio and Kim [MK09]. The
strength of the cross-platform network externality they considered can be mapped to the
strength of the publishers in our model. Whereas, though, they studied the pricing power
of a monopoly provider of both platforms, we investigated the equilibrium prices of two
separate ad networks.

Customers’ preferences were also taken into account in [SJGH10]. Sen et. al.,
considered different adoption models for users. Their model accounts for both externalities
and user heterogeneity, and they give interesting insights on the behaviors of the players.
The difference with our work is that they do not consider the prices to be chosen strategically
by the providers, while we let the ad networks compete in both prices and quality.
Njoroge, et al. [NOSMW09] also considered duopoly competition between Internet Service Providers who compete in quality and prices not only for consumers, but for content providers as well. They study the effects of the asymmetry of the ISPs’ qualities in the equilibria of the game. They show that given two asymmetric platforms a price SPE on both sides of the market exists.

4.5 Conclusions

We presented a model to capture the incentives of advertising networks to fight click fraud. The analysis shows that ad networks maximize their revenues as the limit of the aggressiveness $x$ of classification algorithms approaches 1. Therefore, the ad networks resort to competing in prices to attract a larger fraction of publishers to display ads and get revenue. Our results show that the more asymmetric in quality the ad networks are, the more asymmetric their equilibrium prices will be. Another finding of our work is that as the publishers become more heterogenous, the competition in prices softens.
Chapter 5

On Nash Equilibrium Investment Strategies Against Click Fraud in Online Advertising Markets

In Chapter 4 we analyzed a game in which two ad networks are differentiated in the quality of their classification algorithms. The networks competed in two dimensions: the decided how aggressively to filter for invalid traffic and what fraction of the revenue to share with the publishers. We showed that networks are strongly incentivized to fight click fraud: Given a certain bidding policy from the advertisers, they filter invalid clicks as aggressively as possible and compete in the revenue share.

The above result, though it captures the strong incentives of the ad networks to fight click fraud, is by no means reflective of the reality. No ad network marks most of the traffic as invalid. We thus revisit the assumptions of the advertisers’ behavior and model
their bidding process in a more realistic way. Most advertisers are not willing to pay an
infinite amount for each bid / click even if they expect to receive a high conversion rate. Ad-
vertisers usually set upper limits for the prices they are paying per click for each advertising
campaign they launch. It is therefore more realistic to have a bidding function that results
to revenue\(^1\) that depends on the classification algorithms’ quality and aggressiveness\(^2\).

To address the above issues, we present a new model that captures the “imperfect”
bidding process of the advertisers. By “imperfect” we mean that the advertisers can only
infer the expected quality of their return on investment and bid appropriately. We establish
the bidding behavior based on the Capital Asset Pricing Model Theory (CAPM) which
describes the relationship between risk and expected return and is used in the pricing of
risky securities. We thus see the purchase of a click as the purchase of a risky asset.

This chapter is organized as follows. Section 5.1 describes the underlying economic
model. Section 5.2 provides the Nash equilibrium analysis. Section 5.3 presents numerical
experiments that show how the quality of the ad networks and the distribution of the
publishers’ preferences affect the investment strategies of the two ad networks. Finally, we
conclude with the main results and insights in Section 5.4. For a literature review on the
related work on click fraud we refer the reader to the previous chapter and Section 4.4.

**Main Contributions.**

1. We analyze the click fraud problem as a competition between ad networks who decide
   on the sensitivity of their classification algorithms.

2. We model the advertisers’ billing process using the Capital Asset Pricing Model.

---

\(^{1}\)The revenue is the number of clicks deemed valid multiplied by the price per click.

\(^{2}\)Note than in the previous chapter the revenue generated by the advertisers was a constant parameter,
independent from the quality \(\alpha\) and aggressiveness \(x\).
3. We provide a model that quantifies the impact of the ad networks’ investment to improve the quality of the classification algorithms.

4. We analyze the incentives of ad networks to invest in their classification algorithms in a two-stage game.

5. Our results give insights on how the bidding process of the advertisers affects the ad networks’ strategies and under what assumptions on the platform homogeneity the players are over or under investing compared to a planner that maximizes the social welfare.

5.1 Economic model

We consider two ad networks, called AN₁ and AN₂, that compete to receive clicks and display ads from the advertisers. The publishers are uniformly distributed along a line of length 1 between the two ad networks. Preferences are driven by anticipated reputation: A publisher could believe that one ad network is more qualified to match keywords on their websites with bids by the advertisers. In this context, we assume that AN₁ is “preferred” by some publishers and AN₂ is “preferred” by others.

We first study a one-shot game between the two ad networks. Each ad network ANᵢ, simultaneously decides at which sensitivity (or recall) to operate their classification algorithms. After both ad networks announce their decisions,

1. Publishers choose between the ad networks, according to their preferences and the revenue they get.
2. Ad networks mark a fraction of these clicks as valid.

3. Advertisers adjust their bids in ad auctions – based on the anticipated precision and recall.

4. Advertisers pay for the clicks marked as valid.

Our goal is to compute how sensitive the ad networks will be in marking clicks as valid, and how the market of publishers will react at equilibrium. We then investigate a two-stage game: At stage 1, the two ad networks decide how much to invest to improve their classification algorithms. Investment includes paying to collect more data that will act as training samples to train the classifiers. At stage 2, the ad networks compete for publishers, in the same way as in the one-shot game described before.

Our goal is to explore the investment strategies of the players: whether they are willing to invest to improve their algorithms and how much they are willing to spend to attract publishers. We first describe the classification model used by the ad networks and then we proceed by modeling the other two entities of the advertising market: publishers and advertisers.

5.1.1 Ad networks

Ad networks are endowed with the difficult task of filtering out valid from invalid traffic. It is difficult to know for sure whether a real user with an honest interest for the advertised product generated each click, thus any classification algorithm is prone to mistakes. The four possible combinations of predictions versus truth are shown in Table 5.1.
We assume that ad networks can identify fraudulent (invalid) clicks with a Precision-Recall (PR) curve of the form shown in Figure 5.1. We first give the definition of the precision and recall used throughout this section.

**Definition 9 (Recall).** Recall $y$ is the fraction of valid clicks that is marked as valid (TP).

**Definition 10 (Precision).** Precision $p$ is the fraction of clicks deemed valid that are indeed valid.

PR curves have been used as an alternative to Receiver Operator Characteristic curves (ROC), that show how the number of correctly classified positive examples varies with the number of incorrectly negative examples. PR curves give a more informative picture of an algorithm’s performance: ROC curves can provide a distorted - overly optimistic - view of an algorithm’s quality especially when there is a large skew class. This difference exists because in this domain the number of positive examples (valid clicks in our case) greatly exceeds the number of negative examples. Nevertheless, given any dataset, PR and ROC curves can be shown to be equivalent [DG06], hence our analysis is applicable if we consider a ROC curve as well.

Several methods have been proposed to generate ROC and PR curves. These include the binormal and the empirical (nonparametric) methods. In this paper we will
use the binormal model [BOSB10], which is the most commonly used method to generate smooth ROC curves (see [McC89] and [Met78]). In this setting, we consider two populations: the populations of valid and invalid clicks. We assume that the criterion variable follows a normal distribution in each population. Setting different thresholds on this criterion variable, we can then draw a smooth PR curve.

Let $X_+$ refer to the value of the criterion variable in the population of valid clicks and let $X_-$ refer to the value of the criterion variable in the population of invalid clicks. The binormal model assumes that both $X_+$ and $X_-$ are independently normally distributed with different means and variances. That is, $X_+ \sim N(\mu_{x_+}, \sigma_{x_+}^2)$, $X_- \sim N(\mu_{x_-}, \sigma_{x_-}^2)$. The means of the sampling distributions are $\mu_{x_+}, \mu_{x_-}$ and the variances of the sampling distributions are $\sigma_{x_+}^2 = \sigma^2/N$ and $\sigma_{x_-}^2 = \sigma^2/N$ respectively for the valid and invalid populations, where $\sigma^2$ is the variance of the population and $N$ the total population size.

Given these distributions, the ad networks select a threshold $c_t$, above which they classify the clicks as valid. The PR curve is traced out by the particular decision threshold $c_t$, where

$$y = 1 - \Phi_+(c_t)$$

is the recall and

$$p = \frac{r(1 - \Phi_+(c_t))}{r(1 - \Phi_+(c_t)) + (1-r)(1 - \Phi_-(c_t))}$$

is the precision. We use $\Phi_{+, -}(\cdot)$ to denote the Gaussian cumulative distribution function $\Phi(\cdot; \mu_{+,-}, \sigma_{+,-}^2)$. As a result, we get the PR curve, shown in Figure 5.1 for a number of samples $N = 5$. We will assume that PR curves generated are concave (smooth) and decreasing with respect to recall, without specifying an exact model for this function.
Figure 5.1: Precision vs. Recall curve: If the ad networks are willing to achieve a recall (true positive rate) of 0.8, they can achieve a precision of $p(0.8) = 0.97$.

Each ad network’s inherent type is the effectiveness of their classification algorithms that depends on the number of samples $N_i$ they have decided to acquire. Their strategic decision involves the sensitivity or recall $y_i$ to operate their algorithms: the ability to correctly identify the valid clicks. We define recall $y_i$ to be the fraction of valid clicks classified as valid. An ad network that is more sensitive identifying valid clicks would choose a higher recall $y_i$. In [MWGM08], the authors assume that each ad network $i$ was endowed with a type of receiver operation characteristic (ROC) curve that characterizes their click fraud technology. In this work, each ad network selects how much to invest to obtain more data for the training process of the detection algorithms. The more data each ad network possesses, the better the precision will be, given a certain target recall. Given $N_i$ and $y_i$, each ad network marks a fraction $y_i$ of valid clicks as valid. If $N_1 > N_2$, we say that AN$_1$ is more effective (for the same recall level, the precision is higher).

Precision is a function of the recall $y_i$ and the number of samples $N_i$. Ideally an
Figure 5.2: The advertising market comprises of three entities: advertisers who pay for clicks deemed valid, ad networks who classify clicks as valid and publishers who display ads.

An ad network would like to have a high precision and recall. The goal of the ad networks is to maximize their revenue

\[ J_i(y_i, N_i) = (1 - h) \cdot b(y_i, N_i) \cdot DV_i(y_i, N_i) \cdot V \cdot M_i(y_i, y_{-i}, N_i), \]

(5.1)

where \( h \) is the fraction of the revenue given to the publishers\(^3\), \( b(y_i, N_i) \) is the advertisers' bidding function, \( DV_i \) is the fraction of clicks deemed as valid, \( V \) is the total volume of clicks and \( M_i \) is the market share of each ad network that depends on both players' strategies.

We should note that \( DV_i \) is again a function of both the precision and the recall \( y_i \).

\[ DV_i = r \cdot \frac{y_i}{p_i(y_i)}, \]

where \( r \) is the fraction of total clicks that are real or valid. The latter expression comes from the definition of precision after solving for \( N_i \) (precision is the rate of truly valid clicks

---

\(^3\)In this chapter we assume that the revenue share is common for both networks. As an extension, one can assume a second degree of freedom to attract publishers, that would make the analysis more complex.
5.1.2 Publishers

We develop a simple Hotelling competition model, in which the market of publishers divides according to the revenue given by each ad network and the preferences of the publishers. We consider that publishers have different preferences for ad networks. Publishers are uniformly distributed along a line $\theta \in [0, 1]$, as shown in Figure 5.3.

There are two ad networks, located at each extreme who compete for publishers (to be able to show ads on their websites). An important difference between the ad networks is their location with respect to the publishers (the preferences of the publishers). Each ad network marks a percentage of the clicks as valid (and thus charges the advertisers) and gives a fraction of the revenue to the publishers. Depending on the recall the ad networks are operating at and the resulting precision of their algorithms, the bid of the advertisers and consequently the revenue acquired by each network is different.

Let there be an additive component that reflects the preferences of the publishers. If a publisher is at distance $d$ to the ad network that is located at point 0, the resulting revenue is $t \cdot (1 - d)^2 + V$, meaning that the further the publisher is located from an ad network...
network, the “less” this publisher is expecting to get from the revenue originally shared with this publisher.

Consequently, the aggregate utility of a publisher $\theta$ is

$$U_{p1}(\bar{\theta}) = t \cdot (1 - \theta)^2 + P_1 \text{ from AN}_1$$

$$U_{p2}(\bar{\theta}) = t \cdot \theta^2 + P_2 \text{ from AN}_2,$$

where $P_1, P_2$ is the revenue given out from each ad network to the good publishers, with $P_i(y_i) = h \cdot r \cdot c \cdot V \cdot b \cdot DV_i \cdot p_i(y_i)$, since the fraction of the charged clicks that are coming from the good publishers is the fraction of clicks deemed valid that are indeed valid (and equals the precision $p$).

In order to derive the demands (the volume of the clicks generated by the market of the publishers) for each ad network, we need to derive the publisher $\bar{\theta}$ that is just indifferent between Ad networks 1 and 2. We define $\bar{\theta}$ as the location where $U_{p1}(\bar{\theta}) = U_{p2}(\bar{\theta})$. Equating the delivered publishers’ revenues we have $P_1(y_i) + t \cdot (1 - \bar{\theta})^2 = P_2(y_i) + t \cdot \bar{\theta}^2$ which gives $\bar{\theta}$ or else the market share

$$M_1 = \frac{1}{2} + \frac{P_1(y_1) - P_2(y_2)}{2 \cdot t}. \quad (5.2)$$

For example, if each ad network is giving out the same amount of money to the publishers, the market will split at the point $\bar{\theta} = 0.5$, meaning that each ad network will get half the market of publishers.

Parameter $t$ captures the degree of the heterogeneity or bias of the publishers between the two networks. It reflects the importance of the preferences of the publishers with respect to the prices. When $t$ is large, we say that publishers are highly heterogeneous.
and biased toward a certain ad network and preferences are more important; when \( t \) is small, prices have a greater impact on the decision of the publishers.

5.1.3 Advertisers

As in [MWGM08], we assume that the number of advertisers is sufficiently large and covers the number of ad positions on the publishers’ websites. This is a realistic assumption, as the advertisers are actually competing to display their ads through auctions. The advertisers adjust their bids to increase their return on investment. It is arguable they would have such a strategy since they would want to invest in online advertising up until the point its return is comparable to that achieved from other forms of advertising. Depending on the quality of the clicks they pay for, they adjust their bids to account for clicks of inferior quality. The difference with prior works is that now we assume that the advertisers’ bidding process captures both the precision and the recall and not just the aggressiveness as in [DM13].

We generalize the bidding process of the advertisers by assuming a factor of

\[
b(y_i, p_i) = c \cdot \frac{p_i(y_i)}{y_i} \left(1 - \alpha \sqrt{\frac{1}{ry_i}}\right),
\]

(5.3)

where \( \alpha \) is a coefficient that depends on the expected revenue and the risk associated with the bid, and \( c \) is some constant price per click, without any adjustments. We avoid dealing with the auctions mechanism details and focus on the implications of the recall levels selected by the ad networks.

The nature of the bidding function plays a critical role in the results of the equilibrium strategies for the players. We justify the exact form of the bidding function (5.3)\(^4\).

\(^4\)Note that \( x \) (aggressiveness) + \( y \) (recall) = 1.
### Table 5.2: List of variables introduced in Section 5.1

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h$</td>
<td>revenue share given to publishers</td>
</tr>
<tr>
<td>$y_i$</td>
<td>recall of ad network $i$</td>
</tr>
<tr>
<td>$p_i$</td>
<td>precision of ad network $i$</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>CAPM coefficient</td>
</tr>
<tr>
<td>$r$</td>
<td>fraction of clicks that are valid (“real”)</td>
</tr>
<tr>
<td>$c$</td>
<td>price per click</td>
</tr>
<tr>
<td>$M_i(y_i, y-i, p_i)$</td>
<td>market share for network $i$</td>
</tr>
<tr>
<td>$b(y_i, p_i)$</td>
<td>advertisers’ bidding function</td>
</tr>
<tr>
<td>$DV_i(y_i, p_i)$</td>
<td>fraction of clicks that are deemed valid by network $i$</td>
</tr>
<tr>
<td>$P_i(y_i)$</td>
<td>payment to publishers from ad network $i$</td>
</tr>
<tr>
<td>$V$</td>
<td>volume of clicks for ad network $i$</td>
</tr>
<tr>
<td>$t$</td>
<td>degree of heterogeneity</td>
</tr>
<tr>
<td>$\bar{\theta}$</td>
<td>location of indifferent publisher</td>
</tr>
<tr>
<td>$N$</td>
<td>number of samples acquired</td>
</tr>
</tbody>
</table>

Figure 5.4: In stage 1 ($G_2$) the players select how much to invest while in stage 2 ($G_1$) they select their recall.

in the Appendix, where we also give an overview of the Capital Asset Pricing Model, thoroughly explained in [FF04]. The bidding function should capture the tradeoff between the precision and recall: The bid is increasing in precision but there is an upper bound in how much the bid can increase as recall decreases.

### 5.2 Equilibrium Analysis

In this section we investigate two types of games between the two ad networks, as shown in Figure 5.4.
5.2.1 One-shot game

We first analyze the case when the ad networks have fixed PR (fixed \(N\)) curves and simultaneously decide the recall to operate on that curve.

**Definition 11 (Game \(G_1\)).** Game \(G_1\) is a one-shot game between two ad networks who compete for publishers. They select the recall \(y_i \in [0, 1]\) at which to operate their classification algorithms.

Following the previous analysis and from Eq. (5.1), the profits of the ad networks are given by:

\[
J_1(y_1) = (1 - h) \cdot r \cdot \frac{y_1 \cdot \alpha}{p_1} \cdot \frac{1}{y_1} \cdot M_1(y_1, y_2, p_1) \quad (5.4)
\]

and

\[
J_2(y_2) = (1 - h) \cdot r \cdot \frac{y_2 \cdot \alpha}{p_2} \cdot \frac{1}{y_2} \cdot M_2(y_2, y_1, p_2) \quad (5.5)
\]

where \(M_i \in [0, 1]\) is the market share of ad network \(i\). If there exist combinations of \(y_1, y_2\) that give a market share higher than 1, we impose a saturation of the market at 1. Each ad network \(i\) needs to select \(y_i \in [0, 1]\) to maximize the revenues given by (5.4) and (5.5) respectively.

**Theorem 6.** There exists a Nash equilibrium in the one-shot game \(G_1\).

**Proof.** We will show that the revenue functions of the ad networks are quasi-concave. Simplifying the revenue function, we get

\[
J_i(y_i) = (1 - h) \cdot r \cdot c \cdot f_i(y_i) M_i(y_i), \quad (5.6)
\]
where \( f_i(y_i) = \left(1 - \alpha \sqrt{\frac{1}{ry_i}}\right) > 0 \). We are going to show that \( J_i(\cdot) \) is quasi concave as a product of two positive concave functions one of which is nondecreasing \((f)\). Indeed, differentiating \( f \) twice and eliminating any positive constants yields

\[
\frac{\partial f_i^2}{\partial y_i^2} = -\frac{3}{4} \alpha y_i^{-\frac{5}{2}} < 0. \tag{5.7}
\]

Differentiating twice the market share we get

\[
\frac{\partial M_i^2}{\partial y_i^2} = \frac{1}{2t} \left( \frac{\partial^2 f_i}{\partial y_i^2} p_i + 2 \frac{\partial f_i}{\partial y_i} \frac{\partial p_i}{\partial y_i} + f_i \frac{\partial^2 p_i}{\partial y_i^2} \right) \frac{\partial f_i}{\partial y_i} \frac{\partial p_i}{\partial y_i}. \tag{5.8}
\]

Since the payoff functions of the ad networks are continuous and quasi concave in a compact set, there exists a Nash equilibrium in pure strategies.

\[\square\]

### 5.2.2 Stage game

**Definition 12** (Game \( G_2 \)). Game \( G_2 \) is a two-stage game. At the first stage, the ad networks decide their investment strategies: how many data samples \( N_i \) to acquire. At the second stage, the ad networks play the one-shot game \( G_1 \).

### 5.3 Numerical experiments

In this section, we gain insights into the Nash equilibria of the game through numerical experiments. We also compare the Nash equilibria and the resulting social welfare in two different scenarios.
• Duopoly: In this case we let the players select their optimal investment for the number of data samples and then the recall.

• Social optimum: In this case, we assume that there is a social planner that seeks to maximize the social welfare. Namely the planner seeks to maximize the money given to good publishers and the net profits of the ad networks. The social planner we consider here is a sort of “investment social planner”. In this context, the two ad networks compete to select their optimal recall levels for any combination of investment for the number of samples \( N_1 \) and \( N_2 \). Given the outcomes of this game \( G_1 \), the social planner selects the investment policies for each ad network in a discrete game to maximize the social welfare in the system.

5.3.1 Single-shot game

In this section we delve into the single-shot game, in which the ad networks decide on the recall of their classification algorithms. Their choice of recall affects the bidding strategies of the advertisers, as the total money poured into the system is not constant. Given the model for the bidding strategies of the advertisers presented in Section 5.1, we first investigate how parameter \( \alpha \) affects the Nash equilibrium strategies of the ad networks.

The first experiment studies the impact of the bid adjustment factor of the advertisers’ recall selected at equilibrium. Note that when \( \alpha = 0 \), the bid adjustment factor equally \( cp/y \) and reflects the perfect adjustment scenario that was presented in Chapter 4. Parameter \( \alpha \) represents the price of risk. As \( \alpha \) increases, the bidders (advertisers) decrease their bids.
Equilibrium recall increases as $\alpha$ increases

We then explore the equilibrium strategies in the one-shot game when both players have fixed their investment with respect to AN$_2$'s selection of $N_2$. As we see in Figure 5.6 at which AN$_1$ has fixed his investment at $N_1 = 5$, as the opponent’s PR curve improves, AN$_1$’s equilibrium recall becomes higher. We also observe that competition becomes fiercer as both players increase the recall selected in equilibrium. The market share of AN$_1$ decreases with the opponent’s classification quality. There is a critical sample size $N_2 = 7$ after which AN$_2$ has already won over the market of publishers, and no matter what AN$_1$ is doing (by increasing the recall) he cannot compete with AN$_2$ any more.

### 5.3.2 Two-stage game

We now investigate the two-stage game in which the players first decide their investment strategies and then the optimal recall. We compare the investment selected in two cases: when the ad networks compete and when a social planner decides on their investment strategies to maximize the social welfare.
Figure 5.6: Equilibrium recall versus $AN_2$’s investment: As the quality of one network increases, both networks select higher recall levels at equilibrium.

Whether the players will decide to over or under invest depends mainly on two factors: (a) the market penetration with respect to the preferences of publishers, and (b) the revenue share they are going to keep for themselves. The market penetration or else how much the ad network is preferred versus the competitors is captured by the degree of heterogeneity $t$. When $t$ increases, the impact of the networks’ performance in terms of the accuracy and revenue they generate decreases. Higher $t$ means that one of the ad networks is highly preferred by publishers for reasons exogenous of the quality of the classification algorithms of this ad network. For example, certain publishers might consider Yahoo Ad Network to be closer to their users’ demographics or some other publishers might trust Google Ad Network to better respond to their claims.

The other crucial parameter is the revenue share $h$ the ad networks keep for themselves. As $h$ increases, we expect that ad networks invest more, as the effects of their increased investment, namely, the increased revenue from the advertisers, will be largely concentrated on the ad networks’ side. On the opposite side, when $h$ is small, ad networks
are disincentivized to invest, as the extra cost of their investment will be diluted with the publishers. Our theoretically predicted results are indeed shown in Figure 5.7. We see that there are two regions of interest. For small values for $h, t$, the networks are over investing, whereas for higher values (upper right corner) the players are underinvesting, as compared with the optimal investment strategies imposed by a social planner.

5.4 Conclusion

We presented a model to capture the incentives of ad networks to fight click fraud. We modeled the bidding process of the advertisers to incorporate the risk associated with the return of investment, and let them adjust their bids according to both the precision and the recall of the ad networks’ classification algorithms. In this regime, there is a tradeoff for the ad networks between precision and recall. Increasing recall will give the legitimate (good) publishers the deserved revenue fraction for their valid clicks but will also lower
the precision and the money the advertisers are willing to pay. We explored this tradeoff
and proved that there exists a Nash equilibrium in a single-shot game between two ad
networks who compete for publishers. We also explored the incentives of the ad networks
to invest in a two-stage game, in the first stage of which the ad networks can improve the
performance of their algorithms. Our results indicate that as the quality of one network
increases, the competition becomes fiercer as both networks decide to increase the recall
selected in equilibria. We also compared the resulting equilibria and the social welfare of the
two-stage game with the social optimum. We concluded that as the market of publishers
is more differentiated with respect to the their preferences between the ad networks and as
the revenue share given to the publishers increases, there is a tendency for the ad networks
to invest less than the investment policies a social welfare maximizer planner would impose.
Chapter 6

Conclusions and Future Work

6.1 Summary of Contributions

The focus of this thesis, as stated in the introduction, was:

to develop game-theoretic models capable of capturing the interactions between a defender and an attacker in a strategic classification game and use these models to: (a) predict the equilibrium attack and defense strategies of the adversary and the defender respectively, (b) qualitatively and quantitatively explore the impact of the underlying parameters on the equilibrium strategies and payoffs, and (c) investigate investment incentives of the defenders to increase the quality of their classification algorithms in both the presence and the absence of competition.

The game theoretic models we have explored are general-purpose and simple so that closed form solutions for the equilibria are found, yet intuitive and informative. They provide guidelines and directions to use these results to make classifiers as robust as possible and prevent adversaries from reconstructing the classification rules and slightly shift their attack effort to evade them.
6.2 Future Work

While this thesis has made several contributions in the field of strategic classification using game theoretic tools, the problem is in no way “solved.” This work in fact opens up many avenues for future work. We now discuss open research directions separately for each part of the thesis.

6.2.1 Part I: Adversarial Classification

One of the most promising directions is to combine results by Machine Learning and Game Theory. Our analysis identified the need for more sophisticated countermeasure defenses. One can develop classifiers that successfully identify fraudulent activity but the implementation of these classifiers is as important as the task of classification. A strategic defender will (and should) expect that a strategic adversary will be able to detect over time what classifiers were used, and try to circumvent them, as was shown in the experiment in [TMG+13]. Randomization and Game Theory could help solve this problem, as the strategic interactions of defender and attacker are inherently accommodated and reflected in their payoff functions.

In Chapter 2 we justified the optimality of threshold strategies in equilibrium for the defender. Indeed, rules such as threshold schemes on aggregated information (reward, volume per hour, connections per source and others) are shown to be more efficient in reality versus more complex machine learning approaches, especially under highly noisy environments [SP10]. Applying a randomized threshold on the attacker reward works in the specific model we examined, but when the attack strategies become more perplexed (e.g.,
when sequence of attacks / actions highly affects the reward), it might not be appropriate to use a threshold on the reward. It would be interesting to see in what scenarios qualitative rules on the nature of the attacks are more suitable versus quantitative measures such as a threshold on attack reward. Could a potentially efficient metric be a similarity index in multiple dimensions between the non-attacker’s expected behavior and the attacker’s observed attack? And if so, which dimensions should one consider?

6.2.2 Part II: Click Fraud Classification

The online advertising market comprises of many players and a lot of variables that affect the revenue of the ad networks. It is difficult to build a model that captures all such elements realistically and still be mathematically tractable. Our work provides equilibrium results in two basic scenarios and provides investment guidelines to ad networks, when these are competing for high quality traffic. There are a few directions we would like to point out that are worth of future effort.

Predictive pricing. We limited our study in two decision strategies for the ad networks: revenue share and filtering. There is a third tool that ad networks are using to attract high quality publishers and advertisers: predictive pricing. Predictive pricing allows ad networks to charge the same advertiser different prices for clicks originated in different publishers. Work has been done to investigate the impact of predictive pricing [MGM08]. Equilibrium results that take into effect that tool with or without the combination of the other two dimensions would further enlighten the interactions between the different entities in the advertising ecosystem.

Noisy information. In Game Theory it is standard to assume that there is common
information between the players. In our case, the fraction of valid clicks $r$ was assumed to be known in both the advertisers and ad networks. Nevertheless, the advertisers see a different manifestation of click fraud, having access to different sources of information. For example, the ad network might have access to aggregated information about a specific user who generated a click on other websites, but the advertiser might have access to information related to the aftermath of a click and whether a purchase or some other action was or was not made. It would be useful to see how the uncertainty on the click fraud ratio would impact the advertisers’ bids and consequently the ad networks’ strategies. This uncertainty could be captured by a distribution on a click fraud ratio with a certain mean and variance.

**Tipping game or congestion game.** In this analysis of the click fraud game, we have assumed that there is a constant population of fraudulent publishers. What if fraudulent publishers have the freedom to select their preferred ad network based on their revenue? Will they be gathered in the network with the highest quality classification algorithms (congestion game) or will they be divided among inferior networks (tipping game), resulting to different click fraud ratios among the two ad networks?

The above directions could only enrich our understanding on the Nash equilibria in click fraud games. The more understanding we get, the more realistic economic models we will build and the higher the chances the results will influence important decisions in the major ad networks’ click fraud teams.
Appendix A

Deriving the advertisers’ bidding function using the Capital Asset Pricing Model

We present a model that captures the expected return on investment on the advertisers’ side and the risk associated with bidding (and paying) for clicks that might be fraudulent, depending on the quality of the classification algorithms of the ad networks.

A.1 The Capital Asset Pricing Model

The theory of Capital Asset Pricing Model, introduced by Treynor [Tre61, Tre62], Sharpe [Sha64], Lintner [Lin65b, Lin65a] and Mossin[Mos65] independently, also known as CAPM, offers intuitive predictions about how to measure risk and the relation between expected return and risk.
CAPM Formula.

\[ E(R_i) = R(f) + \beta_i (E(R_m) - R_f), \tag{A.1} \]

where \( E(R_i) \) is the expected return of asset \( i \), \( E(R_m) \) is expected return of the market, \( R_f \) is the risk free rate of interest and \( \beta_i \) is the sensitivity of the expected return of asset \( i \) to the expected excess market return, with \( \beta_i = \frac{cov(R_i, R_m)}{var(R_m)} \).

Note that the expected market return \( E(R_m) \) is usually estimated by measuring the arithmetic average of the historical returns on a market portfolio (e.g. S&P 500), and \( R_f \) is an arithmetic average of historical risk free rates of return (such as interest from government bonds). We are going to interpret the above parameters in the context of the advertising market in the following section.

A.2 Applying CAPM on the advertiser’s bidding functions

In the advertising market, the advertisers are constantly asking themselves the following question when they design their advertisement campaigns: “If bidding \( X \) would give me the preferred return on investment, assuming that the ad networks are maintaining a certain recall and precision, should I bid less, equal or more than \( X \)?” There is an apparent risk associated with the advertisers’ bidding. They are not aware of: (1) the accuracy of the classification algorithms of the ad networks, (2) the recall (or aggressiveness) selected and, (3) the underlying ratio of the valid clicks. We will use the Capital Asset Pricing Model developed above to price the bids in relation to their associated risk.

The advertisers’ goal is to continue doing business with the ad networks as long as they maintain a certain return on investment (ROI). Otherwise they could direct their
efforts (and money) to other forms of advertising, such as newspaper, radio or television. The advertisers need to decide how much to bid \( \hat{b} \). We assume that each valid click yields a constant gain \( c \). Since there are \( r \cdot V \) valid clicks in the market, the ROI of the advertisers is \( r c V \). Therefore

\[
\hat{b} \cdot DV \cdot V = r c V,
\]

where \( \hat{b} \) is the value per click the advertisers are willing to pay (asset value), \( DV \) is the fraction of clicks that are deemed valid. Since \( DV = \frac{ry}{p} \), solving for the asset value we get:

\[
\hat{b} = p \cdot \frac{c}{y}.
\]

The advertiser keeps estimating the asset value \( \hat{b} \) using a moving window of past data. This past data could represent data about actions / traffic of users whose clicks were deemed valid. We assume that the process of collecting these samples is similar with the process of collecting identically and independently distributed samples, therefore the asset value \( \hat{b} \) is stationary. Only information about the clicks deemed valid is useful, because this is what the ad network is charging for (and gives some further information), therefore the sample size is proportional to \( p \cdot DV = ry \). The coefficient of variation \((CV)\) is a measure of dispersion of a probability distribution function. It is defined as the ratio of the standard deviation to the mean \( (CV = \frac{\sigma}{\mu}) \). By the Central Limit Theorem, as the sample size becomes large, the standard deviation of the sampling distribution of the mean approaches \( \frac{\sigma}{\sqrt{\text{sample size}}} \). Therefore as the advertiser collects more and more data, the coefficient of variation \( CV \propto \frac{1}{\sqrt{ry}} \).

We can now apply the CAPM model as follows.
\[ \bar{r}_i = r_f + \beta_i (\bar{r}_m - r_f) \]

\[ \Rightarrow \bar{r}_i = r_f + \frac{\sigma_{m,i}}{\sigma_m} (\bar{r}_m - r_f) \]

\[ \Rightarrow \frac{b}{\text{bid}} = 1 + \alpha \sigma_{m,i} \]

\[ \Rightarrow \frac{(pc)}{\text{bid}} = 1 + \alpha \sigma_{m,i}, \quad (A.3) \]

after substituting for \( r_f = 1 \), \( r_i = \frac{b}{\text{bid}} \), and \( \alpha = \frac{\bar{r}_m - 1}{\sigma_m} \). For the standard deviation of the return from the asset \( i \), we have:

\[ \sigma_{m,i} = \frac{\text{return}}{\text{bid}} = \frac{cp}{y} \frac{1}{\text{bid} \sqrt{ry}}. \quad (A.4) \]

Plugging (A.4) in (A.3) and solving for \( \text{bid} \) finally yields the formula for the bidding function for the advertisers.

\[ \text{bid} = \frac{cp}{y} \left( 1 - \frac{\alpha}{\sqrt{ry}} \right). \quad (A.5) \]
Bibliography


[ASZ02] Rudolf Avenhaus, Bernhard Von Stengel, and Shmuel Zamir. Inspection


