1-1-2015

Analytics and Healthcare Costs (A Three Essay Dissertation)

Lina Bouayad
University of South Florida, lina.bouayad@gmail.com

Follow this and additional works at: http://scholarcommons.usf.edu/etd

Part of the Business Commons

Scholar Commons Citation

This Dissertation is brought to you for free and open access by the Graduate School at Scholar Commons. It has been accepted for inclusion in Graduate Theses and Dissertations by an authorized administrator of Scholar Commons. For more information, please contact scholarcommons@usf.edu.
Analytics & Healthcare Costs

(A Three Essay Dissertation)

by

Lina Bouayad

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
Department of Information Systems and Decision Sciences
with a concentration in Information Systems
Muma College of Business
University of South Florida

Co-Major Professor: Balaji Padmanabhan, Ph.D.
Co-Major Professor: Kaushal Chari, Ph.D.
Terry Sincich, Ph.D.
Wolfgang Jank, Ph.D.

Date of Approval:
April 14, 2015

Keywords: Auditing Algorithms, Sentinel Effect, Time Pressure, Recommender Systems, Patient Compliance, Sentiment Analysis

Copyright © 2015, Lina Bouayad
# TABLE OF CONTENTS

LIST OF TABLES.......................................................................................................................iv

LIST OF FIGURES....................................................................................................................vi

ABSTRACT ...................................................................................................................................... vi

CHAPTER 1: DISSERTATION OVERVIEW .................................................................................1

CHAPTER 2: AUDIT ALGORITHMS UNDER THE SENTINEL EFFECT ............................3
  Introduction ...............................................................................................................................3
  Underlying Theory and the Diffusion-Deterrence Model .....................................................5
    The Provider Network ...........................................................................................................5
    Audit Information Diffusion .................................................................................................6
    Deterrence – Behavior Change ............................................................................................8
    High Level Set-up ................................................................................................................10
  Audit Algorithms - Background ............................................................................................12
  Audit under the Sentinel Effect ............................................................................................13
    Algorithm Outline ..............................................................................................................13
    Expected Network Deterrence Computation ...................................................................15
  Results ......................................................................................................................................20
    Healthcare Results .............................................................................................................20
      Network Topology Influence .............................................................................................21
      Diffusion ..........................................................................................................................23
      Decay (Cases A and D) .....................................................................................................25
      Extent of Waste and Abuse (Cases A, E, and F) .............................................................25
      Detection versus Deterrence .............................................................................................25
    Case Study: Warranty Claims .............................................................................................27
      Audit Claims Data Analysis ..............................................................................................27
      Network Topology ...........................................................................................................28
      Model Modification ..........................................................................................................29
      Simulation Results ...........................................................................................................29
  Analytical Results ..................................................................................................................32
    Special Case Scenarios – Analytical Results .....................................................................37
    Special Case Scenarios – Sensitivity Analysis ..................................................................40
      Scenario 1 .........................................................................................................................40
      Scenario 2 .........................................................................................................................41
      Scenario 3 .........................................................................................................................43
  Conclusions ...........................................................................................................................44
  References ..............................................................................................................................45
CHAPTER 4: PATIENTS’ COMPLIANCE ................................................................. 76
Introduction....................................................................................... 76
Literature Review............................................................................. 77
Compliance Definition..................................................................... 77
Patient Compliance and Outcome................................................... 77
Mortality & Survival................................................................. 78
Hospitalization................................................................. 78
Violence................................................................................. 78
Symptom Reduction, Disease Recovery and Relapse ...................... 79
Disease Progression................................................................. 79
Economic Impact of Patient Noncompliance................................. 80
Compliance Measures................................................................. 81
Direct Measures of Compliance.................................................. 82
Indirect Measures of Compliance ............................................... 83
LIST OF TABLES

Table 1: Variables Used in the Process .................................................................................................................. 19
Table 2: Greedy Deterrence Heuristic’s Percentage Wins .......................................................................................... 26
Table 3: Service Practitioner’s Payoff ..................................................................................................................... 32
Table 4: Insurance Provider’s Payoff ....................................................................................................................... 34
Table 5: Service Provider’s Optimal Defraud Probabilities ......................................................................................... 38
Table 6: Insurance Provider’s Expected Payoff at Equilibrium (F) ........................................................................... 39
Table 7: Systems Recommendations Influence Factors – Variables ........................................................................... 54
Table 8: List of Hypotheses ........................................................................................................................................ 56
Table 9: Experiment - Time Constraint Level ........................................................................................................... 58
Table 10: Experiment Treatments/Groups – Subjects ................................................................................................. 59
Table 11: Descriptive Statistics – Time Pressure Level (View Count, Adjust Count) .................................. 61
Table 12: Descriptive Statistics - Group Level (View Count, Adjust Count) ........................................................ 61
Table 13: Descriptive Statistics – Group/Time Pressure (View Count, Adjust Count) .................................. 63
Table 14: Analysis of GEE Parameter Estimates for “Adjust =1” ............................................................................. 64
Table 15: Analysis of GEE Parameter Estimates for “View =1” ............................................................................. 64
Table 16: Noncompliance Outcomes ........................................................................................................................ 80
Table 17: Noncompliance Measures ......................................................................................................................... 86
Table 18: Noncompliance Factors ........................................................................................................................... 89
Table 19: Experiment - Data Variables ..................................................................................................................... 91
Table 20: Labeling Examples .................................................................................................................................... 92
LIST OF FIGURES

Figure 1: Probability of Audit Information based on Fraud and Deviation Amount ........................................ 8
Figure 2: The Greedy Deterrence Heuristic Algorithm ...................................................................................... 15
Figure 3. The Calculate Expected Network Deterrence Value (DV) Function .................................................... 18
Figure 4. Deterrence vs. Detection Amounts (Network Topology) .................................................................. 22
Figure 5: Deterrence vs. Detection Amounts (Diffusion, Decay and Waste and Abuse Extent) ....................... 24
Figure 6. Audit Claims Data ................................................................................................................................. 28
Figure 7: Deterrence vs. Detection Amounts (Dealer Network) ......................................................................... 31
Figure 8: The Deterrence Algorithm .................................................................................................................. 33
Figure 9: Game Tree ............................................................................................................................................. 34
Figure 10: Insurance Provider Payoff by Audit Cost and Fraud Amount – Scenario 1 .................................. 41
Figure 11. Insurance Provider Payoff by Audit Cost and Fraud Amount – Scenario 2 ............................... 42
Figure 12. Insurance Provider Payoff by Audit Cost and Fraud Amount – Scenario 3 ............................... 43
Figure 13: Participants Profiles ............................................................................................................................. 60
Figure 14: Inductive Tree – Adjusting Treatment Options \{1 = X, 0 = Y\} .............................................................. 65
Figure 15: Influence Dynamics – Comprehensive Model .................................................................................. 67
Figure 16: Sample Simulation Results – Top-N++ ............................................................................................ 71
Figure 17: Literature Survey - Framework .......................................................................................................... 76
Figure 18: Text Processing Steps ....................................................................................................................... 93
ABSTRACT

Both literature and practice have looked at different strategies to diminish healthcare associated costs. As an extension to this stream of research, the present three paper dissertation addresses the issue of reducing elevated healthcare costs using analytics. The first paper looks at extending the benefits of auditing algorithms from mere detection of fraudulent providers to maximizing the deterrence from inappropriate behavior. Using the structure of the physicians’ network, a new auditing algorithm is developed. Evaluation of the algorithm is performed using an agent-based simulation and an analytical model. A case study is also included to illustrate the application of the algorithm in the warranty domain. The second paper relies on experimental data to build a personalized medical recommender system geared towards re-enforcing price-sensitive prescription behavior. The study analyzes the impact of time pressure, and procedure cost and prescription prevalence/popularity on the physicians’ use of the system’s recommendations. The third paper investigates the relationship between patients’ compliance and healthcare costs. The study includes a survey of the literature along with a longitudinal analysis of patients’ data to determine factors leading to patients’ non-compliance, and ways to alleviate it.
CHAPTER 1: DISSERTATION OVERVIEW

Estimates indicate that as much as $700 billion dollars spent every year does not lead to improving healthcare outcomes (Kelly 2009). In the medical realm, several claims for procedures and tests done to patients are classified as improper or unnecessary and do not contribute to better outcomes. The US leads the world in healthcare spending with approximately 20% of GDP related to healthcare, leading to more attention in recent times on whether the money being spent is being spent well.

As defined by the National Health Care Anti-Fraud Association (NHCAA), waste and abuse includes performing medically unnecessary services. Others define waste and abuse as misuse, overuse, or underuse of medical procedures resulting in elevated healthcare costs (Kelly 2009).

The statistics indicate that there is room for improvement. Prior research has identified several strategies to reduce and/or to eliminate waste and abuse ranging from spreading adoption of preventive care measures and promoting healthy lifestyles, to setting up waste-reduction goals and rewarding patient whistle-blowing (Berwick et al. 2012).

To help achieve the healthcare cost reduction goal, we investigate the use of 1) a network-based auditing algorithm, 2) a recommender system that presents cost information to physicians in real-time as they are about to prescribe procedures, and 3) a patient compliance model that illustrates predicted patients compliance to prescribed treatment regimens over time.

Through the first study, we look at the network effect in spreading medical audit information; which in turn is expected to trigger deterrence from inappropriate behavior. Hospitals and medical offices harness the interaction between providers to form social networks.
Even though understudied in academia, medical provider networks create a medium for information and behavior diffusion, referred to in the industry, as the *sentinel effect*.

In light of this effect, the first study aims at developing a deterrence-geared auditing algorithm for medical claims. Using agent-based simulation, a network of provider is created, a new deterrence-geared auditing is implemented, and performance is evaluated. The study also includes a case study through which the same effect is analyzed in the warranty domain.

Taking into account that waste and abuse in healthcare is often caused by lack of providers’ knowledge of procedure cost information, we focus the second study on investigating the use of medical recommender systems that provide procedure cost-related information. Under the Obama administration in the US, measures have been taken to start holding medical providers accountable for any improper billing. Under the new law, providers are required to be “vigilant about the legality of their own activities or potentially pay a price for not doing so… Providing are now required to report and repay within 60 days any overpayment from Medicare or Medicaid.” (Iglehart 2010) Hospitals and physicians in general are thereby required to react swiftly and deter from any improper billing (Morris 2009). By presenting data pertaining to costs associated with each procedure, we can re-enforce a cost-sensitive behavior within the provider network.

After addressing healthcare costs reduction matters at the provider level, we analyze the impact of patient behavior on elevated costs. Patient noncompliance has been recognized as a major concern in healthcare. In prior literature, patients’ non or low compliance has been linked to suboptimal outcomes and waste of medical resources causing higher healthcare costs (Gruman et al., 2010). Even though prior literature has been looking at compliance in several clinical cases, compliance trends over time have not been identified. Thanks to the newly implemented electronic medical record (EMR), we now have the opportunity to analyze compliance data longitudinally, and determine 1) how compliance might deteriorate over time, and 2) factors that lead to a more sustainable patient compliance.
CHAPTER 2: AUDIT ALGORITHMS UNDER THE SENTINEL EFFECT

Introduction

When there is a third party payer there is often a greater chance of inappropriate claims being submitted due to moral hazard. The recipient of the service is often not directly financially accountable. Fraudulent claims however are known to drive up costs and represent a real challenge for several industries. A few years ago, CBS news reported that “Medicare fraud - estimated now to total about $60 billion a year - has become one of, if not the most profitable, crimes in America” (Reiner, 2010). In manufacturing, warranty fraud is estimated to total about $7 billion yearly, representing 10% to 15% of all warranty claims (Froning, 2010).

In addition to fraudulent activity, there is evidence to suggest that “waste and abuse” are common. In healthcare, waste and abuse show up as unnecessary tests or up-coding the severity of a patient’s visit. As defined by the National Health Care Anti-Fraud Association (NHCAA), waste is an act of negligence by medical providers who misuse, over or under utilize services and other practices. Conversely, abuse is the use of services not professionally recognized as standards of care. Examples include inappropriate procedures and unnecessary prescription refills. In other domains, payers face a similar trend referred to as buildup (Tennyson & Salsas Forn, 2002). While healthcare costs are a big issue today, these problems show up in other domains too. Some service providers in the automotive industry also tend to overbill warranty providers for services rendered and parts repaired and/or replaced. These exaggerated loss amounts translate into unnecessary costs to warranty and insurance providers.

While these are not fraudulent activities - there is still a patient who needs treatment or a product to be repaired - they are practices that cumulatively place an enormous burden on service costs.
According to a report published by the Institute of Medicine excess costs amounted to $750 billion in the year 2009 with unnecessary services totaling $210 billion, inefficiently delivered services $130 billion, and fraudulent activities $75 billion (Smith, Saunders, Stuckhardt, & McGinnis, 2012). Payers such as Medicare and Medicaid in healthcare, as well as firms such as Ford and Caterpillar in manufacturing could benefit enormously from effective techniques to control inappropriate billing activity.

One such technique is the auditing of claims, a common industry practice. According to prior research, the benefits incurred from audit have proven to be twofold, namely detection and deterrence (Tennyson & Salsas Forn, 2002). Current auditing research is mostly geared towards maximizing fraud detection through algorithms aimed at detecting the fraudsters. However it has been observed that after an audit, service providers also deter from inappropriate behavior in future (Tennyson & Salsas Forn, 2002). We refer to the direct cost difference due to the audit as the “audit effect”.

In addition, once a set of providers is audited, information about audit and sanctions diffuses to other providers and these (additional) “audit aware” providers make extra effort to ensure billings are accurate. This change in provider behavior triggered by the audit and the sanction of other providers in the network is referred to as the “sentinel effect” (Thornton, 1999).

Current audit practices however fail to capture the audit effect and the sentinel effect present in the provider network. By taking these effects into account, we design audit algorithms to help reduce costs related to fraud, waste and abuse in the long run.

To our knowledge, this research is the first to consider the sentinel effect of the deterrence provided by audit information diffusion, while designing an auditing algorithm to reduce fraud, waste and abuse. This is a significant contribution to the literature on audit and information systems.

In order to study the effectiveness of new auditing algorithms in these domains, we need to model how information diffuses as well as how it then affects provider behavior. Hence we first present a model of stochastic audit information diffusion and behavioral change based on prior literature and collaboration with domain experts.
We then examine prior audit algorithms and design a novel algorithm to maximize the audit and sentinel effects. Since real audit data is confidential and therefore hard to obtain, we test and evaluate our algorithm using an agent-based simulation under stochastic information diffusion in different network structures. Results from both healthcare and automotive domains are presented and discussed followed by analytical results for an auditing game between the insurance company and the service provider. We then also present some results based on real audit data in a warranty fraud in manufacturing, which helps provide some ballpark values for how much waste and abuse might be there in that domain. The problem setup, algorithm presented and analytical results are all unique research contributions. In addition, the potential of such techniques to reduce the high service costs suggests important practical significance in several industries.

**Underlying Theory and the Diffusion-Deterrence Model**

**The Provider Network**

Service providers are socially connected in a network where information and behavior are prone to propagate over time. Such social relationships are well-known to impact innovation adoption (Mahajan, Muller, & Bass., 1990), knowledge transfer (Sales et al. 2010), and behavioral change (Centola, 2010).

Specifically in healthcare, a motivating domain for this work, prior research has examined how connections between medical providers influence their prescription behavior. The seminal work of Coleman, Katz, and Menzel in 1957 indicated the presence of influence among physicians. In their original study a new drug was introduced to a few physicians in four different US cities. Fifteen months later the drug was adopted by a vast majority of doctors within the same specialties. The study had shown that interpersonal relationships among doctors do in fact allow diffusion of information and consequently influence behavior. Subsequent studies (Bulte, Christophe, & Lilien., 2001; Burt, 1987; Strang & Tuma, 1993; Valente, 1995) have explored the structure of the medical provider network and the nature of relationship among doctors.
Unlike some industries where service providers form more formal networks (e.g., dealers under the same manufacturing company), doctors tend to form more informal networks through geographical proximity, collaboration, and conference attendance (Perez et al., 2005). Whether these are formal or informal is less relevant for our context. What matters is that there exists a network that allows for sharing information and business decisions. Therefore, we assume the following (the convention in this section will be to number and italicize all the main assumptions behind the model assumed prior to audit).

1. **There is a network of providers (G) affiliated with the same insurance/warranty company.**

2. **Edges (E) between service providers are created whenever there is potential for direct audit information diffusion.** In practice, connections are based on a composite measure related to provider attributes such as office location, nature of business, common hospital privileges and geographical region, all known in practice to contribute to providers sharing information through word of mouth.

This paper does not focus on learning these networks from data or through other mechanisms. We assume such a network exists (in the paper we do consider various structural forms based on evidence from each domain). In practice too, both health insurers as well as manufacturing firms such as Ford or Caterpillar have extensive knowledge of both physician and dealer networks and are known to use this as part of their sales strategies.

**Audit Information Diffusion**

In the presence of a medium for information diffusion, we posit that audit and sanction information will spread among service providers.

Since providers form an offline network, information may or may not be transmitted from one node to another. Therefore, we assume that audit information between medical practitioners spreads stochastically.

3. **We assume that once some nodes are audited information, about the audit diffuses stochastically with some decay.**
Once any node N’s neighbors receive the audit information they will move from a state of being “unaware” of the audit to an “aware” state. If node N is at “level” t (where the level represents distance from the main audited node) then directly connected neighbors at level t+1 after moving to the aware state, will diffuse the information to the neighboring nodes (i.e., providers) with a probability \( P_{t+1} \). Note that any audited node is at level \( t = 0 \) and the initial probability of information diffusion is \( P_0 \). Given that there is now some distance between the original audited node and this current “aware” node, and that the information is now no longer “first hand”, we assume decay in information diffusion, where we expect \( P_{t+1} \) to be smaller than \( P_t \). We therefore define a decay factor \( \lambda \) such that,

\[
P_{t+1} = \lambda \cdot P_t \quad \text{Where} \quad 0 < \lambda < 1 \tag{1}
\]

This process continues sequentially, where the “new” aware nodes get a shot at propagating the audit information. The process stops when there are no new aware nodes. We note that in this study, we do not model awareness due to mass media coverage of fraud cases. In such cases, we can adjust awareness universally in the network and use the models presented here.

Also, we expect the diffusion probabilities to differ based on the audited provider’s initial level of compliance. Because audit of fraudulent providers results in sanctions, we posit that the information about fraudulent providers will diffuse with a higher probability. Also, because the sanctions imposed are a percentage of the amount of fraud, we expect that information about audit of fraudulent providers with high deviations from the norm will diffuse with higher probability when compared to fraudulent providers with low claims distribution deviation.

4. **We assume that information about audit of fraudulent providers will diffuse with higher probability when compared to audit of non-fraudulent providers.**

Therefore, we set the probability of diffusion of audit information to immediate neighbors at time t (\( P_t \)) as \( P_a, P_b, \) or \( P_c \), depending on the node’s fraudulent state and level of the claims deviation from the norm (which in the healthcare practice is measured using actual claims distributions by specialty and region) as shown in Figure 1 below.
Where $P_a > P_b > P_c$

- $P_a$: Probability of audit information diffusion in case of audit of fraudulent provider with high deviant claims amount
- $P_b$: Probability of audit information diffusion in case of audit of fraudulent provider with low deviant claims amount
- $P_c$: Probability of audit information diffusion in case of audit of a non-fraudulent provider

A node becomes audit aware as soon as the first signal is received. The time period of analysis is assumed to be small, such that once a node is in an aware state, it remains in the aware state during that time period, and that additional signals received have no effect on the node state or behavior. Further, any audit/aware node's claim submission behavior is independent of the number as well as the strengths of the signal received.

The basic model for stochastic information diffusion presented here, which is derived from knowledge extracted from the experts in practice, addresses the process by which nodes become audit aware. This does not specify yet how these providers then act on such information. We turn our focus next to this issue and address how the behavior change mechanism (deterrence) is modeled.

**Deterrence – Behavior Change**

Prior literature suggests that audit information has a significant impact on altering negative behavior. “If (a) a set of criteria was introduced, (b) a pending audit against these criteria was announced, and (c) penalties for nonconformance were established, explicitly, then a behavioral change would take place”(Churchill, Cooper, & Govindarajan, 1982). That is, as we might expect, once fraudulent providers receive information about audit and sanction they are likely to alter their behavior.

**5. We also assume that the claims submission behavior of a node during the time period of analysis, is independent of the number or the strength of audit information signals received from one or more audited nodes in the network.**

A node becomes audit aware as soon as the first signal is received. The time period of analysis is assumed to be small, such that once a node is in an aware state, it remains in the aware state during that time period, and that additional signals received have no effect on the node state or behavior. Further, any audit/aware node's claim submission behavior is independent of the number as well as the strengths of the signal received.

The basic model for stochastic information diffusion presented here, which is derived from knowledge extracted from the experts in practice, addresses the process by which nodes become audit aware. This does not specify yet how these providers then act on such information. We turn our focus next to this issue and address how the behavior change mechanism (deterrence) is modeled.

**Deterrence – Behavior Change**

Prior literature suggests that audit information has a significant impact on altering negative behavior. “If (a) a set of criteria was introduced, (b) a pending audit against these criteria was announced, and (c) penalties for nonconformance were established, explicitly, then a behavioral change would take place”(Churchill, Cooper, & Govindarajan, 1982). That is, as we might expect, once fraudulent providers receive information about audit and sanction they are likely to alter their behavior.

**6. Upon being audit aware, providers alter their claims submission behavior.**
There are some interesting broader connections for this notion. Jeremy Bentham is known for the idea of panopticism, a mechanism of subtle control through surveillance. While he initially introduced this in the context of the design of a prison, it has since been applied in many other contexts to broadly capture the effect of good behavior when being watched. Anechiarico and Jacobs (1984) specifically discuss panopticism and financial controls and note that audit leads to good behavior and deterrence, since entities are aware that they are being watched. Our assumption here is no different. Such an effect is also due to a major phenomenon, referred to as social learning. Social learning, as stated by Young in 2009, refers to the behavioral changes caused by observation of the outcome. Once a fraudulent provider is audited and sanctioned, neighboring providers would consider the containment of fraud and abuse.

Individual providers might of course respond to the audit information differently depending on their relationship to the audited provider, as well as their personal characteristics such as initial level of compliance and risk tolerance. After being audited, or aware of the audit, fraudulent providers are anticipated to reduce their claims deviation. Based on prior literature (Dionne, Giuliano, & Picard, 2009), providers with low risk tolerance are expected to deter with a higher probability when compared to high risk-tolerant providers.

6-a. *Fraudulent providers lower the amounts of claims submitted depending on their individual level of risk tolerance.*

Assuming fraudulent providers initially increase their real amount of submitted claims by a specific mark-up, they are likely to drop this mark-up amount entirely after being audited, or drop the mark-up partially after receiving the audit information.

Hence, if a fraudulent provider is audited then:

\[
\text{Submitted Claims Amount}_{\text{final}} = \text{Submitted Claims Amount}_{\text{initial}} - \text{MarkUp}_i
\] (2)

If a fraudulent provider is not audited directly, but is now audit-aware then:

\[
\text{Submitted Claims Amount}_{\text{final}} = \text{Submitted Claims Amount}_{\text{initial}} - (R_i \cdot \text{MarkUp}_i)
\] (3)
Where, $R_i$ is the level of risk tolerance of provider $i$, $0 \leq R_i \leq 1$. Interestingly, from conversations with domain experts, we expect non-fraudulent providers to also decrease the amount of submitted claims after audit. With waste and abuse exceeding 45% of excess costs ($340billion out of $750billion)(Smith et al., 2012), waste and abuse appears to be widely spread across the network. Therefore, while non-fraudulent providers might only change behavior by a small fraction through reduction of waste and abuse, their deterrence is expected to be more prevalent in the network.

6-b. With probability $P_d$, non-fraudulent providers decrease their claims submission amount by reducing waste and abuse.

In practice, the average extent and magnitude of waste and abuse differ by line of work and are known to auditors. Therefore we formulate the deterrence from waste and abuse as follows:

With a probability $P_d$,

$$\text{Submitted Claims Amount}_{\text{final}} = (1 - \alpha) \cdot \text{Submitted Claims Amount}_{\text{initial}}$$

where $0 < \alpha < 1$ (4)

Where $P_d$ represents the extent and $\alpha$ represents the magnitude of waste and abuse present in the network. Deterrence can therefore be directly calculated at both the individual level as well as the network level by calculating the difference between the initial and final submitted claims amount.

High Level Set-up

The Audit Problem: Given a network of providers $G = (V, E)$, where $V$ is a set of nodes representing providers, $E$ is the set of edges representing interconnections between providers, and $F \subseteq V$ is a set of fraudulent providers, select a set of providers $K \subseteq V$ for audit, so as to maximize (1) $|K \cap F|$ and (2) the Total Network Deterrence, which represents the drop in claims amount in the network after auditing the $|K|$ providers selected by the algorithm.

Based on our discussion, the high level audit diffusion and behavioral change context in which this problem is to be solved is summarized below.

1. We assume a network of providers (G) affiliated with the same insurance company.
2. Edges (E) between service providers are created whenever there is potential for direct audit information diffusion. In practice, connections are based on a composite measure related to provider attributes such as office location, nature of business, geographical region and common hospital privileges, in the healthcare domain.

3. We assume that once some nodes are audited information about the audit diffuses stochastically with some decay.

4. We also assume that information about fraudulent providers audit will diffuse with higher probability when compared to audit of non-fraudulent providers.

5. The claims submission behavior of a node during the time period of analysis is independent of the number or the strength of audit information signals received from one or more audited nodes in the network.

6. Upon reception of audit information, providers alter their claims submission behavior.
   a. Fraudulent providers lower the amounts of claims submitted depending on their individual level of risk tolerance, $R_i$.
   b. With a probability $P_d$, non-fraudulent providers decrease their claims submission amount by reducing waste and abuse.

Using data from the practice, we assume

7. There exists a set of fraudulent providers (F), and the rest of the network population is non-fraudulent.

8. The only way to determine the state of the provider as being fraudulent or non-fraudulent is through audit.

9. There is an overall claims distribution from which individual nodes submit claims. We assume though that the fraudulent nodes then markup the claims in some manner that increases billing.
10. Insurance/warranty companies use a scoring algorithm to calculate a prior probability of fraud for each node based on individual attributes and claims submission bell curves by line of work and region.

Once the network of providers is defined and the process of audit information diffusion and behavior change is set up, we design algorithms to select providers for audit.

**Audit Algorithms - Background**

Investigating all claims submitted by providers is often not cost-effective. Rather insurance/warranty companies employ different procedures to select some provider claims for audit. In practice, several independent entities provide audits to detect and/or prevent fraudulent behavior. According to the National Healthcare Anti-fraud Association (NHCAA), medical investigations include on-site audits, equipment audits, mail-order reviews, claims check, analytics and reporting, product verification, compounding, member lock-in, physician profiling, and credentialing programs. These programs aim at detecting and recovering known types of improper activity. Other practices use predictive statistical models utilizing scoring rules, anomaly detection, predictive modeling and social network analysis techniques in order to optimize detection. Given their proprietary nature, these algorithms are often unknown to the academic world.

Academic auditing research on the other hand has primarily focused on developing scoring algorithms to determine suspicious providers. Based on individual-specific variables and claim-related signals (Dionne et al., 2009), these algorithms calculate suspicion indices (scores) used to select claims for investigation. Traditionally, audit algorithms generate a fraud probability associated with every claim and/or provider. The pre-selected providers are then targeted for audit. Hence, work in the general area of fraud detection can be relevant for audit algorithms (Fawcett & Provost, 1997; Phua, Lee, Smith, & Gayler, 2010). However, there is little work that has explicitly addressed fraud detection models for provider audit applications in healthcare.
One interesting stream of work in IS has been active learning, which can be used by auditing providers to acquire their “fraud” flag (Saar-Tsechansky & Provost, 2007). In such cases, the information acquired from the audit is traditionally used to improve the detection model. However Saar-Tsechansky & Provost (2007) and Kong and Saar-Tsechansky (2014) address decision centric active learning, where the information acquired is also considered based on its utility in a broad sense. Our work here can be viewed as one specific kind of active learning that suggests a new measure of utility that includes the audit effect and the sentinel effect, prevalent in certain kinds of major sectors such as healthcare and warranty. Our approach is therefore a contribution that has not been considered even in the active learning literature. Indeed recent views have recognized the importance of the audit role in deterring fraudulent behavior rather than simply detecting it. Tennyson et al. in 2002, note that “the primary role of auditing of an optimally designed system is the deterrence of buildup rather than its detection.” (Tennyson & Salsas Forn, 2002). This is consistent with the ideas presented next.

**Audit under the Sentinel Effect**

Rather than looking to detect and minimize fraudulent behavior alone we aim at maximizing deterrence as well by considering the behavior change of many audit-aware providers in the network. This can reduce the costs in the system not only due to fraud, but also due to the reduction in waste and abuse that are likely to be more prevalent. This perspective guides the design of our deterrence-based algorithm.

**Algorithm Outline**

Considering the fact that audit information (1) diffuses in the network, and (2) triggers deterrence, the audit of service providers could be classified as an influence maximization problem. The problem is then to target the set of providers for audit that produce the largest deterrence cascade. Though similar to models such as the Independent Cascade Model developed by Kempe et al 2003, our model is different in many aspects. First, the provider network is composed of different types of nodes - fraudulent or non-fraudulent. Second, the diffusion of audit information decays over time.
Third, once a provider changes behavior (deters), others in the network cannot observe this change; and thereby cannot be influenced to similarly deter. Rather, deterrence occurs following a two-step process. First, audit information diffuses to providers in the network. Then, providers alter behavior (deter) depending on their individual characteristics such as compliance category and risk tolerance. Due to these reasons, existing influence maximization algorithms do not apply directly in this auditing context, although our approach presented here does use the salient ideas in such algorithms.

We consider a service provider network where (1) providers are affiliated with the same insurance/warranty company, (2) providers are of two types (fraudulent and non-fraudulent), and (3) the insurance/warranty company is aware of the provider network and the claims amount distribution (they routinely gather and analyze such information). In the provider network, we select providers for audit such that deterrence is maximized (as formulated in Section 3.1).

The audit problem under the sentinel effect is an influence maximization problem in which nodes are heterogeneous, probabilities of diffusion decay over time, and behavioral change depends on individual node characteristics. Since the Influence Maximization problem is NP-hard (Kempe, Kleinberg, & Tardos, 2003), the audit problem here is also NP-hard. Hence we develop a Greedy Deterrence Heuristic.

By making use of the structure of the provider network, the “Greedy Deterrence Heuristic” aims at maximizing overall deterrence. The algorithm:

1. Estimates the fraud probability for each provider in the network
2. Calculates the Network Deterrence value (the individual drop in claims amount in the after audit) of each provider in the network (described more below).
3. Selects a set of providers K for audit such that
   (a) the total network deterrence value is maximized, and
   (b) the overlap among diffusion effects is minimized.
Note that our algorithm is similar to the active learning approach in the sense that a) nodes are selected for audit sequentially, and b) nodes are selected based on the expected deterrence they would generate in the network.

That is, a specific node can have a relatively smaller fraud probability – score (still higher than the normal range) and yet selected for audit because it is highly connected and it is expected to generate more deterrence in the network. A major distinction in our approach is that we are taking into consideration the relationships between nodes in the network. More specifically, nodes are selected for audit in a way that minimizes overlap of expected nodes reached through audit information diffusion.

| Input: A network of providers G, number of providers k, prior fraud probabilities \( P_i \) for all providers |
| Output: A set of k nodes to be targeted for audit |

1: Begin
2: while (k > 0) do
3: for each node in the network do
4: Calculate Expected Network Deterrence Value (DV)
5: Sort vertices based on their DV
6: Select node \( i \) with the highest DV for audit
7: Remove node \( i \) from the list of providers
8: Remove node \( i \)'s immediate neighbors from the list
9: \( k = k-1 \)
10: end while
11: END

Figure 2: The Greedy Deterrence Heuristic Algorithm

**Expected Network Deterrence Computation**

The network deterrence value of a node varies based on how the stochastic diffusion process occurs. Since individual risk tolerance, mark-ups, and magnitudes of waste and abuse are unknown to insurance/warranty companies, these cannot be used to determine the actual deterrence value of every node in the network.
Hence, it is necessary to measure instead, the expected network deterrence value of a node. This can be determined computationally by averaging over values generated by running stochastic diffusion processes a large number of times from each node.

An alternative to this expensive computational measurement is an approximation for this expectation that can be computed based on the number of nodes that are expected to be audit aware at any level. Recall that the diffusion probability decays with levels and this can be used to compute the expected percentage of nodes that are aware at any level. For example, if $P_t = 0.9$ and $\lambda = 0.5$, then 45% of level two neighbors are expected to be aware of the audit. If $\$100K$ is the total deterrence that can be expected at that level if "all" nodes are aware, then $\$45K$ is the contribution of this level to the approximation for the expected deterrence of a node. We formalize this below:

\[
\text{Expected Network Deterrence (i)}
= \text{Expected Detection Value (i)}
+ \sum_{\text{Level}=1}^{n} \text{Deterrence Value}_{\text{Level}(i)} \cdot P_{\text{Level}-1}
\]  

(5)

Where \( n \) refers to the diameter of the audit graph, \( \text{Deterrence Value}_{\text{Level}(i)} \) is the amount expected to be saved if all nodes that are at a distance of "Level" from the chosen node "i" are audit aware, and \( P_{\text{Level}-1} \) is determined from (1). If \( J \) is the set of nodes that are at a distance of "Level" from the chosen node "i", then the \( \text{Deterrence Value}_{\text{Level}(i)} \) can also be written as \( \sum_{j \in J} \text{Expected Aware Value} (j) \). Drawing from equations 2 through 4, the \( \text{Expected Aware Value} \) of each aware node "j" would be set as follows:

\[
\text{Expected Aware Value (j)}
= P^f(j) \cdot [R_j \cdot \text{MarkUp (j)}] + (1 - P^f) \cdot [P_d \cdot \alpha \cdot \text{Submitted Claims Amount}_{\text{initial}} (j)]
\]  

(6)

Similarly, the \( \text{Expected Detection Value} \) of the audited node "i" could be calculated as follows:

\[
\text{Expected Detection Value (j)}
= P^f(j) \cdot [\text{MarkUp (j)}] + (1 - P^f) \cdot [P_d \cdot \alpha \cdot \text{Submitted Claims Amount}_{\text{initial}} (j)]
\]  

(7)
Where $P^f$ represents the node’s prior fraud probability. Any scoring algorithm can be used to determine each node’s own prior fraud probability.

These scores are created based on individual attributes as well as income comparisons with bell curves by line of work and region. In the medical domain, bell curves are publicly available from the Center for Medicare and Medicaid Services (CMS). Note that we do assume away the hard task of calculating the prior fraud probability, which most prior IS research has focused on through design of fraud detection algorithms. Two reasons for this are: (a) it is not the focus of this paper, we can indeed use prior fraud probability if provided by any other method as well, (b) industry uses this based on deviation from the bell curves as we noted above. Note that our greedy deterrence heuristic does also take a node’s own detection value into account. This algorithm therefore has detection combined in it as well, since selecting a node that is fraudulent will generate detection benefits from auditing that node.

However, even though the use of mark-up amounts, risk tolerance ($R_j$) and the exact extent ($P_d$) and magnitude of waste and abuse ($\alpha$) values would result in very accurate estimates of the nodes’ expected network deterrence, these values are usually unknown to the insurance provider, and therefore cannot be used by the algorithm to select providers for audit. A surrogate for these values could be set as follows based mainly on two factors – the submitted claims amount and an estimated prior fraud probability:

\[
\text{Expected Detection Value (j)} = \text{Expected Aware Value (j)} = \text{Submitted Claims Amount}_{\text{initial}} (j) \cdot P^f (j) \tag{8}
\]

This surrogate is directly measurable based on known factors but clearly does not take into account the factors just discussed above. However, as we show in the results this still provides significant value.

Finally, a measure of signal strength can easily be added to the model. In the present model as per assumption 5 these are not used (i.e., the signal strength and signal decay are both set to 1). Calculating this expected network deterrence value for each node is at the heart of the algorithm. While the equation presents how this can be determined, there are additional steps algorithmically that we now note (for ease of exposition this was not presented in figure 2). We consider nodes one at a time to calculate their expected network deterrence value.
First, the diameter of the provider graph is determined, in order to store how many possible levels there are from each node. In many real-life networks, this is known to be fairly low (Guare, 1990; Watts, 1999).

Next we determine the actual set of nodes that can be reached from the current node at each distance value (betweenness centrality). This then provides all the information needed to compute the expected network deterrence value for each node.

Common traversal algorithms such as breadth-first search (BFS) and Dijkstra's algorithm can be computationally expensive depending in the number of nodes (|V|) and the number of edges (|E|). However, newer algorithms take advantage of new accumulation techniques and multi-processing to make the process more efficient. Using the Brandes' algorithm (Brandes, 2001), betweenness centrality calculations require O(|V|+|E|) space and O(|V|.|E|) and O(|V|.|E| + |V|^2 log |V|) time on un-weighted and weighted networks instead of the traditional complexity of O(|V|^3) time and O(|V|^2) space. Using Hadoop and MapReduce, the HADI algorithm can compute the graph diameter in O(d(|V| + |E|)/M) time and O((|V| + |E|) log |V|) space, where M represents the number of machines in the MapReduce or Hadoop cluster, and d is the number of iterations required to complete the process (Tsourakakis, 2008).

**Input:** A network of providers G, A node i in G, P, λ

**Output:** Node (i)’s expected deterrence value (DV)

**Method:**

1. **Begin**
2. Calculate the graph diameter from node i
3. Set MaxLevel to the graph diameter
4. **for** (level=1 ; level< MaxLevel; level++)
5. Use Adjacency Matrix to retrieve P:node (i)’s neighbors at level “level”
6. **for** (j=0; j<|P|; j++)
7. set Deterrence Value_{Level}(i) += Submitted Claims Amount_{initial} (j). P_f(j)
8. end for
9. DV+= Signal_{Level}.Deterrence Value_{Level}(i). P_i. λ^{level−1}
10. end for
11. Calculate node (i)’s Detection Value (i) = Submitted Claims Amount_{initial} (i). P_f(i)
12. DV+= Detection Value (i)
13. return DV
14. **End**

---

**Figure 3. The Calculate Expected Network Deterrence Value (DV) Function**
Table 1 contains a summary of variables used in the auditing process.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_t$</td>
<td>The probability of audit information diffusion at time t</td>
<td>Probability that decays over time</td>
</tr>
<tr>
<td>$P_a, P_b, P_c$</td>
<td>The probability of audit information diffusion based on deviation and fraudulent state</td>
<td>Information about the audit of fraudulent, high-deviant nodes diffuses vastly</td>
</tr>
<tr>
<td>$\Lambda$</td>
<td>Diffusion decay</td>
<td>Information diffusion decays with distance</td>
</tr>
<tr>
<td>$R_i$</td>
<td>The Risk Tolerance of node i</td>
<td>Individual variance of deterrence among fraudulent nodes</td>
</tr>
<tr>
<td>$P_d$</td>
<td>The probability of deterrence on non-fraudulent nodes.</td>
<td>The spread of waste and abuse in the network</td>
</tr>
<tr>
<td>$A$</td>
<td>The magnitude of deterrence of non-fraudulent nodes.</td>
<td>The amount of waste and abuse in the network (Does not include fraud)</td>
</tr>
<tr>
<td>$P_i'$</td>
<td>The prior probability of fraud of node i</td>
<td>Score assigned to each node based on individual attributes and CMS bell curves</td>
</tr>
<tr>
<td>$DV_i$</td>
<td>The expected deterrence value of node i</td>
<td>Node i’s expected deterrence after audit</td>
</tr>
<tr>
<td>K</td>
<td>Set of nodes selected for audit</td>
<td>Actual output of the algorithm</td>
</tr>
<tr>
<td>F</td>
<td>Set of fraudulent nodes in the network</td>
<td>Unknown to the insurance company</td>
</tr>
</tbody>
</table>

To evaluate the economic value from the algorithm, we focus primarily on the change in the overall amount of claims submitted by providers in the network before and after audit. This does incorporate waste and abuse reduction. We discuss in greater detail when we present the results.
Results

We present results relevant to two different domains, healthcare and automotive manufacturing. In collaboration with a medical auditing company we create an agent-based simulation, evaluate the performance of our greedy deterrence heuristic, and present results in Section 5.1.

We next present results from warranty claims data working in collaboration with a different company that studies warranty audit to illustrate the audit and sentinel effects in that domain.

Healthcare Results

We implement two methods for selecting providers for audit, 1) the deterrence heuristic presented in Section 4 and 2) a detection heuristic primarily focusing on detecting fraudulent providers.

Note that the detection heuristic implemented in the simulation uses the nodes’ Detection Value described earlier to select providers for audit \( Detection Value (j) = Submitted \ Claims \ Amount_{initial} (j) \cdot P'(j) \). Also, the deterrence heuristic implemented uses the adjacency matrix and a default diffusion level of 3 to calculate the Deterrence Value of each node in the network.

In the agent based simulation models we consider a network of 1000 providers in the healthcare domain. Two nodes are directly linked if there is a relationship between the two providers based on attributes such as co-location, common hospital privileges and physician specialty. The total amount of claims submitted has a mean of $500,000 and a standard deviation of $100,000.

Since outright fraud is relatively rare, each provider is assigned a random prior fraud probability following a zipf distribution. Those priors are then used to determine actual fraudulent providers before the agent-based simulation model runs. In order to capture the noise associated with prior fraud probabilities known to the insurance company, we introduce a shock as follows:

\[
\text{Shocked}_\text{Prob} = (w \times \text{Fraud}_\text{Prob}) + ((1-w) \times \text{Random}_\text{Prob}) \quad \text{where} \quad 0 < w < 1
\]

“Random_\text{Prob}” is a uniform random number, while \( w \) models the quality of the insurance company’s prior knowledge of the actual fraud probabilities. When \( w \) is high it represents a highly aware insurance company that understands most of its providers well from a fraud perspective.
To assess the effect of different settings on the performance of both algorithms (deterrence and detection heuristics), we set up scenarios with varying network topologies, levels of diffusion and decay, and extents of waste and abuse.

**Network Topology Influence**

In order to study the impact of network topology on the algorithms’ performance, we generated (1) a scale-free network using a power-law degree distribution following the Barabási–Albert model (Barabási & Réka, 1999), and (2) a random network using a uniform degree distribution.

Figure 4 indicates the effects of network topology on the deterrence of providers in the network. The likelihood of deterrence increased with the number of connections. Therefore, highly influential providers generated more deterrence in the network. Because the deterrence heuristic targeted high-degree nodes for audit, more providers were aware of the audit information; thereby, deterring more providers from fraud waste and abuse.

In a world of high diffusion low decay (Figure 4 - Cases A and B), audit information diffused to multi-level neighbors to reach a large proportion of the network. When waste and abuse was prevalent in this world type, deterrence was expected to be at a maximum regardless of the network topology. When diffusion was high and decay was high (Figure 4 - Cases C and D), audit information only reached immediate neighbors.

In a scale-free network, few practitioners are highly connected. On the other end of the spectrum, in random-uniform networks, a vast proportion of practitioners in the network have a relatively high number of immediate neighbors. In our simulation instance, the scale-free network included 114 practitioners with 10 to 20 immediate neighbors, and about 10 practitioners with 20 to 30 immediate neighbors. The random-uniform network on the other hand had 352 practitioners with 20 to 30 immediate neighbors and 353 practitioners with 10 to 20 immediate neighbors. Therefore, more deterrence was generated in random uniform networks when decay was low. The greedy deterrence heuristic generated $15M more than the detection algorithm in total network deterrence (Figure 4 C and D).
Figure 4. Deterrence vs. Detection Amounts (Network Topology)
Y-axis: Network Deterrence Amount X-axis: Number of Audited Providers
(A) Scale-Free Network, High Diffusion, Low Decay, High Waste and Abuse Extent
(B) Random Uniform Network, High Diffusion, Low Decay, High Waste and Abuse Extent
(C) Scale-Free Network, High Diffusion, High Decay, High Waste and Abuse Extent
(D) Random Uniform Network, High Diffusion, High Decay, High Waste and Abuse Extent
(E) Scale-Free Network, Low Diffusion, Low Decay, High Waste and Abuse Extent
(F) Random Uniform Network, Low Diffusion, Low Decay, High Waste and Abuse Extent
(G) Scale-Free Network, High Diffusion, Low Decay, Low Waste and Abuse Extent
(H) Random Uniform Network, High Diffusion, Low Decay, Low Waste and Abuse Extent
In the case of low diffusion (Figure 4 E and F), the algorithms still performed better in the random-uniform network since more practitioners had larger number of immediate neighbors when compared to the scale-free network. Even when audit information, and thus deterrence, only reached about 20% of immediately connected neighbors, that amount was still higher in the random-uniform network. When the extent of waste and abuse was low, deterrence was minimal after the reception of audit information. Therefore, the network deterrence amount was substantially lower in the world type for both the deterrence and detection algorithms, in both scale-free and random-uniform networks.

The practical significance of this comparison is that deterrence based algorithms are likely to be effective if the provider word-of-mouth network was scale free. Given the wide range of networks shown to be scale free, this is likely the case in practice as well.

**Diffusion**

Network deterrence was affected dramatically by the likelihood of diffusion in the network. Figure 4 (Cases B, D, F, and H) contain plots of the network deterrence amounts for both the deterrence and detection algorithms in a world of low diffusion (average diffusion probability set to 0.2). The greedy deterrence heuristic generated approximately fivefold lower deterrence in low diffusion settings compared to high diffusion. The detection algorithm had over a threefold decrease in deterrence amounts in varying diffusion settings (Figures 4). In this type of settings, the detection heuristic outperformed the deterrence by detecting more fraudulent and deviant practitioners.

In the presence of high levels of diffusion, service providers had a tendency to spread information after the audit. In this specific setting, audit information diffused to immediate neighbors with a high probability, generating vast amounts of audit awareness, and consequently large amounts of deterrence in the network. Hence in high diffusion scenarios, the deterrence heuristic had significant economic value (Figure 4 Cases A, C, E, and G).
Figure 5: Deterrence vs. Detection Amounts (Diffusion, Decay and Waste and Abuse Extent)

Y-axis: Network Deterrence Amount  X-axis: Number of Audited Providers

(A) Scale-Free Network, High Diffusion, Low Decay, High Waste and Abuse Extent
(B) Scale-Free Network, Low Diffusion, Low Decay, High Waste and Abuse Extent
(C) Scale-Free Network, High Diffusion, High Decay, High Waste and Abuse Extent
(D) Scale-Free Network, Low Diffusion, High Decay, High Waste and Abuse Extent
(E) Scale-Free Network, High Diffusion, Low Decay, Low Waste and Abuse Extent
(F) Scale-Free Network, Low Diffusion, Low Decay, Low Waste and Abuse Extent
(G) Scale-Free Network, High Diffusion, High Decay, Low Waste and Abuse Extent
(H) Scale-Free Network, Low Diffusion, High Decay, Low Waste and Abuse Extent
Decay (Cases A and D)

Comparing the performance of the auditing algorithms in settings of low versus high decay showed the effects of provider relationships on the deterrence amounts. In the case of high decay (Figure 5 A and C), information only diffused to immediate neighbors generating less audit awareness, and correspondingly less deterrence. In low decay settings, as the number of audited providers increased, audit information was transmitted to multi-level neighbors. Therefore, most providers in the network became audit-aware with low number of audited providers, thereby generating more deterrence and causing the network deterrence amounts to level off (Figure 5A).

Extent of Waste and Abuse (Cases A, E, and F)

As indicated by the simulation results the deterrence amounts were also affected by the extent of waste and abuse in the network. When waste and abuse was prevalent in the network, practitioners were expected to deter after receiving the audit information. Therefore more deterrence was generated in the world of high waste and abuse. As the extent of waste and abuse dropped from high to low, the network deterrence amounts decreased. In a world of high diffusion, the deterrence amount dropped from about $45M to $6M (Figure 5 A and E). When diffusion was low, the deterrence amount dropped from about $25M to $4M for the greedy deterrence heuristic, and from about $10M to $2M for the detection algorithm (Figure 5 C and G).

Detection versus Deterrence

Comparing the performance of the deterrence and detection auditing algorithms in various settings suggests the applicability and trade-offs associated with their use. Looking at precision and recall numbers (Table 2), we observed as expected that the deterrence heuristic had a low detection rate when compared to the detection algorithm. Hence, while the detection heuristic performed better in detecting fraudulent providers, the greedy deterrence heuristic provided higher economic value in most cases.
For some important context considered in this analysis, we note that current auditing algorithms in the industry use scoring mechanisms to target fraud and appear to mainly aim at maximizing the detection of fraudulent claims and providers. While high detection may generate large immediate revenues from the collection of fines, detection algorithms were not likely to produce the largest deterrence in the network, as our results indicated.

Below we present additional analyses that offer more practical insight. Before this, we note that in all the cases presented above, as is standard in simulations, the total deterrence amount was based on averaging multiple runs, where a certain number of providers were audited, and hence these amounts do represent expected values. However in real-life when a certain number of providers are audited and there is stochastic diffusion, the end result is a specific set of providers who are “aware”, along with their real claims change behavior. Reality does not provide an option for multiple runs as simulations do.

To test which algorithm would have worked better “most of the time” in reality, we presented numbers in Table 2 that indicate the percentages where the greedy deterrence heuristic outperformed the detection algorithm over 10,000 runs in the scale free network case. In each cell we also presented two numbers, corresponding to auditing very few (3) versus high (30) number of providers. For instance, in the cell corresponding to Low WA and High Diffusion-Low Decay, the greedy deterrence heuristic resulted in a better economic outcome in 100% of cases when 3 providers were audited. When thirty providers were audited, 74% of the time the greedy deterrence performed better.

<table>
<thead>
<tr>
<th>Table 2: Greedy Deterrence Heuristic’s Percentage Wins</th>
</tr>
</thead>
<tbody>
<tr>
<td>High Diffusion</td>
</tr>
<tr>
<td>Low Decay</td>
</tr>
<tr>
<td>Low WA</td>
</tr>
<tr>
<td>Med WA</td>
</tr>
<tr>
<td>High WA</td>
</tr>
</tbody>
</table>
A few salient observations from this table are noted below.

- Overall, as expected, the greedy deterrence heuristic performed best in the case of high-diffusion/low-decay (WA represents Waste & Abuse).
- For any combination of conditions involving decay and diffusion, deterrence “won” as the waste and abuse extent increased; something that is intuitive.
- Since the network was scale-free, the high diffusion scenarios favored auditing based on deterrence to a significant extent.
- Finally, when there was very little waste and abuse, and very little diffusion, then clearly the economic value came from targeting mainly the expected fraudulent providers. It was only in this scenario (top row, right columns) that deterrence did poorly on a consistent basis. In such cases, traditional fraud detection algorithms could be used quite effectively.

**Case Study: Warranty Claims**

We also analyzed real claims data in collaboration with domain experts in the auto warranty domain where we were able to evaluate the performance of the greedy deterrence heuristic by combining this data and agent-based simulations.

**Audit Claims Data Analysis**

Spanning a period of 4 years (December 2008 to December 2012), the data collected pertained to ten different dealers having warranty claims submitted to the same manufacturer. Audit visits were conducted during the time frame and claims data were analyzed before and after audit. Each dealership comprised several branches, however only one of the branches was audited within each dealership.

Analysis of claims data (Figure 5) showed a decline in claim submission amounts after audit for three dealerships (30% of the sample) considered for the case study, illustrating the direct audit effect. One out of these three dealerships acknowledged having submitted exaggerated claims before the audit as per their customers’ request.
Domain experts confirmed that waste and abuse was highly probable in the case of the remaining two dealerships. Although a small sample, these results can provide crucial input to agent-based simulation models which can be modified to reflect statistics from real data.

Network Topology

In the automotive industry dealers form cliques that operate under the same parent company. Same-clique dealers could also possibly form random connections with other dealers through conferences and geographic proximity. We therefore designed a dealership network specific to providers in this domain. This network had several cliques connected by random connections.

![Dealership 1]

![Dealership 2]

![Dealership 3]

Figure 6. Audit Claims Data
Model Modification

Because of the nature of the dealers’ network structure, the basic diffusion model was modified to better fit the auto warranty domain as noted below.

- Diffusion probabilities did not vary by fraud status and income deviation. Audit information is supposed to diffuse with a very high probability across all branches of the same dealership, regardless of fraud or deviation. This is the case, since the branches under the same ownership, have significant incentives to share audit information.

- There was a low diffusion probability across cliques since dealerships were considered competitors and were not anticipated to necessarily share audit information.

- Waste and abuse were correlated with fraud: Within cliques where fraud existed, waste and abuse was also expected to be significantly higher.

Simulation Results

As before, we created a 1000-dealers network. The network included cliques of 3 to 7 dealers. Dealers in each clique were fully connected. 10% additional connections were created to connect cliques through randomly selected providers. Deterrence and deviation algorithm performances were analyzed under different scenarios of varying inter-clique diffusion, decay, and waste and abuse extent.

Overall simulation results showed that the deterrence and detection algorithms performed comparably well in the dealership network (Figure 7). In comparison with the network deterrence amounts realized in the scale-free and uniform networks (Figures 3 and 4), we observed a much tighter gap between the algorithms performances in the dealership network. That is because the vast majority of nodes in the dealership network were highly connected; Dealers within the same clique were fully connected. Also, diffusion among dealers within the same clique was very high (set in the simulation to 0.9). Therefore, while targeting highly deviant nodes for audit, the detection algorithm also achieved the unforeseen benefit of the deterrence of neighboring within clique nodes.
However, since audit information diffused similarly regardless of the nodes fraudulent status or deviation amounts, the detection algorithm did not outperform deterrence algorithm. Because the deterrence algorithm targeted highly connected (and deviant) nodes, it tended to select nodes that were connected to more than one clique. Hence, the deterrence algorithm triggered both inter and intra clique diffusion of audit information. However, when the inter-clique diffusion probability was low (Figure 7 A, B, C, D), the audit information did not diffuse to neighboring cliques resulting in marginal additional deterrence benefits.

When the inter-clique diffusion probability was high (Figure 7 E, F, G, and H), the audit information diffused to nodes from neighboring cliques, generating more deterrence. For instance, looking at the network deterrence amounts, we saw that the algorithms generated about $15M by auditing 30 providers in a low inter-clique diffusion scenario (Figure 7C), while the same algorithm realized about $20M in deterrence benefits in the high inter-clique diffusion case (Figure 7G). In cases of high inter-clique diffusion probability, we also observed a wider gap in performance between the two algorithms. That could be explained by the fact that the deterrence algorithm achieved benefit from auditing nodes connected to more than one clique.

Comparable to prior simulation results (Figures 3 and 4), we observed higher deterrence amounts in cases of high waste and abuse extent, and low decay. When decay was low (Figure 7 A, C, E, and G), the network deterrence amounts ranged from $7M to about $20M compared to a range of $6M to about $15M in the high decay scenarios (Figure 7 B, D, F, and H). Similarly, the network deterrence amounts ranged from $6M to $10M in the low waste and abuse extent scenarios (Figure 7 A, B, E, F) compared to a range of $12M to $20M (Figure 7 C, D, G, H) in the cases high and abuse extent.
Figure 7: Deterrence vs. Detection Amounts (Dealer Network)
Y-axis: Network Deterrence Amount  X-axis: Number of Audited Providers
(A) Low Inter-Clique Diffusion, Low Decay, Low Waste and Abuse Extent
(B) Low Inter-Clique Diffusion, High Decay, Low Waste and Abuse Extent
(C) Low Inter-Clique Diffusion, Low Decay, High Waste and Abuse Extent
(D) Low Inter-Clique Diffusion, High Decay, High Waste and Abuse Extent
(E) High Inter-Clique Diffusion, Low Decay, Low Waste and Abuse Extent
(F) High Inter-Clique Diffusion, High Decay, Low Waste and Abuse Extent
(G) High Inter-Clique Diffusion, Low Decay, High Waste and Abuse Extent
(H) High Inter-Clique Diffusion, High Decay, High Waste and Abuse Extent
Analytical Results

Grounding our work on prior game theory literature (Cavusoglu, Raghunathan, & Cavusoglu, 2009), we set up and analyzed an audit game model. We identify two players, the Insurance Provider (IP), and the Service Practitioner (SP). Among all practitioners within the network, ε practitioners were influential (with a number of neighboring providers exceeding a threshold predefined by the insurance provider), while the rest were not. Our model was similar to the “IDS-Firewall model” (Cavusoglu et al., 2009) in that the two players represented the firm and user, where the user elected to hack or not, and the firm decided to audit or not, based on output from an Intrusion Detection System (IDS) and firewall.

For modeling purposes, we considered the case of a particular set of an IP and a SP. The SP could elect to defraud with a probability ψ. Hence, the total claims amount could include an amount of fraudulent claims ρ (mark-up). Once audited and if found to be fraudulent, the SP was imposed a penalty γ. Therefore we identified the practitioner’s expected utility as follows:

<table>
<thead>
<tr>
<th>Table 3: Service Practitioner’s Payoff</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service Practitioner’s strategies</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Defraud</td>
</tr>
<tr>
<td>ρ – γ</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>Don’t Defraud</td>
</tr>
<tr>
<td>ρ</td>
</tr>
<tr>
<td>0</td>
</tr>
</tbody>
</table>

The IP handled claims submitted by the SP. In order to investigate the legitimacy of claims, the IP incurred the cost of audit (c). To select providers for audit we followed a two-step process (Figure 8). First, a detection algorithm was used to filter out all genuine practitioners (providers with detection values below a specific threshold). Those were practitioners who had been assigned low prior fraud probabilities and had submitted claims amounts within the industry’s norms.

Pre-selected likely fraudulent practitioners were then presented to the greedy deterrence heuristic which selected the group of practitioners with the highest Expected Network Deterrence Value for audit.
By combining both algorithms in this manner, the insurance provider obtained the benefit of targeting both fraudulent and influential providers. Also combining the detection and deterrence algorithms assured the adherence to some industries’ regulations which outlawed the random audit of providers without a likelihood of sustained or high level of payment error (Medicare Prescription Drug Improvement and Modernization Act, 2003). The IP therefore could choose to audit (1) practitioners \textit{targeted} by the deterrence algorithm with a probability $p_1$, or (2) practitioners \textit{not targeted} by the deterrence algorithm with a probability $p_2$.

**Figure 8: The Deterrence Algorithm**

Within the network we differentiated between two groups of practitioners: influential and non-influential. While all audited practitioners were expected to deter after audit, influential nodes which were highly connected nodes, were anticipated to trigger a larger diffusion of the audit information in the network, thereby generating more deterrence in the network. Therefore, we denoted the additional benefit from auditing an \textit{influential fraudulent} practitioner $\Phi_1$ and the additional benefit from auditing an \textit{influential non-fraudulent} practitioner performing waste and abuse $\Phi_2$. The deterrence algorithm aimed at selecting influential practitioners with the highest Expected Network Deterrence value ($DV$) for audit, thereby maximizing the insurance’s payoff. The insurance company’s payoffs for the different scenarios are expressed in Table 4.
Table 4: Insurance Provider's Payoff

<table>
<thead>
<tr>
<th>Service Provider's Strategies</th>
<th>Don't Audit</th>
<th>Audit</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Non-Influential Practitioner</td>
<td>Influential Practitioner</td>
</tr>
<tr>
<td>Defraud</td>
<td>$-\rho$</td>
<td>$-\rho$</td>
</tr>
<tr>
<td>Don't Defraud</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

In quest of utility maximization a service provider could elect to defraud, waste and abuse, or not, while the insurance provider could elect to either the audit the practitioners selected by the detection algorithm or audit the practitioner who were not tagged by the deterrence algorithm. The game is summarized in strategic format in Figure 9 below.

Figure 9: Game Tree

We represented the performance of the deterrence algorithm through the probabilities of true and false positives. We therefore defined the following metrics:
$P_F^I$: The probability that the **deterrence** algorithm generated an alarm for a **fraudulent influential** practitioner.

$P_F^{NI}$: The probability that the **deterrence** algorithm generated an alarm for a **fraudulent non-influential** practitioner.

$P_{NF}^I$: The probability that the **deterrence** algorithm generated an alarm for a **non-fraudulent influential** practitioner.

$P_{NF}^{NI}$: The probability that the **deterrence** algorithm generated an alarm for a **non-fraudulent non-influential** practitioner.

In order to derive the Nash equilibrium for the game, we calculated both parties’ expected payoffs ($F$) as follows:

$$F_{SP} = P_{Audit/Fraud} \cdot ((p-\gamma)\psi) + P_{No Audit/Fraud} \cdot (p \cdot \psi) \quad (10)$$

The Insurance Provider (IP)’s payoff was derived as follows:

$$F_{IP/Alarm} = -(1-p_1) \cdot (p) \cdot P_{Ni-Fraud/Alarm} - p_1 \cdot (p - \gamma) \cdot P_{Ni-Fraud/Alarm} - (1-p_1) \cdot (p) \cdot P_{i-Fraud/Alarm} -$$

$$p_1 \cdot (p - \gamma - \Phi_1) \cdot P_{i-Fraud/Alarm} - p_1 \cdot (-\Phi_2) \cdot P_{i-Ni-Fraud/Alarm} - p_1 \cdot c \quad (11)$$

$$F_{IP/No Alarm} = -(1-p_2) \cdot (p) \cdot P_{Ni-Fraud/NoAlarm} - p_2 \cdot (p - \gamma) \cdot P_{Ni-Fraud/NoAlarm} - (1-p_2) \cdot (p) \cdot P_{i-Fraud/NoAlarm} -$$

$$p_2 \cdot (p - \gamma - \Phi_1) \cdot P_{i-Fraud/NoAlarm} - p_2 \cdot (-\Phi_2) \cdot P_{i-Ni-Fraud/NoAlarm} - p_2 \cdot c \quad (12)$$

Where $p_1 = P_{(Audit/Alarm)}$ and $p_2 = P_{(Audit/No Alarm)}$

The mixed strategy Nash equilibrium derived is represented by the following:

$$\psi_1^* = \left( \frac{P_{Ni}^{NF} \cdot (1 - \epsilon) + P_{Fi}^{Ni} \cdot c - P_{Fi}^{NF} \cdot \epsilon \cdot \Phi_2}{P_{Fi}^{NF} (1 - \epsilon) \cdot (Y - c) + P_{Fi}^{NF} \cdot Y \cdot \Phi_1 - c} \right)$$

$$p_1^* = \left( \frac{p}{(Y \cdot P_{Fi}^{NF} + (1 - \epsilon) \cdot P_{Fi}^{NF})} \right) \quad \text{given} \quad p_2 = 0 \quad \text{if} \quad p < (Y \cdot (\epsilon \cdot P_{Fi}^{NF} + (1 - \epsilon) \cdot P_{Fi}^{Ni}))$$

$$\psi_2^* = \left( \frac{((c - \epsilon \cdot \Phi_2) - P_{Fi}^{NF} \cdot c \cdot (1 - \epsilon) - P_{Fi}^{NF} \cdot \epsilon \cdot (c - \Phi_2))}{(c + \epsilon \cdot (\Phi_1 - \Phi_2) + P_{Fi}^{NF} (1 - \epsilon) \cdot (c - \gamma) - P_{Fi}^{NF} \cdot c \cdot (1 - \epsilon) + P_{Fi}^{NF} \cdot \epsilon \cdot (c - \gamma - \Phi_1) - P_{Fi}^{NF} \cdot \epsilon \cdot (c - \Phi_2))} \right)$$

$$p_2^* = \left( \frac{(\epsilon \cdot P_{Fi}^{NF} + (1 - \epsilon) \cdot P_{Fi}^{Ni}) \lambda - \rho}{(\epsilon \cdot P_{Fi}^{NF} + (1 - \epsilon) \cdot P_{Fi}^{Ni}) \lambda - \lambda} \right) \quad \text{given} \quad p_1 = 1 \quad \text{if} \quad p > (Y \cdot (\epsilon \cdot P_{Fi}^{NF} + (1 - \epsilon) \cdot P_{Fi}^{Ni}))$$

We provide all probabilities and calculations in the Appendix.
In both scenarios, corresponding to the two strategies above, the insurance firm used a combination of detection and deterrence algorithms. The detection algorithm first selected a pool of presumed non-genuine providers. Afterwards, the deterrence algorithm targeted influential providers for audit.

The first scenario described the case where the insurance provider elected to only audit non-genuine practitioners which generated an alarm through the deterrence algorithm. This strategy was only applicable if the amount of fraud was less than the penalty imposed on audited fraudulent practitioners.

\[ (\rho < \gamma \cdot (\varepsilon \cdot P_{FI}^{I} + (1 - \varepsilon) \cdot P_{FI}^{NI}) \]

In case the fraud amount exceeded the penalty imposed upon fraud, the insurance provider had to use the alternative strategy.

The second strategy consisted of the insurance firm auditing all the service providers selected by the deterrence algorithm \( (p_1 = 1) \), in addition to auditing practitioners who were not targeted by the deterrence algorithm with a probability \( p_2 \).

In the alarm case, the optimal probability to defraud \( (\psi_1^*) \) was null when the insurance provider’s expected cost of auditing non fraudulent practitioners equaled the expected deterrence benefit from auditing non fraudulent influential practitioners \( (p_{NF}^{NI}(1 - \varepsilon) + P_{NF}^{I} \cdot \varepsilon)c = P_{NF}^{I} \cdot \varepsilon \cdot \Phi_2 \).

The optimal fraud probability was at a maximum when the insurance provider’s expected payoff from auditing fraudulent practitioners was null, i.e. \( (p_{FI}^{NI}(1 - \varepsilon)(\gamma - c) + P_{FI}^{I} \cdot \varepsilon(\gamma + \Phi_1 - c)) = 0 \)

That means that 1) the cost of audit \( (c) \) equals the penalty collected upon auditing fraudulent providers \( (\gamma) \) and 2) the deterrence benefit from auditing influential practitioners \( (\Phi_1) \) is null \( (\gamma = c \text{ and } \Phi_1 = 0) \).

As intuitively expected, the optimal probability to audit was a function of the loss incurred by the insurance providers in case of unaudited fraud cases \( (\rho) \), the penalty imposed on fraudulent providers \( (\gamma) \), and the algorithm’s positive rate \( (\varepsilon \cdot P_{FI}^{I} + (1 - \varepsilon) \cdot P_{FI}^{NI}) \). The optimal probability to audit was expected to increase as the loss incurred from fraud increased.
Similarly, the more efficient the algorithm was at detecting fraud, and the higher the penalties imposed, the lower was the optimal probability to audit.

**Special Case Scenarios – Analytical Results**

Given a combination of both the detection and deterrence algorithms, the insurance provider needs to select a strategy for auditing practitioners. To avoid auditing genuine providers we first used the detection algorithm to filter out genuine practitioners. We then manipulated the deterrence algorithm to various profiles. It is to be noted that we set the algorithm to generate an alarm targeting different segments of practitioners. This did not necessarily meant auditing all the targeted providers. We examined the insurance provider’s payoff values at equilibrium as derived through solving the game above.

One of the insurance provider’s alternatives was to not take the sentinel effect into consideration, and target the entire pool of non-genuine providers selected by the detection algorithm. This scenario could be achieved by setting the deterrence algorithm to target all (non-genuine) providers. We therefore set both the true positive and false positive rates to 1($P_{F}^{I} = P_{F}^{NI} = P_{NF}^{I} = P_{NF}^{NI} = 1$).

With the deterrence idea in mind, the insurance provider could elect to target all influential non-genuine providers. After filtering out the genuine practitioners, the insurance provider could use the deterrence algorithm to target influential practitioners. This setting aimed at diffusing the audit information and deterrence of neighboring providers. In which case $P_{F}^{I} = P_{NF}^{I} = 1$ and $P_{F}^{NI} = P_{NF}^{NI} = 0$.

Given that a considerable amount of fraud occur though Home Health Care providers who were not connected to the rest of the service provider community (US Department of Justice 2013), the insurance provider could elect to target specifically non-influential providers. In our game set up, we set the deterrence algorithm to generate an alarm for all non-influential non-genuine providers. This scenario is represented by the following: $P_{F}^{NI} = P_{NF}^{NI} = 1$ and $P_{F}^{I} = P_{NF}^{I} = 0$.

For each scenario, we first computed the optimal probability to audit in both the alarm and no-alarm cases (Table 5).
In the first scenario, an alarm was generated by the deterrence algorithm for all types of practitioners (influential/ non-influential, fraudulent/ non-fraudulent).

Therefore, no practitioner fell under the “No alarm” category. As illustrated in Table 5, the optimal defraud probability in this case increased as the cost of audit exceeded the deterrence benefit from auditing non-fraudulent influential practitioners \((c > \varepsilon.\Phi_2)\). The same probability decreased as 1) the penalty imposed on fraudulent practitioners increased, and 2) the deterrence benefit from auditing influential fraudulent practitioners exceeded the deterrence benefit from auditing influential non-fraudulent practitioners \((\varepsilon.\Phi_1 > \varepsilon.\Phi_2)\). It is important to note that in this scenario, the ratio of influential practitioners in the network affected the defraud probabilities.

In the second scenario, the deterrence algorithm generated an alarm for all influential (non-genuine) practitioners \((\varepsilon = 1)\). Practitioners targeted by the alarm were therefore expected to defraud as long as 1) the cost of audit was larger than the deterrence benefit from auditing non-fraudulent practitioners in the pool and 2) the deterrence benefit from auditing fraudulent practitioners exceeded the deterrence benefit from auditing non-fraudulent practitioners.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Algorithm’s Profile</th>
<th>Optimal Defraud Probability</th>
<th>Optimal Defraud Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Generate alarm for all non-genuine providers</td>
<td>(c - \varepsilon.\Phi_2) (\gamma + \varepsilon(\Phi_1 - \Phi_2))</td>
<td>Not Applicable (An alarm is generated for all non-genuine providers → There is no provider that falls in the No Alarm Case)</td>
</tr>
<tr>
<td>2</td>
<td>Generate alarm for all Non-genuine Influential providers</td>
<td>(c - \Phi_2) (\gamma + \Phi_1 - \Phi_2)</td>
<td>(c) (\gamma)</td>
</tr>
<tr>
<td>3</td>
<td>Generate alarm for all Non-genuine non-influential providers</td>
<td>(c) (\gamma)</td>
<td>(\frac{c - \Phi_2}{\gamma + \Phi_1 - \Phi_2})</td>
</tr>
</tbody>
</table>
The last scenario targets all non-influential practitioners for alarm. Thus, practitioners in the alarm pool did not take into consideration the deterrence benefit, and defraud with a probability \( \frac{c}{\gamma} \).

The defraud probability in this case increased with the rise of audit cost \( (c) \), and decreased with the rise of the penalty imposed upon detecting fraud \( (\gamma) \).

Using the optimal defraud probability set above, we calculated the expected insurance provider payoff in each of the three scenarios.

\[
F = P_{\text{Alarm}} \cdot F_{\text{IP/Alarm}} + P_{\text{NoAlarm}} \cdot F_{\text{IP/No Alarm}}
\]

We summarize our findings in table 6 below.

<table>
<thead>
<tr>
<th>Algorithm’s Profile</th>
<th>Strategy 1 ((p_1 = p_1^<em>, p_2 = 0 \text{ and } \psi = \psi^</em>))</th>
<th>Strategy 2 ((p_1 = 1, p_2 = p_2^* \text{ and } \psi = \psi_2^*))</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scenario 1</td>
<td>Generate alarm for all non-genuine providers (PFI = PFNI = 1 &amp; PNFI = PNFNI = 1)</td>
<td>Not Applicable (\text{An alarm is generated for all non-genuine providers} \Rightarrow \text{There is no provider that falls in the No Alarm Case})</td>
</tr>
<tr>
<td></td>
<td>(F = F_{\text{Alarm}} ) (= \frac{\rho \cdot (e \cdot \Phi_2 - c)}{(\gamma + e \cdot (\Phi_1 - \Phi_2))})</td>
<td></td>
</tr>
<tr>
<td>Scenario 2</td>
<td>Generate alarm for all Non-genuine Influential providers (PFI = PNFI = 1 &amp; PNFI = PNFNI = 0)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(F = F_{\text{Alarm}} = \frac{\rho \cdot (\Phi_2 - c)}{(\gamma + \Phi_1 - \Phi_2)})</td>
<td>(c \cdot e \cdot (\Phi_1 - \Phi_2) + e \cdot \gamma \cdot \Phi_2 + e \cdot \rho \cdot (\Phi_2 - c) - \rho \cdot c) (= \frac{c \cdot e \cdot \rho \cdot (\Phi_1 - \Phi_2) + e \cdot \gamma \cdot \Phi_2 + e \cdot \rho \cdot (\Phi_2 - c) - \rho \cdot c}{(\rho + \gamma)})</td>
</tr>
<tr>
<td>Scenario 3</td>
<td>Generate alarm for all Non-genuine non-influential providers (PFNI = PNFNI = 1 &amp; PFI = PNFI = 0)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(F = F_{\text{Alarm}} = F_{\text{NoAlarm}} ) (= \frac{-\rho \cdot c}{(\gamma)})</td>
<td>(\rho \cdot (\Phi_2 - c) - (1 - e) \cdot (c \cdot \rho + \gamma \cdot \Phi_2 + c \cdot (\Phi_1 - \Phi_2))) (= \frac{\rho \cdot (\Phi_2 - c) - (1 - e) \cdot (c \cdot \rho + \gamma \cdot \Phi_2 + c \cdot (\Phi_1 - \Phi_2))}{(\gamma + \rho + \Phi_1 - \Phi_2)})</td>
</tr>
</tbody>
</table>

To better understand the effect of the different variables on the expected payoffs, we perform a sensitivity analysis.
Special Case Scenarios – Sensitivity Analysis

We considered a network similar to the one used in previous sections composed of 1000 providers for whom the total amount of claims submitted had a mean of $500,000 and a standard deviation of $100,000.

Within the network, 10% of providers were set to be highly connected (having more than 10 immediate neighboring providers). Therefore, the fraction of influential practitioners in the network (ε) was set to 0.1.

According to the CMS (Center of Medicare and Medicaid Services), the rate of improper billing for the year 2012 (CERT) was set to 8.6%. We used the CERT rate in the simulation as rate of waste and abuse in the network.

We calculated the benefit from auditing an influential fraudulent practitioner ($\Phi_1$), as well as the benefit from auditing an influential non-fraudulent provider ($\Phi_2$).

By varying the expected amount of fraud ($\rho$) and cost of audit ($c$), we looked at the expected insurance provider payoff in two different worlds namely High diffusion/Low Decay, and Low diffusion/High Decay.

Scenario 1

Simulation results indicated that the deterrence benefit from auditing a non-fraudulent practitioner was set to about $120K in the low diffusion high decay case ($\Phi_2 = 120$), and $500k in the high diffusion low decay case ($\Phi_2 = 500$). The deterrence amount from auditing a fraudulent influential practitioner averaged to about $210K in the low diffusion high decay case ($\Phi_1 = 210$), and $4300k in the high diffusion low decay case ($\Phi_1 = 4300$). Looking at the insurance provider payoff in this scenario (Figure 10), we note that no matter what the fraud amount was, the payoff from auditing all non-genuine providers was negative as the audit cost exceeded the benefits from deterrence ($c > \varepsilon . \Phi_2$).
In the high diffusion low decay world, the insurance provider payoff was much larger than the payoff expected in the low diffusion high decay world.

Also, the insurance provider could still expect positive payoff with relatively higher audit cost. This is due to the large deterrence benefit resulting from auditing fraudulent influential providers.

In practice, the insurance provider can use these metrics, along with information about intensity of diffusion in their network, and number of influential practitioners to calculate the expected payoff amount from auditing all non-genuine practitioners.

![Figure 10: Insurance Provider Payoff by Audit Cost and Fraud Amount – Scenario 1](image)

**Strategy 1:** All Non-Genuine Alarm (Low Diffusion/High Decay)

**Strategy 2:** All Non-Genuine Alarm (High Diffusion/Low Decay)

**Scenario 2**

In this specific scenario, the deterrence algorithm was intended to generate an alarm for every influential non-genuine practitioner. The insurance provider in this case had to elect to either utilize audit strategy 1 or audit strategy 2 as derived in the game equilibrium above.

The first strategy consisted of 1) setting the fraud penalty to be relatively high ($\gamma > \frac{\rho}{\epsilon}$) and only audit practitioners who generated an alarm (influential non-genuine in this case) with an optimal probability less than 1 ($p_1 = \frac{\rho}{\gamma \epsilon}$).

Results indicated that the expected insurance provider payoff resulting from auditing influential practitioners was much larger than the payoff resulting from auditing all non-genuine practitioners (Figures 10 and 11).
As expected, the payoff was much larger in the “High Diffusion Low decay World”, where audit information spread across the network, and deterrence amounts were at a maximum (Figures 10 B, D). Another alternative provided for the insurance firm was to set the penalty relatively low ($\gamma < \frac{\rho}{\epsilon}$), audit the practitioners who generated an alarm (influential non-genuine in this case) with a probability of 1, and audit the practitioners who did not generate an alarm (influential non-genuine in this case) with a probability ($p_2 = \frac{\epsilon \gamma - \rho}{1 - (1 - \epsilon) \gamma}$). Results show that by selecting this strategy, the insurance provider incurred the most benefit when diffusion was high and magnitude of fraud was low. Since all the influential practitioners in the network were audited, the audit information reached a larger proportion of practitioners in the network, and generated a much higher deterrence benefit.

Figure 11. Insurance Provider Payoff by Audit Cost and Fraud Amount – Scenario 2
(A) Strategy 1 Influential Non-Genuine Alarm/False Alarm (Low Diffusion/ High Decay)
(B) Strategy 1 Influential Non-Genuine Alarm/False Alarm (High Diffusion/ Low Decay)
(C) Strategy 2 Influential Non-Genuine Alarm/False Alarm (Low Diffusion/ High Decay)
(D) Strategy 2 Influential Non-Genuine Alarm/False Alarm (High Diffusion/ Low Decay)
Scenario 3

By targeting non-influential practitioners for audit, the insurance provider did not incur the deterrence benefit from auditing influential non-genuine practitioners. As the cost of audit increased, the expected payoff increased. It is to be noted that, as per our equilibrium criteria: \( \rho < \gamma \cdot (\varepsilon \cdot P_F^I + (1 - \varepsilon) \cdot P_F^{NI}) \), that the penalty imposed on fraudulent practitioners was correlated with the amount of fraud:

\[
\frac{\rho}{\gamma} = k \quad \text{such as} \quad k < 1 - \varepsilon
\]

Therefore, the overall payoff did not vary with the amount of fraud as illustrated in Figure 12 A, B. The second option for the insurance provider was to audit all non-influential non-genuine practitioners with a probability 1, and influential non-genuine practitioners with probability \( p_2 = \frac{(1 - \varepsilon)\lambda - \rho}{(1 - \varepsilon)\lambda - \rho} \) (Strategy 2). In this case, as expected, the insurance provider’s loss was much larger because more practitioners were audited and no deterrence benefit was incurred.

**Figure 12. Insurance Provider Payoff by Audit Cost and Fraud Amount – Scenario 3**

(A) Strategy 1 Non-Influential Non-Genuine Alarm/False Alarm (Low Diffusion/High Decay)
(B) Strategy 1 Non-Influential Non-Genuine Alarm/False Alarm (High Diffusion/Low Decay)
(C) Strategy 2 Non-Influential Non-Genuine Alarm/False Alarm (Low Diffusion/High Decay)
(D) Strategy 2 Non-Influential Non-Genuine Alarm/False Alarm (High Diffusion/Low Decay)
Looking at the expected payoff for the insurance providers in the above algorithm’s profiles, it was obvious that it was more beneficial for the insurance provider to target influential non-genuine practitioners for audit.

In summary, in this section we formulated and analyzed an audit game, as well as studied special cases. In addition to closed form results the sensitivity analyses permitted us to evaluate the magnitude of the benefits from the different strategies.

In addition to providing strategies for audit these results and sensitivity analyses can even be used by insurance providers to set important factors such as penalties for fraud, waste and abuse. The analytical results reinforce some conclusions from the agent-based simulation studies, such as the value of auditing highly influential/connected providers in certain cases.

These are also complementary and provide new findings since they do take a different and important perspective. In the game, the setting is one where both parties are making strategic decisions given the information available. In the agent-based simulation, the agents are assumed to be fraudulent or not and determine their billings, and the algorithm then works to determine providers for audit. Both perspectives are useful, and in this case, point to the value of taking the sentinel effects into account in audit algorithms.

**Conclusions**

Healthcare costs have risen immensely in the past decades. In addition to fraudulent activity, the healthcare system is known to encompass a great amount of waste and abuse. Mostly, existing auditing algorithms aim at detecting fraudulent providers and hence take a narrow perspective of this issue. Our paper here presented a fundamentally new approach that showed value in deterrence-based auditing algorithms in applications like healthcare and warranty. To our knowledge, ours is the first study that took into account, the sentinel effect while designing auditing algorithms, an important research contribution. We also showed how incorporating the audit effect into these algorithms addressed the modeling of waste and abuse reductions commonly seen after audit.
In addition to the finding on the value of deterrence-oriented audit, we showed that this effect should not necessarily be taken for granted in all applications. Network structure and the diffusion mechanisms in place significantly impacted the effect of using such algorithms. Certain network topologies, such as power law networks did lend themselves to deterrence-oriented audit. Likewise with certain diffusion mechanisms, such as a high propensity to propagate to immediate neighbors with low decay. Real-world applications such as healthcare and warrant fraud did likely have their own specific forms of networks and diffusion, and these needed to be considered before utilizing such algorithms.

Our work offers significant theoretical as well as practical contributions. This paper presents the first deterrence-based audit algorithm under network effects, a significant contribution. Further through collaboration with industry in healthcare and auto warranty, we have been able to design and study realistic agent based simulations, augmented by real data where possible. Our analytical results and the study of special cases provide important theoretical insights into this challenging domain.

An important issue to incorporate in future work is the policy aspect of deterrence oriented audit. Is this a fair approach for audit? We defer this consideration to a more exhaustive treatment needed here. One approach may be to use deterrence as a second filter after a detection algorithm is employed, as done in the game theoretical model studied here. In such cases, the algorithm first flags fraudulent providers. Conditional on having been flagged, deterrence is used to then select providers for audit. However there are clearly many other ways in which this can be addressed and it may depend on the domain and the legal frameworks that apply. A for-profit private insurer may approach trade-offs differently than, say, Medicare or the IRS. These are beyond the scope of a single study but are fascinating questions for policy that we hope to examine in ongoing work.

References


Medicine, I. o. (2012).


CHAPTER 3: MEDICAL RECOMMENDATIONS IN REDUCING HEALTHCARE COSTS

Introduction

Healthcare costs have reached skyrocketing numbers in the US. With spending set at almost 18% of the GDP, headed for 20% by 2020 (Berwick & Hackbarth, 2012), the US is the leading country in the world for healthcare spending. Medical fraud represents a large portion of this spending (Kalb, 1999). In addition, overuse of medical procedures, otherwise referred to as waste and abuse, also contributes to this high level of spending.

Particular to healthcare, prices for the same procedures vary tremendously depending on the paying party (Beck, 2014). Insurance companies negotiate prices for each procedure based on the health plans provided (Thorpe, 1997). These price variations are very significant. In an example reported by the Wall Street Journal (Beck, 2014), the average charge for a joint-replacement surgery ranged from $5,300 in Ada, Oklahoma, to $223,000 in Monterey Park, California. Price variations have also been reported within a single city, where the cost of treating a case of heart failure varies from $9,000 in one hospital, to $51,000 in another, in Jackson, Mississippi (Beck, 2014).

It is important to note that providers are typically unaware of healthcare costs. Physicians are able to identify generic drugs within each drug type. However, they remain in the dark about exact test and drug costs (Beck, 2014).

A strategy for reducing healthcare costs would therefore be to make cost information visible to medical providers at the right time. This could be accomplished through a medical recommender system that presents alternative prescription options that are: 1) appropriate for the patient being consulted, and 2) are lower in cost compared with the physician’s initial selection.
While incorporating such recommendations in the medical decision making process can result in cost-aware providers, and therefore help reduce costs, the use of recommender systems could suffer from lower acceptance rates because of the time pressure experienced during consultation. We note that in this domain, because of the time involved in evaluating different recommendations (especially in emergency rooms), recommender systems have been either selectively used or completely removed (Drescher et al., 2011). Therefore, understanding the role of time pressure is important in the adoption of recommender systems in the healthcare domain.

Medical recommender systems could be useful for medical diagnosis, given symptoms, as well as for suggesting treatment options, given diagnosis. While we can study the medical recommendation process in both scenarios, this research’s scope is limited to recommending appropriate procedures given a diagnosis.

In this paper, we first present different factors expected to impact the use of recommender systems in the medical domain. We then present and evaluate two different influence dynamics models using a combination of different factors. The first model focuses on the impact of cost variance and time pressure on the physicians’ influence by the system recommendations. The second, more comprehensive dynamics model, is built using an informal focus group of four physicians at one of the leading US healthcare facilities. The model integrates the impact of 1) time pressure, 2) cost variances, 3) outcome, 4) risk, as well as 4) influence predisposition on the use propensity of such recommendations. Our “costs and time pressure effects model” is evaluated using data from a field experiment with medical providers. Our “comprehensive model” is then evaluated using an agent-based simulation.

**Systems Recommendations Influence Factors**

Designing recommender systems requires an understanding of the underlying influence dynamics, which models use (or lack thereof) of the recommender under different conditions.

In this section, we present the factors assumed to influence recommendation use. We then present two different influence dynamics models using the stated factors.
The five factors used in the influence model are based on the presence of time pressure, risk level, procedure outcome and cost, provider type, and the predisposition to be influenced.

**Time-Pressure:** In the medical domain, delay occurs when patients arrive late, when consultation time lasts longer than expected, and when physician emergencies arise. As delay builds up during the day, the provider experiences time pressure.

However, depending on various factors such as personality and experience, our model assumes that people react to delay build-up differently. Specifically, providers are assumed to differ in the amount of build-up needed to occur before they start experiencing time pressure. We refer to that amount as time pressure retardancy $\alpha$.

Once the medical doctor exceeds his/her “time pressure retardancy” threshold, time pressure starts to increase gradually with a “delay-pressure coefficient” $\beta$. Because delay needs to accumulate before creating build-up and thus generating time pressure, the relationship between delay build-up and time pressure is not linear. We model the time pressure (TP) as a sigmoid function of delay build-up ($x$), as follows:

$$TP = \frac{1}{1 + e^{(-x/\beta + \alpha)}} \text{ where } \alpha > 0 \text{ and } \beta > 0$$

(15)

When delay build-up is very low (below the time pressure retardancy threshold), the provider is viewed as under low-pressure. Otherwise, the provider is under high time pressure in this model.

**Risk:** Even though determining diagnosis related procedures is a relatively low risk task (when compared to identifying a diagnosis from set of symptoms), the level of risk involved differs by provider specialty. Emergency department physicians for example deal with higher risk cases, whereas primary care providers usually handle more routine low risk cases. In addition, specialties where new diagnoses are set, or new drugs are prevalent could also be categorized as being of a relatively higher risk.

**Cost Relative Difference (CD):** We differentiate between three different provider views when it comes to the relative cost difference between the procedure recommended by the system and the cost difference of the procedure initially selected by the providers.
We therefore define three types of providers namely 1) The Price-Indifferent Provider (PI), 2) The Price-Quality Provider (PQ), and 3) The Price-Sensitive Provider (PS).

The Price-Indifferent Provider (PI): In the US medical system, because most procedures are covered by external payers, many doctors typically ignore cost (until recently, when the new healthcare law was enacted).

We refer to this type of practitioners as “Price-Indifferent, for whom the probability of influence by the recommender agent does not depend on the relative cost difference between the recommended procedure and the pre-selected one.

The Price-Quality Provider (PQ): As with several consumers, some medical practitioners and patients may perceive price as a cue to quality (Teltis, 1990). For the “Price-Quality Provider”, a procedure might be viewed as more effective because it is more expensive. For this type of practitioner, the influence probability is positively correlated with cost difference.

The Price-Sensitive Provider (PS): By providing cost information some providers might be susceptible to altering their prescribing behavior in favor of the less expensive procedure. With the new regulations in place in the US, medical practitioners are indeed held accountable for any unnecessary costs imposed on the system. For this type of practitioners, the influence probability would be inversely correlated with the relative cost difference.

Outcome: Physicians ‘decision quality is difficult to measure in healthcare because of 1) lack of reliable outcome measures, and 2) the variance in outcome between different patients. With the introduction of EMR, individual patient outcomes can be tracked. However, such measures are still under development. Instead, as we do in this paper, aggregate levels of success of each procedure could be used as a measure of outcome (e.g. “63% of all patients who use drug X see a reduction in triglycerides”). We assume these are “given” for each procedure type.

Influence Predisposition (IP): In this study, we also differentiate between two types of providers when it comes to the likelihood of being influenced by the recommendation provided by the system.
The “Low-Swap User”, represents experienced physicians, and may be less likely to be influenced by recommendations. Being domain experts, physicians do not necessarily recognize the need of using decision support systems or recommenders during consultation (Berner, 2008).

The “High-Swap User”, on the other hand, describes providers who are more likely to alter their chosen procedure in favor of another recommended one, given the evidence in support. Perhaps the typical medical student and/or resident doctor may fall into such a category.

Table 7 below lists all the variables used in describing the systems recommendations influence factors.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Definition</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>α</td>
<td>Time Pressure Retardancy</td>
<td>Threshold after which delay build-up triggers Time Pressure</td>
</tr>
<tr>
<td>β</td>
<td>Delay-Pressure Coefficient</td>
<td>Rate of change of time pressure as delay build-up increases (Provider attribute).</td>
</tr>
<tr>
<td>TP</td>
<td>Time Pressure</td>
<td>Level of time pressure caused by delay build-up</td>
</tr>
<tr>
<td>OD</td>
<td>Outcome Difference</td>
<td>Outcome difference between recommended &amp; pre-selected procedure.</td>
</tr>
<tr>
<td>CD</td>
<td>Cost Difference</td>
<td>Cost difference between recommended and pre-selected procedure.</td>
</tr>
<tr>
<td>IP</td>
<td>Influence Predisposition</td>
<td>Predisposition to swap from the pre-selected procedure.</td>
</tr>
</tbody>
</table>

**Influence Dynamics - Cost and Time Pressure Effects Model**

Given the factors noted above, here we present a simple normative model for medical recommender use under various cost and time pressure conditions.

*The Cost Effect*

Because of the lack of prior literature related to the impact of cost information on physician decision making, it is difficult to predict the physicians’ influence by low cost recommendations. In this domain, there is no direct link between the patient’s cost of the prescribed procedure and the physicians’ utility. Prior literature lacks research that investigates the impact of cost in such scenarios.
In e-commerce, recommender systems have been successful at helping users find items meeting their exact specific needs (Schafer et al. 1999). Knowledge-based recommender systems, for example, take into account user preferences, such as price, to provide lists of recommendations (Trewin, 2000).

However, consumers have been shown to react to price differently. Some consumers thrive to optimize value, which is defined as the lowest price for a set quality level. Others view price an indicator of quality. That perception is more or less relevant depending on the availability of quality cues, price variation within the same class of products, level of consumer price awareness, and consumers ability to detect quality variation within the same group of products (Zeithaml, 1988).

Therefore, we consider two different recommender settings. In the first setting, referred to as the low-cost setting, physicians would be presented a list of low cost alternatives. In the other recommender setting, the list of recommendations presented would include a mix of high and low cost alternatives. This last setting we refer to as the mixed-costs recommender. We therefore posit that while physicians would likely to be interested in viewing recommendations and adjusting to the lower treatment options (given that the resulting outcome is similar), it is anticipated that their behavioral change would also be affected by the cost of the recommendations presented. Hence, we make the following hypotheses:

\[ H1-a. \text{ Viewing of recommendations is lower in the mixed-costs recommender settings than in the low-costs recommender setting.} \]

\[ H1-b. \text{ Adopting recommender treatment options is lower in the mixed-costs recommender settings than in the low-costs recommender setting.} \]

Under the assumption that physicians are interested in minimizing the patient’s share of cost whenever possible, we anticipate physicians to prescribe the lower cost procedure whenever one is available. However, that behavioral change might be affected by other factors such as time pressure.
**The Time Pressure Effect**

Time pressure is common in the medical practice. In a study of managed care physicians, medical doctors indicated feeling time pressure, and “acknowledged needing up to 41% more time than allotted to provide quality care during new patient visits” (Linzer et al., 2000).

Under high time pressure, prior literature indicates that the load of information processed while making decisions is different than when under low time pressure. In 1992, Hahn and colleagues reported that when subjects in their study were not “hurried” while making decisions, their decision quality steadily increased as more information was presented (Hahn, Lawson, & Lee, 1992).

As applied to the medical domain, when enough consultation time is provided, physicians would benefit from processing more information. Under low time pressure, physicians are therefore anticipated to view and process additional information provided by the recommender system. They would also be more inclined to “optimize” their treatment options to the best possible outcome/cost combination when time permits. On the other hand, physicians under high time pressure would be inclined to process less information (Wright, 1974), and thereby ignore systems recommendations. Therefore, we posit the following hypotheses:

- **H2-a.** Viewing of recommendations is lower when physicians are under high time pressure than when under low time pressure.
- **H2-b.** Adopting recommender treatment options is lower when physicians are under high time pressure than when under low time pressure.

Table 8 below lists the hypotheses developed in the “Cost and Time Pressure Effects Model”.

<table>
<thead>
<tr>
<th>Hypothesis</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>H1</strong></td>
<td></td>
</tr>
<tr>
<td>a.</td>
<td>Viewing of recommendations is lower in the mixed-costs recommender settings than in the low-costs recommender setting.</td>
</tr>
<tr>
<td>b.</td>
<td>Adjusting of treatment options is lower in the mixed-costs recommender settings than in the low-costs recommender setting.</td>
</tr>
<tr>
<td><strong>H2</strong></td>
<td></td>
</tr>
<tr>
<td>a.</td>
<td>Viewing of recommendations is lower when physicians are under high time pressure than when under low time pressure.</td>
</tr>
<tr>
<td>b.</td>
<td>Adjusting of treatment options is lower when physicians are under high time pressure than when under low time pressure.</td>
</tr>
</tbody>
</table>
Cost and Time Pressure Effects Mode Evaluation: Experimental Analysis

An experiment was designed to evaluate out “cost and time pressure effects model”. Therefore, we specifically looked at the effects of recommendation cost variance and time pressure on the physicians’ probability of viewing and influence by medical recommendations. The experiment was conducted online. The medical provider (participant) was presented a set of medical cases (scenarios), a description of the accompanying context, and a list of drugs to select from (for prescription purposes). After the medical provider made an initial selection of drugs to be prescribed, related cost information was displayed. The procedure cost presented provided the expected cost to the patient. The medical provider then had the option to view systems recommendations. System recommendations included drugs similar to the one initially selected by the provider along with cost information. Note that all recommendations presented were reviewed by two medical experts (please refer to Appendix B for the list of medical cases created for the experiment). Providers were randomly placed into different groups. Depending on their group assignment, providers were presented with either 1) all less expensive, or 2) mixed costs recommendations. The experiment was conducted using Qualtrics; which allowed the dynamic allocation of procedures and recommendations costs.

The initial cost for each prescription selected was stored in temporary variables. For the low-cost groups, recommendation costs were adjusted dynamically to be lower than the cost of the procedure initially selected by the participant.

For the mixed-costs groups, the initial treatment option (s) selected by the participant was dynamically set to a cost X. At least two recommendations were then presented with costs Y and Z; where Y<X<Z.

The provider then had the option to alter his/her initial selection. Each provider's choice to 1) view the system recommendations, and 2) adjust the treatment option were recorded. In order to simulate high time pressure, in half of the cases, we displayed a message indicating lack of remaining time (THE SYSTEM HAS JUST ADDED SEVERAL CASES TO YOUR QUEUE! Please try to complete all cases within the allocated time.), along with a timer counting the number of seconds spent on each page.
The order in which the time pressure treatment was presented varied in sequence, as means of counterbalancing.

Follow-up survey questions were also presented to capture the providers' designation, specialty, and years of experience.

**Experiment Design**

Three different treatments were identified for this experiment: Time Pressure, Recommendations Costs, and Time Pressure Display Order.

Subjects in the experiment were presented six different fictional patient cases and asked to select the most appropriate prescription regimen (Appendix B).

A within-subjects design was used to evaluate the impact of time pressure on the physicians' use of the system recommendations. In half of the cases, the participants were under high time pressure (Table 9).

<table>
<thead>
<tr>
<th>Participant</th>
<th>Time Pressure</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>3 Cases</td>
<td>3 Cases</td>
</tr>
</tbody>
</table>

*Note: Each subject within the groups was presented Cases 1 – 6*

A between-subjects design was used to investigate the effect of the time pressure display order on the use of the recommendations. Depending on the group of the participant, the high-time pressure was either presented first (with cases 1-3) or last (cases 4-6)

A between-subjects design was also used to test the effect of cost on the use of the recommendations. Depending on their group, participants were presented with varying recommendation costs. Half of participants were presented a list of all low cost recommendations, while the other groups received recommendations of mixed costs. Participants were randomly placed into four different groups (Table 10).
Table 10: Experiment Treatments/Groups – Subjects

<table>
<thead>
<tr>
<th>Recommendations Costs</th>
<th>Time Constraint Display Order</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>High TP First</td>
<td>High TP Last</td>
</tr>
<tr>
<td>All Less Expensive</td>
<td>Group 1</td>
<td>Group 2</td>
</tr>
<tr>
<td>Mixed Costs</td>
<td>Group 3</td>
<td>Group 4</td>
</tr>
</tbody>
</table>

Note: Participants were randomly placed in each of the treatment groups.

It is important to note that the cases used in the experiment were created in collaboration with two medical providers. Even though the cases are different in nature, they have been assessed by our experts as being similar in terms of complexity and risk level.

Instrument Description

Experiment Results

Traditional quantitative statistical methods were used to test the relative difference between the groups. Results provided insights on how the systems’ recommendations were used in different scenarios. An inductive analysis also confirmed our statistical results.

Subjects

A total of 40 medical providers participated in the experiment. By virtue of the generic nature of the cases used (related to primary care practice), clinicians of all specialties were able to complete them. Our pool of participants was mainly composed of medical doctors, most with considerable number of years of experience (Figure 12).

We were able to collect a balanced set of responses in terms of both subjects and cases. Ten complete responses were recorded in each of the four groups.

With each participant completing three cases under low time pressure, and three cases under high time pressure, we had a set of 240 observations (at the case level).
Data Description

Two different dependent variables were measured; namely: 1) “View” which represents the number of occurrences, the participants viewed the system recommendations, and 2) “Adjust” which refers to the number of times the participants adjusted their initial treatment plan to the recommended one.

The tables below provide descriptive statistics showing viewing and adjusting counts by group (Table 9), time pressure level (Table 10), and a combination of group and time pressure (Table 11).

Overall, we see that participants did view system recommendations for 181 out of the 240 cases, and adjust their treatment plans for 117 out of the 240 cases. These large numbers indicate the providers’ general inclination to reducing patient treatment costs.

Recommendations Costs Variance Effect

When comparing the number of recommendations viewed in the low-cost recommendations groups, we see a subtle difference indicating that participants did react differently to the recommendation cost treatment.
That difference is much more significant when it comes to adjusting treatments plans. In the low-cost recommendations groups the number of adjustments reached 73 out of 120, when it only got to 44 out 120 in the mixed-costs recommendations groups.

**Time Pressure Level Effect**

Numbers in Table 11 below suggest that participants tended to view recommendations (95 versus 86), and adjust treatment options (59 versus 58) more frequently when under low time pressure.

<table>
<thead>
<tr>
<th>Table 11: Descriptive Statistics – Time Pressure Level (View Count, Adjust Count)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Time Pressure</strong></td>
</tr>
<tr>
<td><strong>Recommendations Costs</strong></td>
</tr>
<tr>
<td><strong>All Less Expensive</strong></td>
</tr>
<tr>
<td><strong>Mixed Costs</strong></td>
</tr>
<tr>
<td><strong>Total</strong></td>
</tr>
</tbody>
</table>

**Time Pressure Display Order Effect**

Table 12 below shows some differences between viewing and adjusting counts between the “high time pressure first” and “high time pressure last”. With 98/120 versus 83/120 viewed recommendations, and 63/120 versus 54/120 adjusted treatment plans, the order in which time pressure is presented seems to have an effect of our participants’ propensity to change.

<table>
<thead>
<tr>
<th>Table 12: Descriptive Statistics - Group Level (View Count, Adjust Count)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Time Pressure Display Order</strong></td>
</tr>
<tr>
<td><strong>High TP First</strong></td>
</tr>
<tr>
<td><strong>Recommendations Costs</strong></td>
</tr>
<tr>
<td><strong>All Less Expensive</strong></td>
</tr>
<tr>
<td><strong>Mixed Costs</strong></td>
</tr>
<tr>
<td><strong>Total</strong></td>
</tr>
</tbody>
</table>

This could be explained by the fact that when time pressure is experienced early on, it could be perceived as the norm; and might therefore be overlooked.
When, on other hand, providers start off with an ample time to consult patients, and then when imposed a
time constraint, they are more likely to react by ignoring systems recommendations. The time pressure
effect is hence posited to differ depending on the sequence of events.

**Interaction Effect**

Looking at the interaction effects of all of our three different treatments, recommendation costs,
time pressure level, and time pressure display order (Table 13), we see large differences in viewing and
adjusting counts in some distinct cases. When high time pressure is presented last, we see that both viewing
and adjusting counts varied depending on the time pressure level. In that scenario, viewing
recommendations increased from 18/30 under the high time pressure to 27/30 under low time pressure for
the low cost recommendations. That same number increased from 14/30 under the high time pressure to
24/30 under low time pressure for the mixed cost recommendations. Similarly, adjusting treatment option
counts increased from 13/30 under the high time pressure to 21/30 under low time pressure for the low cost
recommendations. That same number increased from 9/30 under the high time pressure to 11/30 under low
time pressure for the mixed cost recommendations.

This could be explained by the anchoring and adjustment bias of judgement. When a time constraint is
imposed early on, it could be perceived as the norm. Participants then might adjust to that initial anchor
(Tversky & Kahneman, 1974) and not experience any time pressure. When, on the other hand, the time
constraint is imposed after a phase of ample processing time, providers would probably feel more
significant time pressure. Therefore, the time pressure level effect might be more substantial when time
constraint appears later in the process.
Table 13: Descriptive Statistics – Group/Time Pressure (View Count, Adjust Count)

<table>
<thead>
<tr>
<th>Recommendations Costs</th>
<th>Time Pressure Display Order</th>
<th>High TP First</th>
<th>Low TP</th>
<th>High TP Last</th>
<th>Low TP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>High TP</td>
<td>Low TP</td>
<td>High TP</td>
<td>Low TP</td>
<td></td>
</tr>
<tr>
<td>All Less Expensive</td>
<td>27, 22</td>
<td>26, 17</td>
<td>18, 13</td>
<td>27, 21</td>
<td></td>
</tr>
<tr>
<td>Mixed Costs</td>
<td>27, 14</td>
<td>18, 10</td>
<td>14, 9</td>
<td>24, 11</td>
<td></td>
</tr>
</tbody>
</table>

Statistical Findings

To investigate the statistical significance of our findings, we have used a two-level logistic regression analysis. The analysis was performed at the patient case level (a total of 240 observations). Since each participant responded to six different cases (repeated measures as per experiment’s within-subject design), we have used generalized estimating equations. Observations related to the same participants were therefore grouped within the same cluster; resulting in 40 different clusters.

Two different models were created for each of the dependent variables: Viewing system recommendations (View=1) and adjusting treatment options (Adjust=1).

\[
Pr(\text{Adjust}=1 \mid \text{Recs}, \text{TPLevel}, \text{HiTPLast}) = \beta_0 + \beta_1 \text{Recs} + \beta_2 \text{TPLevel} + \beta_3 \text{HiTPLast} + \beta_4 \text{TPLevel} \times \text{HiTPLast} \quad (13)
\]

\[
Pr(\text{View}=1 \mid \text{Recs}, \text{TPLevel}, \text{HiTPLast}) = \beta_0 + \beta_1 \text{Recs} + \beta_2 \text{TPLevel} + \beta_3 \text{HiTPLast} + \beta_4 \text{TPLevel} \times \text{HiTPLast} \quad (14)
\]

Where \text{Recs}, \text{TPLevel}, and \text{HiTPLast} are all dichotomous predictors. \text{Recs} is set to 0 for low cost recommendations, and 1 for mixed cost recommendations. \text{TPLevel} is set to 0 for low time pressure and 1 for high time pressure. \text{HiTPLast} is set to 0 when time pressure is presented for the first cases and 1 for cases when time pressure was presented last.

Statistical results for our adjusting treatment options model (Table 14) indicate the significance of most of our model terms. In order to take into account the correlation between the main effect and interaction terms, effect, rather than reference coding was used for this analysis.
The recommendations cost term was significant at (p=0.0198) indicating a significant difference in adjusting treatment options between the cases showing low cost recommendations versus cases with mixed costs recommendations; providing support for our hypothesis H1-b. As expected, the interaction between time pressure level and order of time pressure display was highly significant with a p value of 0.0031.

| Parameter          | Estimate | Standard Error | 95% Confidence Limits | Z  | Pr>|Z| |
|--------------------|----------|----------------|-----------------------|----|------|
| Intercept          | -0.0702  | 0.2131         | -0.4879               | 0.3475 | -0.33 | 0.7419 |
| Recs               | 0.5166   | 0.2218         | 0.0819                | 0.9512 | 2.33  | 0.0198 |
| TPLLevel           | 0.0193   | 0.1133         | -0.2027               | 0.2412 | 0.17  | 0.8649 |
| HiTPLast           | 0.1744   | 0.2163         | -0.2495               | 0.5983 | 0.81  | 0.4201 |
| TPLLevel*HiTPLast  | -0.3430  | 0.1160         | -0.5703               | -0.1157 | -2.96 | 0.0031 |

When it came to viewing system recommendations, differences in recommendation costs were not statistically significant. Hence, providing no support for out hypothesis H1-a. Other terms in the model however, were highly significant. Similar to the adjusting model, the interaction between time pressure and order of time pressure display was also highly significant for the viewing model with a p value of 0.0002.

| Parameter          | Estimate | Standard Error | 95% Confidence Limits | Z  | Pr>|Z| |
|--------------------|----------|----------------|-----------------------|----|------|
| Intercept          | 1.3284   | 0.2840         | 0.7717                | 1.8850 | 4.68  | <0.0001 |
| Recs               | 0.4478   | 0.2910         | -0.1226               | 1.0183 | 1.54  | 0.1239 |
| TPLLevel           | 0.1039   | 0.1911         | -0.2706               | 0.4785 | 0.54  | 0.5864 |
| HiTPLast           | 0.3752   | 0.2851         | -0.1835               | 0.9340 | 1.32  | 0.1881 |
| TPLLevel*HiTPLast  | -0.7186  | 0.1923         | -1.0955               | -0.3416 | -3.74 | 0.0002 |

**Inductive Analysis**

Even though it was not the original intend of the study, we conducted an analysis from the results of the experiment using decision trees. The model generated (Figure 14) provided clear rules that go in line with our statistical results.
According to the decision tree model, providers are less likely to adjust their treatment options \((1=44, 0=76)\) when recommendations provided are of mixed costs. When low cost recommendations are presented, the adjusting probability differs depending on the order of time pressure. When time pressure is displayed first, providers are more likely to adjust to the recommended options \((1=39, 0=21)\). On the other hand, when time pressure is presented later, time pressure plays a role in the influence by system recommendations. Providers under low time pressure are more inclined to adjust their prescription \((1=21, 0=9)\), whereas providers under high time pressure are more likely to ignore system recommendations \((1=13, 0=17)\).

---

**Figure 14: Inductive Tree – Adjusting Treatment Options \(\{1 = X, 0 = Y\}\)**

**X:** Number of observations where prescription was altered to a lower-cost recommendation.

**Y:** Number of observations where prescription was altered to a lower-cost recommendation.

---

Overall, the experiment provided very interesting insights on how physicians would react to cost-sensitive recommendations. The experiment also indicated that recommendation costs as well as time pressure do play a significant role in viewing systems recommendations and adjusting treatment prescriptions. In practice, these results could be used by healthcare providers and insurance companies to influence physicians prescription behaviors, and eventually reduce overall healthcare costs.
In this section, we extend our influence dynamics model to include factors other than cost and time pressure. In collaboration with a few practicing physicians we learned that several influence dynamics would be anticipated to exist in the provider network. Under different levels of risk and time pressure, physicians are anticipated to view different amounts of information, and therefore accept recommendations at different rates. We present one such comprehensive model.

**High Time Pressure / High Risk**

This scenario represents the case of emergency department (ED) physicians. In this case, physicians ignore such systems in order to save time (Drescher, et al., 2011). In this scenario users will consider an alternative prescription for review only if the outcome of the recommended procedure significantly exceeds the outcome of the pre-selected procedure. Because of lack of time, physicians will most likely go with their chosen procedure and refrain from considering alternatives even if they are cost effective.

**Low Time Pressure / High Risk**

Under normal time pressure, physicians will be more disposed to evaluate a larger number of alternatives, especially when the protocol is not well defined or when a new drug is introduced. A typical example of this scenario would be oncology, where the medical community has not yet reached consensus regarding treatment options.

In this case, if the recommender system is trusted, providers will most likely consider procedures that have been shown to 1) yield better patient outcomes, and 2) represent the best cost alternative. It is important to note that physicians are expected to select the procedure that best fits their cost-related type.

**Low Time Pressure / Low Risk**

Under low time pressure, physicians are most likely to consider alternative options. Typically, primary care providers (PCP) would fall in this category. Those are physicians who usually work outside hospital settings, and are also more conscious about healthcare costs.
Additionally, the PCP’s office work setting allows for less time pressure, and therefore more opportunity to evaluate various recommendations. That is, even if the recommender system is not fully trusted, physicians can allocate the time to evaluate alternative procedures that are potentially beneficial.

| TP – High |
| --- | --- |
| Risk – Low ➔ <Ignore – Use Current> |
| Risk – High |
| Influence Probability (IP) – Low ➔ <Ignore – Use Current> |
| IP – High ➔ < Prob Swap – Use High Outcome Rec> |

TP – Low
| Risk – High |
| IP – Low ➔ <Ignore – Use Current> |
| IP – High ➔ < Prob Swap – Use High Outcome/Cost Aligned Rec> |
| Risk – Low ➔ <Prob Swap – Use High Outcome/Low Cost Rec> |

Figure 15: Influence Dynamics – Comprehensive Model

Figure 15 presents the influence dynamics under the different scenarios of varying physicians’ levels of time pressure, as well as procedure costs, outcomes, risks, influence predisposition. The individual paths in the tree are self-explanatory and map to the dynamics discussed in this section.

**Comprehensive Model Evaluation: Agent-Based Simulation**

To evaluate the comprehensive influence dynamics model presented above, we implemented a recommender system we refer to as Top-N++. Top-N++ is a Top-N recommender which provides procedure cost and outcome information at the time of prescription. That information is expected to alter the provider’s prescribing behavior; thereby allowing such systems to steer the prescription behavior towards better patient outcomes and lower healthcare costs.

The recommender keeps track of a list of alternative procedures pertaining to each diagnosis, along with the outcome, the cost, and the rate of prescription associated with each procedure.
Simulation Model

We consider a group of 100 providers. Each provider has a list of patients to be consulted daily.

Providers vary in terms of their individual attributes:

- Attitude towards the recommender system (high versus low trust).
- “Time Pressure Retardancy α”, “Delay-Pressure Factor β”, and “Time Pressure Capacity”.
- Attitude towards cost difference (PI, PQ, or PS).

Patients’ appointments are set at fixed intervals of time. However, delay is introduced stochastically during the simulation lifetime. Three sources of delay are included. The real patients’ arrival time includes a random delay, modeling the late arrival of some patients. If the delay exceeds a specific threshold, the appointment is cancelled, or re-scheduled for a later date. The consultation time includes a random delay, as some consultations might exceed the expected allocated time. With a small probability, X minutes are added the provider’s daily delay to model any unexpected emergency cases.

Using a traditional top-N algorithm, the recommender displays the three popular most prescribed procedures. However, the Top-N++ also presents information about the cost, the patient outcome, and the percentage of prescription associated with each procedure.

The provider first pre-selects a procedure, and then evaluates the list of procedures presented by the recommender using the influence dynamics described earlier.

Measures

In order to evaluate the performance of the TOP-N ++ recommender under time pressure, we consider three measures, namely 1) the recommendations acceptance rate (M1), 2) the recommendations’ swap-eligible rate (M2), and 3) the overall cost savings (M3).

The percentage of used recommendations represents the precision of the algorithm and is defined as:

\[
M1 = \frac{\text{Count of Accepted Recommendations}}{\text{Count of Recommendations Displayed}}
\]
The percentage of swap-eligible procedures presented measures the relevance of recommendation provided by the system, and is calculated as follows:

\[ M_2 = \frac{\text{Count of Swap Eligible Recommendations}}{\text{Count of Recommendations Displayed}} \]

To better illustrate the effect of using medical recommenders on costs, we also measure cost savings. Cost savings is defined as the difference in cost between the pre-selected procedure and the recommended one. In case the provider is not influenced by the recommender system, the cost savings is considered to be null. Note however, that the system recommends procedure based on percentage of prior prescription by other providers, and based not on cost. Therefore, depending on the provider type, the cost of the recommended procedure could be higher than the pre-selected one; in which case, the cost savings amount will be negative.

\[ M_3 = \sum (\text{Cost}_{\text{PreSelected}} - \text{Cost}_{\text{Recommended}}) \]

**Simulation Results**

Using an agent-based simulation we analyzed the TOP-N++ recommender’s performance under various levels of time pressure.

The analysis considers different scenarios based on the providers’ types, risk, outcome and cost variance. Below we present sample results under a few cases and discuss the interpretation.

Figure 16A illustrates how the Top-N++ generated positive outcome benefits for all types of patients. That is because physicians are typically influenced by the recommender agent only if the recommendations presented provide significantly higher patient outcomes. However, it is to be noted that as time pressure increases, providers tend to ignore recommendations for low risk patients; which results in much lower acceptance rates and lower overall outcome benefits.

Figure 16B compared the costs savings generated by providers of different cost types while treating high risk patients.
When patient delays are low, providers are under low time pressure and tend to select recommended procedures that are aligned with their cost type. Therefore, when a large percentage of providers are price-sensitive, low cost recommendations are selected and positive cost savings are generated. Under the same settings, a group of mostly price-praising providers generates negative cost savings. As delay builds up and time pressure increases, providers do not take the time to evaluate cost effective alternatives. However, it is interesting to see how the initial selections made by physicians under low time pressure is impacted the subsequent selections. Because initially selected (cost effective) procedures gained higher prescriptions rates, they were included in the Top-N++ recommendations list used in subsequent iterations. Thus, under high time pressure, the group of price-sensitive providers continued to generate positive cost savings, while price-praising providers generated negative cost savings.

Looking at Figure 16C, we can see that the TOP-N++ generated better outcome benefits when the outcome variance among alternative treatment options was relatively large. In practice, Top-N++ recommenders would therefore be more effective in cases where treatment options significantly vary in outcome, such as cancer treatments or alternative medicine.

Figure 16D shows that treatments with high cost variance only generated significant cost savings under low time pressure.

That is because providers need time to evaluate lower cost treatment options and determine how suitable they might be for each specific patient. Such cases include considering generic versus brand name prescription drugs.
Conclusion

This study explored the use of recommender systems in the medical practice in order to reduce the elevated healthcare costs. Such practices are characterized by low influence by recommendations mainly due to time pressure.

Our strategy to lowering healthcare costs strategy leverages medical recommender systems by presenting procedure cost information at the time prescription. To our knowledge, this is the first study to make use recommender systems for the purpose of cost reduction.

Results from our field experiment completed by physicians revealed some very interesting insights on how medical practitioners would be influenced by such systems.
In summary, keys findings indicate a general inclination among physicians to reduce patients’ share of cost. However, this influence effect was shown be moderated by recommendations attributes such as cost variance, and contextual attributes such as time pressure.

When recommendations presented are all less expensive than the procedure initially selected, the influence rates are significantly high. Evidence also shows that consultation of both, viewing of and influence by recommendations, are significantly lower under high time pressure.

Other factors impacting the use of recommendations in the medical settings also included outcome, risk, and influence predisposition. The influence dynamics of such factors were identified with collaboration with domain experts. In settings of high time pressure and high risk, cost-sensitive recommendations were anticipated to be ignored. Under low time pressure, recommendations were more likely to be evaluated; and eventually used. The influence by recommendations was also anticipated to be higher for physicians with higher influence predisposition such as novice providers.

The evaluation of our cost-sensitive recommender was performed using an agent-based simulation under various scenarios of risk, outcome, and influence predisposition. Results indicate generally positive cost savings from using the recommender system; confirming our experiment results. Savings were also less substantial in high time pressure cases where recommendations tend to be ignored. Cost savings were also minimal when the majority of providers were price-praising; associating high cost procedures with higher outcome.

**Research Contributions**

This study provides an initial understanding on the physicians’ use of cost information presented through recommender systems. We show how simple recommender systems that incorporate procedure cost can result in significant cost savings and better outcomes in healthcare.

In field experiment research, the study provides a very interesting contribution on how to design experiments with a time pressure treatments.
Our findings clearly indicate that applying high time pressure towards the end of the experiment triggers a high sense of time pressure; as opposed to providing a higher workload and time constraint upfront. A plausible explanation being that when high loads are presented first, they are perceived to be the norm, and hence do not produce time pressure.

**Research Implications**

Recently, there have been several initiatives to reduce healthcare costs in the US indicating both the importance and urgency of the matter. Our findings from the field suggest that presenting similar-outcome low-cost alternatives to physicians at the time of prescription would be well adopted by physicians in the practice; leading to an overall reduction of healthcare costs.

Our simulation results indicate that such systems might not be equally effective in different healthcare sectors. In environments of high risk and high time pressure settings, for example, these low-cost recommendations would most likely be ignored. Such systems might also create additional burden for physicians whenever time is scarce. The practical implications are that, when implementing cost-sensitive recommender systems, it is important to identify, and take into account, the characteristics of the specific setting in which the system will be used.

**Recommendations for future work**

While this study indicates the potential success of our novel strategy in practice, additional research is needed to generalize and advance our knowledge.

Even though our experiment was completed by physicians, and therefore provided a relatively high level of reliability, our sample size was small. More research is needed to duplicate the study and generalize our findings. Because a convenient sample was used, the majority of participants were specialized in internal medicine. Additionally, the medical cases used in the experiment were pertaining to primary care; limiting generalizability. Future research is needed to assess the physicians influence by such cost-sensitive recommendations in different specialties where cost variance among similar-outcome alternatives might be more or less relevant, and where time pressure might be more significant.
Last, our cohort of subject consisted mostly of highly experienced physicians; which might have biased our results. Medical residents and less experienced doctors might react differently to recommendations provided by the system.

If recommendations provided by the system are deemed reliable, novice providers might be more inclined to use the system recommendations; which would be viewed as the general practice. These providers would also include a younger generation, relatively more acquainted with the use of recommender systems in general.

In the recommender systems research, more studies are needed to assess the effect of time pressure on recommendation adoption. Because of the increasing prevalence of recommender systems in different settings, such as online retail, a better understanding of factors impacting their use is crucial. Such understanding would enable the design of more effective recommender systems; leading to higher returns. Personalized recommenders, for example, could be designed to learn and take into account time pressure in order to display person-tailored as well as context-tailored recommendations.

References


CHAPTER 4: PATIENTS’ COMPLIANCE
A SURVEY AND SENTIMENT ANALYSIS

Introduction

Patient compliance has been the focus of clinical research for decades. Tremendous amounts of research have addressed the issue of treatment noncompliance among patients with various diseases. Because noncompliance has been shown to generate poor patient outcomes and waste of medical resources (Gruman 2010), it has been the subject of investigation by scholars of various disciplines since the 1960’s (Davis 1968; Swartz et al. 1965).

Through this study, we look at the issue of patient compliance, and how it relates to patient health outcome, and healthcare costs. In this study, we first define patient noncompliance. We then attempt to assess the magnitude of the noncompliance issue through a literature survey of its various outcomes as well as its economic impact. We then present 1) the different measures of non-compliance, and 2) factors leading to noncompliance present in the literature (Figure 17).

![Figure 17: Literature Survey - Framework](image)

Looking at the extensive literature in this domain, we survey the data analytics techniques used. We then propose different ways data mining could be used to complement this stream of research.
We use sentiment analysis to examine patients’ data in the special case of taking the drug Lipitor. Results from the field experiment indicating potential for these techniques are then reported.

**Literature Review**

Through this literature review, we first look at the different definitions of patient compliance. We then investigate how low medication compliance relates to patient outcomes and healthcare costs. Last, we survey the different measures used in the compliance stream of research, and identify factors leading to noncompliance.

**Compliance Definition**

Patient compliance refers to the degree of a patient’s adherence to prescribed medical recommendations (Davis 1968). The term compliance therefore encompasses various concepts ranging from compliance with postoperative instructions (Marshall, Penckofer, & Llewellyn, 1986), to medication adherence, and preventive care (Lin et al., 2004). The extent of non-adherence categorizing patients as noncompliant varies greatly from one study to another, making generalization difficult (Farmer 1999). Depending on the purpose of the study and the type of disease, researchers elect to define the extent of compliance differently. In some studies, the noncompliance threshold is set at 60% adherence to the prescribed drugs (Rand & Wise, 1994). In some other instances, patients scoring in the bottom 20% of the compliance scale are the ones referred to as noncompliant (Chin & Goldman, 1997). This variance in operationalization of noncompliance also impacts the results of statistical analyses. A different categorization of compliance has identified five different categories namely errors of omission, errors of dosage, mistakes in timing, taking medications for the wrong reason, and taking additional medication not prescribed (Wade & Bowling, 1986).

**Patient Compliance and Outcome**

Regardless of how it is defined, patient noncompliance has been recognized as a chief concern in healthcare. In prior literature, patients’ non or low compliance has been linked to suboptimal outcomes (Gruman 2010).
A meta-analysis of 63 studies over three decades indicates a 26% outcome difference between high and low adherence patients (Dimatteo, Giordani, Lepper, & Croghan, 2002). The analysis included compliance with a vast range of doctor recommendations including diet, exercise, and blood pressure measurement. In Table 16 below, we survey outcomes related to medication non-adherence in various diseases.

*Mortality & Survival*

Studies of compliance in chronic diseases have been using mortality rates as a major measure for outcome. In a clinical trial with 2175 patients with a history of myocardial infarction, non-adherence to Beta Blockers treatments (taking <= 75 % of prescribed medication) were 2.6 times more like than good adherers to die within 12 months (Horwitz et al., 1990). In cancer studies, investigators have been using survival rates to indicate the effects of the adherence. Among women with breast cancer, for example, the hazard ratio for mortality in relation to duration at 2.4 years was 0.85, where adherence <80%, was associated with poorer survival hazard ratio of 1.10 (McCowan et al., 2008).

*Hospitalization*

Medication noncompliance has also been linked to patient hospitalization. In a study involving elderly patients, 11.4% of hospitalizations were due to noncompliance (Col N, 1990). More frequent readmissions were reported among noncompliant mental health patients (Haynes, McKibbon, & Kanani, 1996). Another widely used measure in the compliance-outcome literature is hospital readmission. In a study of patients surviving heart failure, even though no correlations were found significant, the investigators used early readmission rates (<60 days) as measure of outcome (Chin & Goldman 1997).

*Violence*

When combined with substance abuse, noncompliance with medical treatment has been linked to violence among mental health patients (Swartz et al., 1965). While this outcome is specific to severely ill patients, the impact of these violent acts extend from the patients themselves to impacting the community at large.
Symptom Reduction, Disease Recovery and Relapse

One of the most widely used, and perhaps the most sought after, outcome of compliance with treatment, is the effectiveness in alleviating symptoms, eradication of disease and avoidance of relapse. In a study of Staphylococcus aureus treatment, compliant patients were more likely to cure the infection and less likely to relapse (Fowler et al., 1998). Lower relapse probabilities have also been reported among patients adhering to depression treatment guidelines (Melfi et al., 1998). Studies among patients with chronic disease also revealed a pattern of symptom reduction among compliant patients.

Looking at symptoms reduction, forty-three percent of high-adherent patients attained their target blood pressure goal compared with thirty-three percent and thirty-four percent of patients with medium or low adherence to antihypertensive monotherapy (Bramley, Gerbino, Nightengale, & Frech-Tamas, 2006). Similarly, each incremental 25% increase in proportion of days covered for statin medications was associated with an 3.8-mg/dL reduction in LDL cholesterol (Ho, Bryson, & Rumsfeld, 2009). Using HbA to measure outcome, 10% increase in non-adherence to metformin and statins was associated with an increase of 0.14% in HbA (Pladevall et al., 2004).

Disease Progression

Disease progression has been used as a measure for outcome linked to low medication adherence among Human Immunodeficiency Virus patients. However, findings among these studies have been indicating contradicting results. While some studies advocate the adherence to the treatment therapy (Chesney, Ickovics, Hecht, Sikipa, & Rabkin, 1998), others indicate drug resistance, and thereby lower clinical outcome linked to adherence (Bangsberg et al., 2001).

Even though noncompliance has been linked to various suboptimal healthcare outcomes, that relationship seems to be difficult to quantify. In a study looking at patients with Coronary Artery Disease (Carney, Freedland, Eisen, Rich, & Jaffe, 1995), it has been suggeted that depression is associated with poor adherence to treatment regimens.
Since depression is also associated with increased medical morbidity and a two-to fourfold increase in mortality in older medical patients (Horwitz et al., 1990), studies linking compliance to mortality should consider the depression as a confounding factor in the analysis. Noncompliance outcomes are also difficult to assess for cases where 1) noncompliance is prevalent, or 2) outcomes are less observable - asymptomatic diseases such as hypertension (Hays et al., 1994).

<table>
<thead>
<tr>
<th>Noncompliance Outcome</th>
<th>Disease</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mortality</td>
<td>Coronary Heart Disease</td>
<td>(Horwitz et al., 1990)</td>
</tr>
<tr>
<td>Survival</td>
<td>Cancer</td>
<td>(Hershman et al., 2011)</td>
</tr>
<tr>
<td>Elderly Hospitalization</td>
<td>Cardiac diseases, falls, gastrointestinal diseases, chronic obstructive pulmonary disease, and pneumonia</td>
<td>(Col N, 1990)</td>
</tr>
<tr>
<td>Hospital Readmission</td>
<td>Congestive Heart Failure</td>
<td>(Chin &amp; Goldman, 1997)</td>
</tr>
<tr>
<td>Violence</td>
<td>Mental Health</td>
<td>(Swartz et al., 1965)</td>
</tr>
<tr>
<td>Disease Recovery</td>
<td>Infectious Disease</td>
<td>(Fowler et al., 1998)</td>
</tr>
<tr>
<td>Disease Relapse</td>
<td>Depression</td>
<td>(Melfi et al., 1998)</td>
</tr>
<tr>
<td>Symptom Reduction</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BP Control</td>
<td>Hypertension</td>
<td>(Bramley et al., 2006)</td>
</tr>
<tr>
<td>LDL Control</td>
<td>Cardiovascular Disease</td>
<td>(Ho et al., 2009)</td>
</tr>
<tr>
<td>HbA Level</td>
<td>Diabetes</td>
<td>(Pladevall et al., 2004)</td>
</tr>
<tr>
<td>Disease Progression</td>
<td>Human Immunodeficiency Virus</td>
<td>(Chesney et al., 1998)</td>
</tr>
</tbody>
</table>

Economic Impact of Patient Noncompliance

Patient noncompliance has also been linked to waste of medical resources causing higher healthcare costs (Gruman et al., 2010). The annual cost of non-adherence has been estimated at US$300 billion dollars (Bender & Rand, 2004).

A study examining medication waste reports that medications discarded by patients over 65 represent 2.3% of medication cost; translating to over $1 billion per year (Bender & Rand, 2004; Morgan, 2001). Other costs incurred by noncompliant patients include costs due to hospitalization, lost productivity, and premature deaths.
Among others, disease-specific examples of the economic impact of noncompliance include adherence to antidepressants, tyrosine-kinase inhibitors, and immunosuppressant drugs. Adherence to antidepressant therapy indicates differences in yearly medical costs averaging about $450. Among cancer patients prescribed Imatinib, low compliant patients were observed to incur an additional $US 4072 in medical costs annually compared with less compliant patients (Darkow et al., 2007). Persistent low immunosuppression compliance after kidney transplant was linked to $12,840 increase in individual 3-year medical costs (Pinsky et al., 2009).

However, it is to be noted that when costs are adjusted for prescription and other medical costs, differences between adherent and non-adherent costs might be altered. Among patients with depression, “when antidepressant prescription costs were added to medical costs, patients requiring a therapy change and titrating therapy incurred higher costs than adherent patients, whereas non-adherent and adherent patients incurred similar costs.” (Cantrell, Eaddy, Shah, Regan, & Sokol, 2006). In other cases, such as diabetes, hypercholesterolemia, and cardiovascular disease, drug costs among compliant patients are offset by other non-drug costs among noncompliant patients; generating overall cost savings (Muszbek, Brixner, Benedict, Keskinaslan, & Khan, 2008; Sokol, McGuigan, Verbrugge, & Epstein, 2005).

Looking at its patient outcomes and economic impacts, noncompliance seems to be a persistent issue. A survey of past research as well as understanding the status quo are therefore necessary in order to advance our knowledge in this domain and help remedy the problem.

**Compliance Measures**

Even though compliance has been subject to research for decades, there still is no gold standard to measure it (Farmer, 1999). However, well defined set of measures have been used in compliance research each with its own advantages and disadvantages. Compliance measures used in prior research could be classified as either direct or indirect (Table 17).
Direct Measures of Compliance

Direct measures of compliance are ones that provide proof that the patient has taken the drug (Farmer 1999). These include objective measures such as drug assays, biological markers, and patient observation.

Drug Assays of Bodily Fluid

This measurement method uses lab tests on the patients’ bodily fluids to indicate the existence of the drug. For blood drug assays, venous blood is taken for the patient. Plasma is then separated by centrifugation, frozen, and then analyzed for the molecules composing the drug to be taken (George, Peveler, Heiliger, & Thompson, 2000). While this measure might be accurate, it is relatively invasive. Because it requires the collection of patients’ bodily fluids, this measure suffers from low subject-acceptance.

Biological Markers

Biological markers are compounds added to the medication that can later be detected in lab test of biological fluids. These additives are non-toxic and include substances such as digoxin and phenol red. Biological markers have been used recently in clinical trials and can provide proof that patient has been taking the drug (Farmer 1999).

Pill Count

The pill count measure uses a percentage of the drug tablets used over the number of tablets that should have been taken (George et al. 2000). The tablet count is performed within a pre-defined period, usually set to 1 week intervals.

In their study of 1988, Rudd and colleagues have shown that compliance measures based on pill counts were consistent with pre-indicated marked inter-subject and intra-subject variability (Rudd et al. 1988). Because of ease of administration, this method has been widely used in clinical trials.

Patient Observation

This measure of compliance relies on personal observation of the patient while taking the drug.
Patient compliance could be very accurate measure but might be inconvenient and time-consuming from the researchers’ point of view. Even though difficult, patients might still manage to avoid taking the drug in disguise (Farmer, 1999).

Indirect Measures of Compliance

Indirect measures of compliance provide a proxy for non-compliance. These include any measures that have not directly retrieved by interaction with the patient. Those include qualitative measures such as patient reported information about drug-taking, as well as quantitative data collected through monitoring devices or retrieved from the patient’s records.

Patient-Reported Medication-Taking Behavior

Perhaps the least invasive, but most unreliable measure of compliance, patient reported medication taking behavior has been used extensively (Farmer 1999). One of the widely used measures is the Morisky score which is based on four standard questions (Morisky, Green, & Levine 1986):

- Do you ever forget to take your medication?
- Are you careless at times about taking your medicine?
- When you feel better, do you sometimes stop taking your medicine?
- Sometimes when you feel worse, do you stop taking your medicine?

Patients receives a 1 score for each positively asserted answer, and 0 otherwise. Therefore, a total score of 0 indicates full compliance, and a score of four suggests major noncompliance.

Other similar instruments include the Steward two-question interviews (Stewart 1986), and the recently developed Brief Medication Questionnaire (BMQ) (Svarstad, Chewning, Sleath, & Claesson 1999).

In general, the patient questionnaires and interviews measure compliance has been validated in several studies. In their study of compliance with tricyclic antidepressants, George et al. have shown that self-reported scores have proven to be useful in measuring compliance (George et al. 2000).
This compliance measure has been used mainly because of ease of administration. Also, being qualitative measure, the questionnaires have been used to indicate reasons leading to noncompliance. The accuracy of this measure has been evaluated through sensitivity and specificity analyses; which have been relatively high but varied across studies.

Other shortcomings of the patient-reported measure include the necessity to administer different questionnaires for each medication the patient is taking (Farmer 1999). Also, because it is a self-reported measure, these scores might suffer from biases such as recall (Farmer 1999).

**Electronic Monitoring Devices (EMD)**

Electronic medical devices such as the Medication Event Monitoring System (MEMS), are microprocessors placed on the medication bottle cap, which records the time, date, and frequency of bottle cap opening (Bova et al. 2005). This technology enabled measure provides an objective and precise assessment of medication adherence behavior (George et al. 2000). However, practical issues related to the use of such systems have been reported resulting in limiting its use. In their 2005 study, Bova and colleagues have reported 41% of their sample patients taking more than one dose at a time. 26% reported opening the bottle but not taking the drug. Also, more than one third of participants electing to use the EMD as part of the study have discontinued the use.

In another study performed by Kudielka and colleagues in 2003, Salivette Sampling Devices were used to monitor patient adherence to Cortisol intake. Those are electronic monitoring devices that process saliva samples provided by the patients and records results. The study reports that only 74% of participants were found to follow sampling instructions in ambulatory settings; which, according to the authors, partially invalidates the results of the study (Kudielka, Broderick, & Kirschbaum 2003).
Pharmacy Records Review

In the late 80s, pharmacy records have been identified pharmacy records as data source for measuring patients’ drug compliance. Data retrieved from pharmacy databases includes the drug name, drug dosage, the quantity of medication dispensed at each pharmacy fill, and the dates of prescription refills (Steiner, Koepsell, Fihn, & Inui, 1988). Hence, several measures are computed from the pharmacy data collected such as compliance distribution, number of refill intervals, and the time period the medication was available (Steiner et al. 1988).

In their 1988 study, Steiner and colleagues confirmed that refill compliance (RC) is a valid and useful method of measuring compliance. Results indicated a direct relationship between compliance as measured by the refill data and drug effects for most patients in the study. In a study performed in 1999, Choo and colleagues reported that adherence levels measured through pill counts and pharmacy records were more accurate than the ones reported by electronic monitoring devices.

Claims Review

Claims data has been recently introduced as a measure for medication compliance. Measures such as length of therapy (LOT) and medication possession ratio (MPR) are calculated from claims data to assess adherence levels. Initial results indicate a correlation between claims-based compliances and clinical outcomes in patients with diabetes (Pladevall et al. 2004). Other results link LOT and MPR-measured compliant to lower cost among antidepressant-adherent patients (Cantrell et al. 2006). However, authors of the study recognize the need for additional research to assess the validity of the method (Pladevall et al. 2004).


Table 17: Noncompliance Measures

<table>
<thead>
<tr>
<th>Noncompliance Measure Type</th>
<th>Noncompliance Measure</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct</td>
<td>Drug Assays of Blood</td>
<td>(George et al., 2000)</td>
</tr>
<tr>
<td></td>
<td>Drug Assays of Urine</td>
<td>(Cone, Caplan, Black, Robert, &amp; Moser, 2008)</td>
</tr>
<tr>
<td></td>
<td>Biological Markers</td>
<td>(Farmer, 1999)</td>
</tr>
<tr>
<td></td>
<td>Drug Refill</td>
<td>(Steiner &amp; Prochazka, 1997)</td>
</tr>
<tr>
<td></td>
<td>Pill Counts</td>
<td>(Rudd et al., 1988)</td>
</tr>
<tr>
<td></td>
<td>Patient Observation – taking the drug</td>
<td>(Farmer, 1999)</td>
</tr>
<tr>
<td>Indirect</td>
<td>Patient-Reported of Medication-Taking Behavior</td>
<td>(Edmonds et al., 1985)</td>
</tr>
<tr>
<td></td>
<td>Electronic Monitoring Device (EMD)</td>
<td>(Svarstad et al., 1999)</td>
</tr>
<tr>
<td></td>
<td>(e.g. Medication Event Monitoring System (MEMS))</td>
<td>(Kudielka et al., 2003)</td>
</tr>
<tr>
<td></td>
<td>Pharmacy Records Review</td>
<td>(Bova et al., 2005)</td>
</tr>
<tr>
<td></td>
<td>Prescription Claims Review</td>
<td>(George et al., 2000)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Steiner et al., 1988)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Choo et al., 1999)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Pladevall et al., 2004)</td>
</tr>
</tbody>
</table>

Non-Compliance Factors

Even though a vast majority of compliance research has focused on specific groups of patients, reviews of treatment characteristics revealed common factors leading to noncompliance across patients with different diseases (Haynes et al. 1996) except in psychiatric disorders (Haynes & Sackett, 1979). Prior research attributes higher levels of noncompliance to a plethora of potential factors pertaining to the patients themselves, the medical provider, the prescribed treatment/recommendation regimen, the relationship between the provider and the patient, and the prescription’s contextual characteristics (Table 18).

Factors of Noncompliance- Patient

In several studies, the patient has been placed at the heart of the noncompliance problem. Patient attributes leading to noncompliance include age, severity of the medical problem, socio-medical attitudes, personality traits (Davis 1968), forgetfulness (Miller 1997), lack of medication education (Vlasnik, Aliotta, & DeLor 2005), and exemption status (Beardon et al. 1993). Also, social support seems to play an important role in patient compliance particularly among adolescents (Ammassari et al. 2002).
Other major attributes in the compliance literature includes patient beliefs (Vermeire, Hearnshaw, Van Royen, & Denekens, 2001) and attitudes (Pound et al. 2005) towards medication taking. Because the reported rates of non-compliance are high (about one third to one half of all patients), some researchers posit that noncompliance cannot be due to simple factors such ignorance or forgetfulness (Donovan 1992). Research suggests that patients actively engage in making the decision of not following recommended regimens. In the same school of thought, a significant portion of the compliance literature has used the Health Belief Model to explain noncompliant behavior. The Health Belief Model is a conceptual framework that is used to understand reasons of patients’ compliance or lack thereof (Janz & Becker, 1984). The model has been developed in 1974, and has been applied extensively in the literature to understand noncompliance behavior for different diseases (Barker, Cook, Kahook, Kammer, & Mansberger, 2013; Hall 2012). The model consists of four main dimensions (Janz & Becker 1984):

- Perceived Susceptibility: Patient’s subjective perception of the risk of contracting the condition.
- Perceived Severity: Patient’s perceived seriousness of the condition.
- Perceived Benefits: Patient’s belief regarding the effectiveness of the prescribed regimen.
- Perceived Barriers: Patient’s perceived negative aspects of the regimen such as side effects.

The model includes a dimension referred to as “cue to action”; which includes factors causing the patient to start the decision-making process. These cues include attributes like symptoms, or triggers such as media or interpersonal interactions (Janz & Becker 1984).

Factors of Noncompliance- Prescribed Regimen

Regarding treatment regimen, repeated-prescriptions (Cline et al. 1999), preventive medicines (Beardon et al. 1993), and medicines with known side-effects (Donovan 1992) have shown to be trigger the highest levels of noncompliance. Pound and colleagues also mention that some patients are cautious about taking medications because of potential severe adverse drug reactions (Pound et al. 2005).
In asymptomatic conditions such as hyperlipidemia, patients do not observe disease related symptoms and therefore underestimate the need for medication (Miller 1997). Therefore, treatments of asymptomatic diseases as well as preventive medications suffer from high patient noncompliance.

**Factors of Noncompliance- Medical Encounter**

Other factors of noncompliance include the patient physician relationship where trust plays an important role in adherence.

Higher compliance is therefore observed with prescriptions made by medical doctors versus trainees (Beardon, et al. 1993). Other attributes such as doctor’s responsiveness seem to also impact patient compliance.

**Factors of Noncompliance- Prescription Context**

Interestingly, the context of prescription has also been shown to significantly impact compliance. When comparing compliance rates of treatment prescribed during the weekend versus the weekdays, weekend issued prescription suffered a higher noncompliance rates (Beardon et al. 1993). As noted by the authors, this finding seems counter-intuitive, since weekend consultations often include high emergency cases. However, as explained by the authors, those patients could be the most disabled and might not be able to reach the pharmacy.

To remedy non-compliance issues, various strategies have been implemented. Interventions ranging from simply informing patients of the prescribed medication to telephone follow ups and patients counseling, therapy, video games (Kato, Cole,Bradlyn, & Pollock 2008) and rewarding have been proven to be partially successful at deterring the noncompliant behavior (R. Brian Haynes et al. 1996).

These strategies have been geared towards informing and empowering the patients to take responsibility in following prescribed regimens. These techniques have been proven to be relatively effective at increasing compliance in the short-run. However, more longitudinal studies are needed to quantify the sustainability of such measures.
### Table 18: Noncompliance Factors

<table>
<thead>
<tr>
<th>Non-Compliance Factor</th>
<th>Factor-Compliance Relationship</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age</td>
<td>Elderly heart failure patients have higher non-compliance</td>
<td>(Cline, 1999)</td>
</tr>
<tr>
<td>Gender</td>
<td>Women are more likely to be noncompliant</td>
<td>(Beardon et al., 1993)</td>
</tr>
<tr>
<td>Exemption Status</td>
<td>1/3 patient who fail to redeem medication have to pay charges</td>
<td>(Beardon et al., 1993)</td>
</tr>
<tr>
<td>Medication Education</td>
<td>Limited Medication education/awareness</td>
<td>(Vlasnik et al., 2005)</td>
</tr>
<tr>
<td>Forgetfulness</td>
<td>More prevalent among asymptomatic disease patients</td>
<td>(Miller, 1997)</td>
</tr>
<tr>
<td>Social Support</td>
<td>Important among adolescents</td>
<td>(Ammassari et al., 2002)</td>
</tr>
<tr>
<td>Patient Beliefs</td>
<td>Formed by patients own, as well as family members and friends’ knowledge, ideas and experiences.</td>
<td>(Vermeire et al., 2001)</td>
</tr>
<tr>
<td>Attitude towards medication-taking</td>
<td>Varying degrees of resistance to medication taking</td>
<td>(Pound et al., 2005)</td>
</tr>
<tr>
<td>Adverse Drug Reactions</td>
<td>Might lead to hospital admission</td>
<td>(Pound et al., 2005)</td>
</tr>
<tr>
<td>Side Effects</td>
<td>Fear of side effect such as gastrointestinal problems</td>
<td>(Donovan, 1992)</td>
</tr>
<tr>
<td>Repeat Preventative</td>
<td>Non-redemption is higher for repeat prescriptions</td>
<td>(Cline, 1999)</td>
</tr>
<tr>
<td>Preventative</td>
<td>Preventative treatments are more prone to noncompliance</td>
<td>(Miller, 1997)</td>
</tr>
<tr>
<td>Asymptomatic Frequently Modified / Misleading Prescription</td>
<td>Non-adherence to asymptomatic prescriptions</td>
<td>(Miller, 1997)</td>
</tr>
<tr>
<td>Prescribing Physician</td>
<td>High noncompliance among trainees</td>
<td>Beardon, et al. 1993</td>
</tr>
<tr>
<td>Medical Encounter</td>
<td>Doctor responsiveness</td>
<td>(Weddington WW, 1988)</td>
</tr>
<tr>
<td>Time of Prescription</td>
<td>Prescriptions issued at weekend had higher non-redemption rates</td>
<td>(Beardon et al., 1993)</td>
</tr>
</tbody>
</table>
Data Analytics in Patient Compliance Research

In prior compliance literature, various methods of analysis have been used. In measuring compliance, several studies have use ANOVA to compare different measures of compliance, and how they relate to drug effects (Choo et al 1999; Hays et al. 1993).

Regression analysis has been used to analyze the relationship between noncompliance and associated factors (Pinsky et al., 2009). Multivariate analysis was also used to depict any moderating factors that could like compliance and outcome (Horwitz et al., 1990). More recently, survival analysis has been used to measure the effect of adherence on the survival outcome (McCowan et al., 2008). Other statistical analyses used in the literature include Cox proportional-hazards modeling (Chin 1997). None of the studies in the compliance literature used data mining techniques.

Because of the increasing use of patients’ electronic medical records, portals, and forums, we expect these techniques, when applied to patients’ data, to help advance compliance research.

To investigate the efficiency of such algorithms, we perform a field experiment; which uses sentiment analysis on patient reviews to 1) predict compliance, and 2) discover factors leading to noncompliance.

Sentiment Analysis

For this experiment, we look at the special case of compliance with taking the drug Lipitor (a statin drug) as recommended by physicians for treating Hyperlipidemia. Hyperlipidemia is very common is the US. According to the 2015 report from The American Heart Association, 73.5 million adults (31.7%) have LDL cholesterol levels above the ideal 130 mg/dL (Mozaffarian D, 2014). Hence, cholesterol-lowering drugs are widely used drug in the US.

According to a Health report published by the US Department of Health and Human Services in 2014, “the use of cholesterol-lowering drugs among adults ages 18-64 has increased more than six-fold since 1988-1994, due in part to the introduction of statins” (Hyattsville, 2014). Among adults aged 65 and over, 70.2% took at least one cardiovascular agent and 46.7% took a cholesterol-lowering drug in the past 30 days in 2007–2010.
Data Description

The data used for this experiment was collected online. The data included 936 patients’ ratings and reviews for the drug Lipitor, and was publicly available at *AskaPatient.com*.

The data collected included several pieces of information that allowed for 1) measuring compliance at a patient level, and 2) extracting factors of non-compliance among the patients. The list of variable extracted are listed in Table 19 below.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rating</td>
<td>Numerical score from 1 to 5; 1 being worst rating.</td>
<td>Perceived Value of the Drug</td>
</tr>
<tr>
<td>Reason</td>
<td>Text data entered by the patient</td>
<td>Disease/Symptom Description</td>
</tr>
<tr>
<td>Side Effects</td>
<td>Text data entered by the patient</td>
<td>Perceived Side Effects</td>
</tr>
<tr>
<td>Comment</td>
<td>Text data entered by the patient</td>
<td>Patient’s Opinion about the Drug</td>
</tr>
<tr>
<td>Gender</td>
<td>Binomial value set to either Male or Female</td>
<td>Patient’s Gender</td>
</tr>
<tr>
<td>Age</td>
<td>Numerical value ranging from 19 to 89</td>
<td>Patient’s Age</td>
</tr>
<tr>
<td>Duration</td>
<td>Number of days, weeks, months, or years.</td>
<td>Patient’s reported duration of drug intake to date of the review</td>
</tr>
<tr>
<td>Dosage</td>
<td>Dosage in mg (10, 20, 30, or 40).</td>
<td>Patient’s reported dose of drug taken</td>
</tr>
<tr>
<td>Review Date</td>
<td>System date</td>
<td>Review Date</td>
</tr>
</tbody>
</table>

Data Labeling

In order to utilize the data for analysis this needed to be labeled appropriately. The labeled data served as a training set for data mining models. For this study, a group of graduate students volunteered to complete the labeling task. The students were provided instructions on how to label each of the patients’ reviews as either indicating compliance or non-compliance.

Students were instructed to label each patient review as an instance of “noncompliance” if there was indication in the text that the patient is *not* taking the drug (Lipitor) as recommended by his/her doctor. Otherwise, if it could be inferred from the text that the patient was taking the medication as recommended, the patient’s note was to be labeled as an instance of “compliance”. The assumption was that the patient was compliant unless otherwise specified. Labeling examples (Table 20) were provided to the students before the task.
### Table 20: Labeling Examples

<table>
<thead>
<tr>
<th>Label</th>
<th>Text</th>
</tr>
</thead>
</table>
| COMPLIANT      | Sore arms and pins and needles  
not sleeping and getting bad sweats all night only happens when i take the tab  
Side Effects: None. Quick reduction in LDL. No side effects.  
Side Effects: None. My levels went from 518 down to 175 to 190. No muscle pain or side effects. I also take 2000 MG of Niacin a day. Liver functions normal.  
Side Effects: None. Maintains my cholesterol levels at normal.                                                                                                                                 |
| NON-COMPLIANT | I took Lipitor for my total cholesterol if 301! But i have depression and take lexapro 20 mg. as my dose of lipitor was increased from 10 to 20mg i started having serious knee pain and depressive mode. Came off of Lipitor and within few days i felt much better.  
I experienced permanent nerve damage to my feet severe leg cramping. Symptoms started as a warming sensation on my lower legs after a few months and progressed to the permanent nerve damage. I discontinued taking Lipitor over 5 years ago. |

Among the 936 reviews initially selected for the analysis, 48 were dropped because the student were not able to indicate whether the patient writing the review was compliant or not. The final set of reviews used for the analysis therefore included 888 labeled notes.

**Sentiment Analysis**

Sentiment analysis included two major steps: 1) data transformation, and 2) document classification.

**Data Transformation**

Labeled data was divided into individual documents to allow for text classification. Text data was then processed using a series of steps (Figure 18). First, text data was transformed to all lower case characters. Text was then tokenized on the space character, allowing us to keep important numerical data such as treatment dosage and duration. Common words in the English alphabet were filtered from the data set. We then used the Porter stemming algorithm to reduce each term to its basic form (e.g. experiencing → experience). Our last processing step was to generate bi-grams. Those are two-word combinations that seem to appear frequently in the text.
Text Classification

Data in this experiment was analyzed using a Support Vector Machine algorithm. Based on the statistical learning theory (Vapnik, 1999), the algorithm maps the data into a higher dimensional input space and then constructs an optimal separating hyper-plane in this space (Suykens & Vandewalle, 1999). The algorithm used for this analysis is an SVM operator available through RapidMiner; referred to as Evolutionary SVM. This operator has been described as using an evolutionary algorithm to solve the dual optimization SVM problem.

Experiment Results

The model was evaluated using a 70-30 validation method. The accuracy level on test data reached 73.68%. Note that the initial labeled data set consisted of 478 reviews labeled as noncompliant and 410 reviews labeled as compliant. Even though the accuracy level is relatively low, the model provided interesting insights on characteristics differentiating between compliant and non-compliant patients.

Sample model attributes (Table 21), seem to provide indications as to why some patients are compliant with the drug intake and others are not. Most relevant terms used to classify patients as compliant indicate that compliant patients are the ones who perceive the drug as being effective in reducing their cholesterol level (reduce, lowered cholesterol). Those are also patients who were taking 20mg of the drug and were less likely to report experiencing any side effect (none). On the other hand, noncompliant patients seem to have experienced side effects such as pain, stiffness, and depression. Some of those patients might have also elected to use dieting as an alternative to taking the drug.
### Table 21: Sample Model Attributes

<table>
<thead>
<tr>
<th>Compliance Attribute</th>
<th>Weight</th>
<th>Non-Compliance Attribute</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time_Taken_months</td>
<td>2.03</td>
<td>Stop</td>
<td>-3.22</td>
</tr>
<tr>
<td>High_Cholesterol_Years</td>
<td>1.76</td>
<td>Take</td>
<td>-2.59</td>
</tr>
<tr>
<td>Cholesterol</td>
<td>1.72</td>
<td>Drug</td>
<td>-2.25</td>
</tr>
<tr>
<td>Time_Taken_years_High</td>
<td>1.69</td>
<td>Pain</td>
<td>-1.96</td>
</tr>
<tr>
<td>Years_Reason_High</td>
<td>1.64</td>
<td>Dai</td>
<td>-1.89</td>
</tr>
<tr>
<td>Months_reason_high</td>
<td>1.52</td>
<td>Symptom</td>
<td>-1.80</td>
</tr>
<tr>
<td>Reduce</td>
<td>1.15</td>
<td>Week</td>
<td>-1.69</td>
</tr>
<tr>
<td>lxd</td>
<td>1.14</td>
<td>Month</td>
<td>-1.65</td>
</tr>
<tr>
<td>Lowered_cholesterol</td>
<td>0.94</td>
<td>Quit</td>
<td>-1.57</td>
</tr>
<tr>
<td>20 mg</td>
<td>0.92</td>
<td>Doctor</td>
<td>-1.53</td>
</tr>
<tr>
<td>None</td>
<td>0.91</td>
<td>Muscle</td>
<td>-1.49</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Statin</td>
<td>-1.40</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Weak</td>
<td>-1.31</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Loss</td>
<td>-1.28</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Walk</td>
<td>-1.25</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Again</td>
<td>-1.23</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Recommend</td>
<td>-1.07</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Gradual</td>
<td>-1.02</td>
</tr>
<tr>
<td></td>
<td></td>
<td>cramp</td>
<td>-1.01</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Leg</td>
<td>-0.97</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Permanent</td>
<td>-0.95</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Depress</td>
<td>-0.93</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Diet</td>
<td>-0.925</td>
</tr>
</tbody>
</table>

**Conclusions**

Looking at the different outcomes and costs associated with noncompliance, the literature review performed in this paper reiterates the importance of the issue. Patient noncompliance has been shown to result in patient outcomes of various severity. In chronic diseases such as coronary heart and obstructive pulmonary diseases, noncompliance has led to an increase in mortality, hospitalization, and hospital readmission rates. In other diseases like hypertension and diabetes treatment noncompliance has caused worsening of disease and symptoms, a slower recovery, and higher rates of disease relapse.

Estimated at about $300 billion a year, costs of noncompliance mainly encompassed 1) costs related to worsening patient outcomes as presented earlier, 2) costs due to lost productivity, and 3) costs due to premature deaths.
A large portion of noncompliance costs also included the ones due to medications being discarded. These findings reported in the literature indicate that noncompliance is an important and persistent issue that would benefit from any advances led by research and/or practice.

Compliance measures were categorized as direct and indirect. Direct measures of compliance were mostly used in clinical trials and require personal contact with the patients. These measures included drug assays of biological fluids, pill counts, and patient observation taking the drug. Indirect measures make use of patients’ secondary data such as pharmacy and prescriptions records to assess compliance. These measures also include data reported electronic medical devices (e.g. MEMS) which record timing and frequency of medication bottle opening patterns.

Even though measured in different ways, noncompliance factors seem to be common across different diseases. Studies investigating reasons of noncompliance reported factors characterizing the patients themselves, the prescribed regimen, the doctor-patient encounter, and the prescription context.

Overall, prior research made use of traditional statistical analysis to measure compliance, investigate the impact of various factors on noncompliance, and estimate the outcome and economic impacts of noncompliance. However, none of the studies used data mining techniques. Limited use of patients records were used to measure noncompliance through pharmacy and claims records.

In this study, we offer a major contribution by suggesting better leverage of patients’ data in the compliance research. Comprehensive data to be used for the analysis shall not be restricted to pharmacy or claims records, but be extended to include patients structured data retrieved from the electronic medical records, patients patterns of healthcare usage, and text data entered through patient portals and forums.

A sample analysis was performed using an experiment on publicly available patient reviews. Data was scraped from the online forum, labeled by graduate students, and then used for sentiment analysis. Initial findings allowed the prediction of patient compliance at 73% accuracy. The analysis also allowed the extraction of several factors leading to noncompliance in that patient cohort.
Overall, experiment results suggested promising potentials for such analysis in this domain. Additional data mining techniques could be used in this domain as listed in the recommendations for future work below.

**Recommendations for Future Work**

In light of findings from the literature survey and early results from the field experiment, several recommendations could be made for future work. Those focus mainly on applying supervised data mining for noncompliance measurement and prediction. Unsupervised data mining techniques could also be used to uncover factors and patterns leading to noncompliance.

**Compliance Measures**

*Research:* In order to allow for generalizability of research findings, more research needs to be done to create consensus regarding the definition and operationalization of compliance. Even though the definitions of compliance generated might differ by disease or groups of disease, it should be consistent among each segment of research.

Research on measures of noncompliance could be extended to include analysis of data entered by patients through portals and forums. Structured data extracted from electronic medical records such as doctors follow up appointments, as well as text data entered through the forum could be leveraged to measure compliance. By annotating the different ways the compliance and noncompliance concepts in patients’ text (reviews), information extraction routines could be built to automate the process of compliance measurement. More research is needed to design and evaluate such algorithms in this domain.

*Practice:* In practice, early identification of noncompliance information could be automatically transmitted to physicians through the electronic health record; potentially enabling reduction of noncompliance rates.

**Compliance Factors**

*Research:* Sentiment analysis seems to provide promising results when it comes to determining factors of noncompliance, as well as prediction of compliance. More research is needed to design more efficient algorithms geared specifically towards mining patient text data.
Practice: Since new analytical tools such as sentiment analysis are promising in this domain, policy makers should encourage the creation and use of patient portals and forums. These are mediums designated for patients to enter opinions and feedback about treatments in particular and opinions about health behavior in general. By mining the patients’ reviews and opinions, healthcare providers would have the ability to proactively install measures that would prevent noncompliance.

If abundant patient data is available, intelligent models could be built to provide 1) early prediction of noncompliance especially for rare diseases and newly marketed treatments, and 2) noncompliance interventions customized to every patient’s special need. Looking the cost incurred by noncompliance, subsidizing such portals could potentially be beneficial.

Compliance and Related Outcomes/Costs

Patient compliance certainly is a chief issue that requires the attention of every player in the healthcare community. With major impact on patient outcomes and healthcare costs, more effort is needed to 1) measure compliance, 2) understand the various and newly evolving factors of noncompliance, and 3) develop interventions that are geared towards alleviating noncompliance.

Research: More research needs to be done to differentiate between areas where high noncompliance is associated with different levels of patient outcomes and healthcare costs (Table 22).

<table>
<thead>
<tr>
<th>Table 22: Compliance - Outcome - Cost Relationship</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Patient Outcome</strong></td>
</tr>
<tr>
<td>High, High</td>
</tr>
<tr>
<td>Low, High</td>
</tr>
</tbody>
</table>

Practice: In the practice, efforts to implement interventions to increase patient compliance should thereby be placed in cases with the most impact. Practitioners should promote compliance whenever that later is suggested to lead to higher patient outcomes; regardless of cost. If compliance is associated with higher costs, healthcare provider could focus to providing similar-outcome, low-cost alternatives.
References


APPENDICES

Appendix A: Analytical Model - Proofs

SP: Service Provider - Practitioner (e.g. medical provider)
IP: Insurance Provider

\(p_1\): Probability to audit practitioners \textit{targeted} by the deterrence algorithm
\(p_2\): Probability to audit practitioners \textit{not targeted} by the deterrence algorithm
\(\Psi\): SP’s fraud probability
\(\rho\): Mark-up included in fraudulent claims
\(\gamma\): Penalty imposed on audited-fraudulent practitioners
\(\varepsilon\): Proportion of influential service practitioners
\(\Phi_1\): The additional benefit from auditing an \textit{influential fraudulent} practitioner
\(\Phi_2\): The additional benefit from auditing an \textit{influential non-fraudulent} practitioner performing waste and abuse

\(P_{DFI}\): Probability that the \textit{deterrence} algorithm generates an alarm for a \textit{fraudulent influential} practitioner
\(P_{DFNI}\): Probability that the \textit{deterrence} algorithm generates an alarm for a \textit{fraudulent non-influential} practitioner
\(P_{NFII}\): Probability that the \textit{deterrence} algorithm generates an alarm for a \textit{non-fraudulent influential} practitioner
\(P_{NFNI}\): Probability that the \textit{deterrence} algorithm generates an alarm for a \textit{non-fraudulent non-influential} practitioner.

\(P_D^T\): Probability that the \textit{deterrence} algorithm generates an \textit{alarm in case of fraud}
\(P_F^T\): Probability that the \textit{deterrence} algorithm generates an \textit{alarm in case of no fraud}

\[P_D^T = \varepsilon \cdot P_{DFI} + (1-\varepsilon) \cdot P_{DFNI}\]
\[ P_T^f = \epsilon \cdot P_{NF}^I + (1-\epsilon) \cdot P_{NF}^{NI} \]

\( P_{\text{Alarm}} \): Probability that the \text{deterrence} algorithm \textit{generates} an alarm

\( P_{\text{No Alarm}} \): Probability that the \text{deterrence} algorithm \textit{doesn’t generate} an alarm

\[ P_{\text{Alarm}} = \psi \cdot P_D^T + (1-\psi) \cdot (P_D^T - P_T^f) \]

\[ P_{\text{No Alarm}} = 1 - P_{\text{Alarm}} \]

\( P_{NI-\text{Fraud/Alarm}} \): Given an alarm, probability that a \textit{non-influential} practitioner \text{defrauds}

\( P_{I-\text{Fraud/Alarm}} \): Given an alarm, probability that an \textit{influential} practitioner \text{defrauds}

\( P_{I-NI-\text{Fraud/Alarm}} \): Given an alarm, probability that an \textit{influential} practitioner \text{does not defraud}

\[ P_{NI-\text{Fraud/Alarm}} = \frac{(1-\epsilon) \cdot P_{NF}^{NI} \cdot \psi}{P_{\text{Alarm}}} \]

\[ P_{I-\text{Fraud/Alarm}} = \frac{\epsilon \cdot P_I^T \cdot \psi}{P_{\text{Alarm}}} \]

\[ P_{I-NI-\text{Fraud/Alarm}} = \frac{\epsilon \cdot P_{NF}^{I}(1-\psi)}{P_{\text{Alarm}}} \]

\( P_{NI-\text{Fraud/NoAlarm}} \): Given no alarm, probability that a \textit{non-influential} practitioner \text{defrauds}

\( P_{I-\text{Fraud/NoAlarm}} \): Given no alarm, probability that an \textit{influential} practitioner \text{defrauds}

\( P_{I-NI-\text{Fraud/NoAlarm}} \): Given no alarm, probability that an \textit{influential} practitioner \text{does not defraud}

\[ P_{NI-\text{Fraud/NoAlarm}} = \frac{(1-\epsilon) \cdot P_{NF}^{NI} \cdot (1-P_F)}{P_{\text{NoAlarm}}} \]

\[ P_{I-\text{Fraud/NoAlarm}} = \frac{\epsilon \cdot P_I^T \cdot (1-P_F)}{P_{\text{NoAlarm}}} \]

\[ P_{I-NI-\text{Fraud/NoAlarm}} = \frac{\epsilon \cdot (1-\psi) \cdot (1-P_{NF}^{I})}{P_{\text{NoAlarm}}} \]

\( P_{\text{Audit/Fraud}} \): Probability of audit given fraud
\[ P_{\text{NoAudit/Fraud}} : \text{Probability of no audit given fraud} \]

\[ P_{\text{Audit/Fraud}} = (p_1 P_{DF}) + (p_2 (1-P_{DF})) \]

\[ P_{\text{NoAudit/Fraud}} = 1 - P_{\text{Audit/Fraud}} \]

\[ F_{\text{SP}}: \text{Payoff of service provider} \]

\[ F_{\text{SP}} = \psi [P_{\text{Audit/Fraud}}(\rho - \gamma) + P_{\text{NoAudit/Fraud}}(p)] \]

\[ \frac{\partial F_{\text{SP}}}{\partial \psi} = -\rho (p_1 (P_{FI} \varepsilon - P_{FNI} (\varepsilon - 1)) + p_2 (P_{FNI} (\varepsilon - 1) - P_{FI} (\varepsilon + 1)) - 1) - (\gamma - \rho) (p_1 (P_{FI} \varepsilon - P_{FNI} (\varepsilon - 1)) + p_2 (P_{FNI} (\varepsilon - 1) - P_{FI} \varepsilon + 1)) \]

Solve \[ \frac{\partial F_{\text{SP}}}{\partial \psi} = 0 \] for \( p_1 \) given \( p_2 = k \)

\[ p_1^*: \text{Probability to audit practitioners targeted by the deterrence algorithm at equilibrium} \]

\[ p_2^*: \text{Probability to audit practitioners not targeted by the deterrence algorithm at equilibrium} \]

\[ p_1^* = \frac{\rho + k (1 - \gamma)}{(\gamma (\varepsilon P_{FI}^2 + (1-\varepsilon) P_{FNI}^2))} \]

Solve \[ \frac{\partial F_{\text{SP}}}{\partial \psi} = 0 \] for \( p_2 \) given \( p_1 = 1 \)

\[ p_2^* = \frac{-(\rho - P_{FNI}^2 (1-\varepsilon) \lambda - P_{FI}^2 \varepsilon \lambda)}{\lambda (P_{FNI}^2 (1-\varepsilon) + P_{FI}^2 \varepsilon - 1))} \]

\[ F_{\text{Alarm}} : \text{Payoff of insurance provider given in case of alarm} \]

\[ F_{\text{NoAlarm}}: \text{Payoff of insurance provider given in case of no alarm} \]

\[ F_{\text{Alarm}} = -(1 - p_1).P_{\text{NI-Fraud/Alarm}} - p_1.(\rho - \gamma).P_{\text{NI-Fraud/Alarm}} - (1 - p_1).(p).P_{\text{I-Fraud/Alarm}} \]
\[ -p_1(\rho - \gamma \cdot \Phi_1) \cdot P_{\text{1-Fraud/Alarm}} - p_1(\rho \cdot \Phi_2) \cdot P_{\text{1-Not Fraud/Alarm}} - p_1 \cdot c \]

\[ F_{\text{No Alarm}} = -(1 - p_2) \cdot (\rho \cdot P_{\text{Not Fraud/No Alarm}} - p_2(\rho - \gamma) \cdot P_{\text{Not Fraud/No Alarm}} - (1 - p_2) \cdot (\rho) \cdot P_{\text{1-Fraud/No Alarm}} \]

\[ - p_2(\rho - \gamma \cdot \Phi_2) \cdot P_{\text{1-Fraud/No Alarm}} - p_2(\rho \cdot \Phi_2) \cdot P_{\text{1-Not Fraud/No Alarm}} - p_2 \cdot c \]

\[ \frac{\partial F_{\text{Alarm}}}{\partial p_1} = - \left( \frac{P_{\text{NI}}^1 \cdot (c - \Phi_2) + P_{\text{NI}}^1 \cdot c \cdot \psi (1 - \epsilon) - P_{\text{NI}}^1 \cdot (\rho - \gamma) - P_{\text{NI}}^1 \cdot c \cdot \psi (1 - \epsilon) + P_{\text{NI}}^1 \cdot c \cdot \psi (c - \gamma - \Phi_2) - P_{\text{NI}}^1 \cdot c \cdot \psi (c - \Phi_2)}{(P_{\text{NI}}^1 \cdot (1 - \epsilon) + P_{\text{NI}}^1 \cdot c \cdot (1 - \epsilon))} \right) \]

\[ \frac{\partial F_{\text{No Alarm}}}{\partial p_2} = \left( \frac{-c + P_{\text{NI}}^1 \cdot c \cdot \psi (1 - \epsilon) + P_{\text{NI}}^1 \cdot c \cdot \psi (c - \Phi_2) + \gamma + P_{\text{NI}}^1 \cdot c \cdot \psi (1 - \epsilon) - P_{\text{NI}}^1 \cdot c \cdot \psi (1 - \epsilon) \cdot P_{\text{NI}}^1 \cdot c \cdot \psi (1 - \epsilon) + P_{\text{NI}}^1 \cdot c \cdot \psi (c - \gamma - \Phi_2)}{(P_{\text{NI}}^1 \cdot (1 - \epsilon) + P_{\text{NI}}^1 \cdot c \cdot (1 - \epsilon) + P_{\text{NI}}^1 \cdot c \cdot (1 - \epsilon) - P_{\text{NI}}^1 \cdot c \cdot \psi (1 - \epsilon) - P_{\text{NI}}^1 \cdot c \cdot \psi (1 - \epsilon) - 1) \right) \]

Solve \( \frac{\partial F_{\text{Alarm}}}{\partial p_1} = 0 \) and \( \frac{\partial F_{\text{No Alarm}}}{\partial p_2} = 0 \) for \( \psi \) \( \rightarrow \) Empty Set

We can verify that \( \frac{\partial F_{\text{Alarm}}}{\partial p_1} > \frac{\partial F_{\text{No Alarm}}}{\partial p_2} \). Therefore, at equilibrium, \( \frac{\partial F_{\text{Alarm}}}{\partial p_1} = 0 \) and \( \frac{\partial F_{\text{No Alarm}}}{\partial p_2} < 0 \) or \( \frac{\partial F_{\text{No Alarm}}}{\partial p_2} = 0 \) and \( \frac{\partial F_{\text{Alarm}}}{\partial p_1} > 0 \)

\( \Psi_1^* \): SP’s fraud probability at equilibrium – Strategy 1

\( \Psi_2^* \): SP’s fraud probability at equilibrium – Strategy 2

Solve \( \frac{\partial F_{\text{Alarm}}}{\partial p_1} = 0 \) for \( \psi \) given \( \frac{\partial F_{\text{No Alarm}}}{\partial p_2} < 0 \)

\[ \Psi_1^* = \frac{(P_{\text{NI}}^1 \cdot c \cdot (1 - \epsilon) + P_{\text{NI}}^1 \cdot c \cdot (c - \Phi_2))}{(P_{\text{NI}}^1 \cdot c \cdot (1 - \epsilon) + P_{\text{NI}}^1 \cdot c \cdot (1 - \epsilon) + P_{\text{NI}}^1 \cdot c \cdot (c - \gamma + \Phi_1))} \]

Solve \( \frac{\partial F_{\text{No Alarm}}}{\partial p_2} = 0 \) for \( \psi \) given \( \frac{\partial F_{\text{Alarm}}}{\partial p_1} > 0 \)

\[ \Psi_2^* = \frac{(c - \Phi_1 - \Phi_2) - P_{\text{NI}}^1 \cdot c \cdot \psi (1 - \epsilon) - P_{\text{NI}}^1 \cdot c \cdot \psi (c - \Phi_2)}{(\gamma + c \cdot \Phi_2) + P_{\text{NI}}^1 \cdot c \cdot \psi (1 - \epsilon) - P_{\text{NI}}^1 \cdot c \cdot \psi (c - \gamma - \Phi_2) - P_{\text{NI}}^1 \cdot c \cdot \psi (c - \Phi_2)} \]
Appendix B: Experiment Instrument

Patient 1

Name: James Smith  DOB: 02/25/1972  Age: 42  Sex: Male

Visit Date: SYS DATE  Visit Type: Problem Visit  Visit
Provider: YOU  Primary Plan: BCBS

Problem List
Duodenal Ulcer  Status  Active

Medication List

Prescribed within Practice
Ranitidine  150mg/12h
Maalox  200-200- 20mg/5mL

Allergy List
Latex Exam Gloves
Sulfur (rash)

Clinical Alerts
New Endoscopic Exam Needed

Vital Signs

<table>
<thead>
<tr>
<th>Date</th>
<th>BP</th>
<th>HR</th>
<th>RR</th>
<th>T(F)</th>
<th>Wt</th>
<th>Ht</th>
<th>O2</th>
</tr>
</thead>
<tbody>
<tr>
<td>12/20/2014</td>
<td>124/78</td>
<td>82</td>
<td>17</td>
<td>98.5</td>
<td>146lbs 3oz</td>
<td>5'6&quot;</td>
<td>99%</td>
</tr>
<tr>
<td>04/07/2010</td>
<td>125/78</td>
<td>84</td>
<td>15</td>
<td>98.8</td>
<td>145lbs 6oz</td>
<td>5'6&quot;</td>
<td>99%</td>
</tr>
</tbody>
</table>

Chief Complaint
  - Epigastric Pain
  - Weight Loss

History of Present Illness

James Smith is a 42 year old Caucasian male who presents today for recurrent epigastric pain treated in the last year with ranitidine. Patient experience loss of weight. He lost 5 pounds within the last month. Exacerbation of pain after meals. Patient has had endoscopic exam with biopsy that revealed the presence of 1 cm bulbar ulcer in the posterior part of the duodenum.

Past Medical History
Duodenal ulcer for 10 years

Family Medical History

Significant for Hypertension

Social History

Significant for Caffeine (Current); College graduate, 2 year; Divorced;

SOAP Note

VS
- Height: 64.0 in
- Weight: 140.3 lb
- BMI: 34.7
- Blood Pressure: 130/78 mmHg
- Temp: 98.6 F
- Pulse: 70 bpm
- Resp Rate: 12 rpm

CC  Epigastric Pain, weight loss


A  Duodenal Ulcer

P  TO BE DETERMINED

PLAN

Next, you will see a list of drugs to be prescribed for this patient. Please select the medication list you view as most appropriate for this specific patient. Note that this list might not be comprehensive.

AminoPenicillin
- Amoxicocot
- Apo-Amoxi
- Amoxil
- DisperMox
- Moxatag
- Moxilin
- Trimox
- Wymox
Macrolide
- Biaxin
- Biaxin XL
- Biaxin XL-Pak

Nitroimidazole Antimicrobial
- Flagyl
- MetroCream
- Metrogel
- Noritate
- Rosadan
- Vandazole
- Vitazol

PPI
- Prilosec
- Omesec
- Losec
- Dexilant
- Nexium
- Prevacid
- Zegerid
- Protonix
- Aciphex
- Kadipex

Prostaglandin E1 Analog
- Arthrotec
- Cyprostol
- Cytotec
- Mibetec
- Oxaprost
Patient 2

Name: Shawn Jones  DOB: 02/25/1960  Age: 55  Sex: Male
Visit Date: SYS DATE  Visit Type: Problem Visit
Provider: YOU  Primary Plan: BCBS

Problem List  Status
Diabetes Mellitus, Type II Active
Hypertension Active

Medication List  Dose

*Prescribed within Practice*
Metformin 2 500mg
Captopril 2 25mg

Allergy List
None

Clinical Alerts
Diabetics: Eye Exam Needed
Diabetic: Foot Exam

Vital Signs

<table>
<thead>
<tr>
<th>Date</th>
<th>BP</th>
<th>HR</th>
<th>RR</th>
<th>T(F)</th>
<th>Wt</th>
<th>Ht</th>
<th>O2</th>
</tr>
</thead>
<tbody>
<tr>
<td>06/20/2014</td>
<td>135/80</td>
<td>82</td>
<td>17</td>
<td>98.5</td>
<td>246lbs</td>
<td>5'8&quot;</td>
<td>99%</td>
</tr>
<tr>
<td>12/07/2013</td>
<td>130/79</td>
<td>84</td>
<td>15</td>
<td>98.8</td>
<td>240lbs</td>
<td>5'8&quot;</td>
<td>99%</td>
</tr>
</tbody>
</table>

Chief Complaint
- Diabetes follow-up

History of Present Illness

Shawn Jones a 55 year old Caucasian male who comes in for a follow-up visit. In the previous encounter, patient’s dose of metformin was increased to 500mg, 3 times a day. Patient presented today with an A1C level of 9% and glucose test of 220. Patient not following recommended diet and physical activity.

Past Medical History

Diabetes Type II
Hypertension
Family Medical History

Social History

Significant for Alcohol (Current); College graduate, 4 year; Married;

SOAP Note

<table>
<thead>
<tr>
<th>VS</th>
<th>Height:</th>
<th>Weight:</th>
<th>BMI:</th>
<th>Blood Pressure</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>68.0 in</td>
<td>250.0 lb</td>
<td>34.7</td>
<td>140/85 mmHg</td>
</tr>
</tbody>
</table>

| CC | follow up diabetes, BP |


| A  | Hypertension, Diabetes II |

| P  | TO BE DETERMINED |

**PLAN**

Next, you will see a list of drugs to be prescribed for this patient. Please select the medication list you view as most appropriate for this specific patient. Note that this list might not be comprehensive.

- Biguanides
  - Fortamet
  - Glucophage
  - Glucophage XR
  - Glumetza
  - Riomet
Sulfonylureas
- DiaBeta
- Glycron
- Glynase
- Micronase
- Glipizide XL
- Glucotrol
- Glucotrol XL
- Amaryl

Meglitinides
- Glufast
- Starlix
- Prandin

Thiazolidinediones
- Actos
- Avandia
- Rezulin

DPP-4 inhibitors
- Tradjenta
- Onglyza
- Januvia

GLP-1 receptor agonists
- Tanzeum
- Byetta
- Victoza
- Lyxumia

SGLT2 inhibitors
- Invokana
- Farxiga
- Suglat

Alpha-Glucosidase Inhibitors
- Precose
- Glyset
- Voglib

Bile Acid Sequestrants
- Questran
- Welchol
- Colestid
- Colestipid
Combination Pills
- Metaglip
- Glucovance
- Duetact
- Actoplus Met
- Prandimet
- kombiglyze
- Janumet

Insulin therapy
- Novolog
- Levetiracetam
- Lantus
- Apidra
- Humulin N
- Novolin N
- Humalog
Patient 3

Name: Claudia Santiago  DOB: 04/25/1995  Age: 19  Sex: Female
Visit Date: SYSTEM DATE  Visit Type: Problem Visit

Provider: YOU  Primary Plan: Aetna

Problem List

Asthma

Status

Active

Medication List

Prescribed within Practice

<table>
<thead>
<tr>
<th>Drug</th>
<th>Dose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Loratidine</td>
<td>1- 10mg</td>
</tr>
<tr>
<td>Fluticasone</td>
<td>2- 50mcg</td>
</tr>
</tbody>
</table>

Prescribed outside Practice

Albuterol      As needed

Allergy List

Aspirin
Non-steroid anti-inflammatory drugs

Clinical Alerts

Vital Signs

<table>
<thead>
<tr>
<th>Date</th>
<th>BP</th>
<th>HR</th>
<th>RR</th>
<th>T(F)</th>
<th>Wt</th>
<th>Ht</th>
<th>O2</th>
</tr>
</thead>
<tbody>
<tr>
<td>11/04/20014</td>
<td>124/84</td>
<td>78</td>
<td>15</td>
<td>98.8</td>
<td>158lbs</td>
<td>5'6&quot;</td>
<td>99%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3oz</td>
<td></td>
<td></td>
</tr>
<tr>
<td>07/07/20013</td>
<td>124/79</td>
<td>83</td>
<td>15</td>
<td>98.8</td>
<td>147lbs</td>
<td>5'6&quot;</td>
<td>99%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>7oz</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Chief Complaint

- Dyspnea
- Wheezing

History of Present Illness

19 year old female comes in for worsening of asthma symptoms. She refers to difficulty breathing with effort. Constant dry cough. The condition is worse at night. Wheezing.

Past Medical History

Chronic sinusitis; Allergic Rhinitis; Usual Childhood disease

Family Medical History
Social History

Non-smoker; High-school graduate; Exercises regularly

SOAP Note

<table>
<thead>
<tr>
<th>VS</th>
<th>Height: 66.0 in</th>
<th>Weight: 150.0 lb</th>
<th>BMI: 28.7</th>
<th>Blood Pressure: 121/68 mmHg</th>
<th>Temp: 98.6 F</th>
<th>Pulse: 70 bpm</th>
<th>Resp Rate: 28 rpm</th>
</tr>
</thead>
<tbody>
<tr>
<td>CC</td>
<td>Asthma</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>Here for worsening of asthma symptoms. Shortness of breath. Wheezing. Taking medications without difficulty.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>Asthma</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P</td>
<td>TO BE DETERMINED</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Next, you will see a list of drugs to be prescribed for this patient. Please select the medication list you view as most appropriate for this specific patient. Note that this list might not be comprehensive.

**Adrenergic Bronchodilators**
- AccuNeb
- Airet
- Proventil
- Proventil HFA
- Ventolin
- Ventolin HFA
- Volmax
- Vospire ER
- Adrenalin
- Adrenalin Chloride
- Asthmahaler
- Auvi-Q
- EpiPen
- Primatene Mist
- Twinject
- Isuprel
- Isuprel Mistometer
- Medihaler-Iso
- Xopenex
- Xopenex Concentrate
- Xopenex HFA
- Alupent
- Orciprenaline
- Metaprel
- Brethine
- Bricanyl
- Brethine

**Anticholinergics Bronchodilators**
- Tudorza Pressair
- Atrovent
- Atrovent HFA
- Spiriva
- Spiriva Respimat

**Methylxanthines**
- Dilor
- Dylix
- Lufyllin
- Theo-24
- Theo-Dur
- Uniphyl
Leukotriene Modifiers
- Singulair
- Accolate
- Zyflo

Inhaled Corticosteroids
- Aerospan
- Qvar
- Pulmicort
- Asmanex
- Flovent

Bronchodilator Combinations
- Combivent
- Symbicort
- Advair Diskus
- Advair HFA
- Anoro Ellipta

Oral Corticosteroids
- Baycadrion
- Cortef
- Orapred
Patient 4

Name: Jessica Korman     DOB: 11/28/1949     Age: 65     Sex: Female
Visit Date: SYSTEM DATE     Visit Type: Problem Visit
Provider: YOU     Primary Plan: United Health

Problem List

Chronic Ischemic Heart Disease     Active

Medication List

Prescribed within Practice

Prescribed outside Practice
Aspirin 1 – 50 mg
Nitroglycerin 3 – 1 mg

Allergy List

Clinical Alerts

Vital Signs

<table>
<thead>
<tr>
<th>Date</th>
<th>BP</th>
<th>HR</th>
<th>RR</th>
<th>T(F)</th>
<th>Wt</th>
<th>Ht</th>
<th>O2</th>
</tr>
</thead>
<tbody>
<tr>
<td>04/20/2013</td>
<td>130/78</td>
<td>82</td>
<td>17</td>
<td>98.5</td>
<td>190lbs3oz</td>
<td>5'6&quot;</td>
<td>99%</td>
</tr>
<tr>
<td>07/07/2012</td>
<td>135/79</td>
<td>83</td>
<td>15</td>
<td>98.8</td>
<td>186lbs7oz</td>
<td>5'6&quot;</td>
<td>99%</td>
</tr>
</tbody>
</table>

Chief Complaint

- Nocturnal Cough
- Fatigue

History of Present Illness

65 year old African-American female, diagnosed with chronic ischemic heart disease in 2013, refers dyspnea during her ordinary activities and asthenia. Symptoms began 3 months ago, and worsened in the past 2 weeks.
Past Medical History
Chronic Ischemic Heart Disease

Family Medical History
Mother died with stroke at the age of 60

Social History
College graduate, 4 year; Married

SOAP Note

<table>
<thead>
<tr>
<th>VS</th>
<th>Height: 64.0 in</th>
<th>Weight: 202.0 lb</th>
<th>BMI: 32.6</th>
<th>Blood Pressure: 130/78 mmHg</th>
<th>Temp: 98.6 F</th>
<th>Pulse: 70 bpm</th>
<th>Resp Rate: 12 rpm</th>
</tr>
</thead>
<tbody>
<tr>
<td>CC</td>
<td>Dyspnea, nocturnal cough, fatigue</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>Here for worsening cardiac failure symptoms. Pt suffers from dyspnea, asthenia.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>Chronic Ischemic Heart Disease</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P</td>
<td>TO BE DETERMINED</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

PLAN
Next, you will see a list of drugs to be prescribed for this patient. Please select the medication list you view as most appropriate for this specific patient. Note that this list might not be comprehensive.
Calcium Channel Blocking Agents
- Norvasc
- Cardizem
- Diltzac
- Tiazac
- Cardene IV
- Adalat CC
- Nifediac CC
- Procardia
- Calan
- Isoptin
- Verelan

Cardiac Glycoside
- Cardoxin
- Lanoxicaps
- Lanoxin

Vasodilators
- Nitro-Bid
- Nitrostat
- Rectiv

Angiotensin Converting Enzyme Inhibitors
- Capoten
- Monopril
- Aceon
- Enalapril

Peripheral Vasodilators
- Cyclospasmol
- Vosuprine
- Pavaco
- Papacon
- Pavagen

Angiotensin Receptor Blockers
- Edarbi
- Teveten
- Candesartan
- Cozaar
- Benicar
- Micardis
Statins
- Lipitor
- Lescol
- Mevacor
- Livalo
- Pravachol
- Crestor
- Zocor

Platelet Aggregation Inhibitors
- Ecotrin
- Fasprin
- Miniprin
- Clavix
- Clopirad
- Plavix
Patient 5

Name: Natasha Wood   DOB: 7/25/1991   Age: 23   Sex: Female
Visit Date: SYSTEM DATE   Visit Type: Problem Visit   Visit
Provider: YOU   Primary Plan: United Health

Problem List Status

Hypothyroid  Active

Medication List

<table>
<thead>
<tr>
<th>Prescribed within Practice</th>
<th>Dose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synthroid</td>
<td>25 mcg daily</td>
</tr>
<tr>
<td>Fioricet</td>
<td>325 mg one tablet every 6 hours as needed for headache</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Prescribed outside Practice</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Claritin</td>
<td>10 mg daily as needed</td>
</tr>
</tbody>
</table>

Allergy List

Macrodantin – emesis
NSAIDS/ ASA – GI Bleed
Food allergies: oranges – hives, Chocolate – anaphylaxis

Clinical Alerts

EKG
Echocardiography

Chief Complaint

- Fatigue

History of Present Illness

Pt. presents to the office for a routine checkup. She denies feelings of chest pain or pressure. She denies any edema or numbness in her extremities. She states she has felt chronic fatigue over the past three months.

Past Medical History

Hypothyroid x 2 yrs.

Family Medical History

Mother: HTN (alive)
Father: Stroke at age of 40
Maternal Grandmother: Ischemic heart disease (deceased)
Maternal grandfather: HTN (deceased)
Paternal grandmother: DM type II (deceased)
Paternal Grandfather: CAD, MI at age 52 (deceased)

Social History

Patient denies ever having used tobacco or alcohol. She lives alone and has never been married. She has no children. She drinks 3 cups of coffee every morning.

SOAP Note

<table>
<thead>
<tr>
<th>VS</th>
<th>Height: 62.0 in</th>
<th>Weight: 111.0 Lb3.2oz</th>
<th>BMI: 20.34</th>
<th>Blood Pressure: 110/80 mmHg</th>
<th>Temp: 98.5 F</th>
<th>Pulse: 70 bpm</th>
<th>Resp Rate: 18 rpm</th>
</tr>
</thead>
<tbody>
<tr>
<td>CC</td>
<td>Fatigue</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>23 y.o. with familiar antecedents of familial hypercholesterolemia and stroke at early age (like 40 in father), presents in routine exam high levels of cholesterol and triglycerides.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>O</td>
<td>General: African American female who appears her age, in no acute distress. Appears to have a flat affect.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Skin: Light brown, warm and dry. No lesions, rashes or ulcers. Skin turgor good</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Hair: texture is course, shoulder length black hair. Equal distribution with no areas of hair loss</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Chest: Symmetric expansions, no rales/ rhonchi/ wheezes noted. Respirations equal and clear throughout all lung fields</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Heart: RRR, S1 and S2 audible, No gallops or rubs, PMI @ 5th ICS @ mideclavicular line, no edema noted, peripheral pulses present</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Abdomen: Soft, non-tender, non-distended. Liver and spleen non palpable.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ears: TM pearly gray, bony landmarks visible, no bulging or drainage noted bilaterally.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Eyes: PERRLA, no erythema or visible discharge noted bilaterally</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Nose: No erythema or edema noted. No nasal discharge. Septum intact.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Throat: No visible exudates, no petechiae. Mucus membranes moist and pink. Teeth intact</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Neck: No lymphadenopathy noted. Thyroid non-palpable</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Neuro: CN II – XII intact, sensory intact, strength equal bilaterally, no tremors or nystagmus noted.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>Hyperlipidemia</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P</td>
<td>TO BE DETERMINED</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
PLAN

Next, you will see a list of drugs to be prescribed for this patient. Please select the medication list you view as most appropriate for this specific patient. Note that this list might not be comprehensive.

Statins
- Lipitor
- Lescol
- Mevacor
- Livalo
- Pravachol
- Crestor
- Zocor

Combination Statins
- Caduet
- Advicor
- Vytorin

Bile Acid-Binding Resins
- Prevalite
- WelChol
- Colestid

Fibrates
- Abirate
- Antara
- Tricor
- Triglide
- Lopid

Nicotinic Acid
- Niacor
- Niaspan
- Slo-Niacin

Selective Cholesterol Absorption Inhibitors
- Zetia
Patient 6

Name: Aliya White  DOB: 9/25/1967  Age: 47  Sex: Female
Visit Date:  Visit Type: Follow-up Visit
Provider: YOU  Primary Plan: United Health

Problem List

Hypertension  Active

Medication List

*Prescribed within Practice*

<table>
<thead>
<tr>
<th>Medication</th>
<th>Dose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Motrin PRN headaches</td>
<td>600mg 3-4x weekly</td>
</tr>
<tr>
<td>Exforge</td>
<td>10/320 mg tablet once daily</td>
</tr>
</tbody>
</table>

*Prescribed outside Practice*

Allergy List

Clinical Alerts

Vital Signs

<table>
<thead>
<tr>
<th>Date</th>
<th>BP</th>
<th>HR</th>
<th>RR</th>
<th>T(F)</th>
<th>Wt</th>
<th>Ht</th>
<th>O2</th>
</tr>
</thead>
<tbody>
<tr>
<td>02/05/2015</td>
<td>185/104</td>
<td>69</td>
<td>18</td>
<td>98.5</td>
<td>170lbs</td>
<td>5'6&quot;</td>
<td>99%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3oz</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Complaint

- Follow-up of physical exam – HTN
- Headaches

History of Present Illness

47 y.o. A.A. F presents to clinic for f/u of physical exam findings. Found to be hypertensive during physical exam 1 week ago. PCP ordered blood work.
Past Medical History
Hypertension x 20 yrs.

Family Medical History
Father has HTN, is on dialysis for renal failure. Mother has DM Type II.

Social History
Accountant, works 50-60 hrs/week, lives alone, poor diet: lots of fast food. Caffeine 2-3 /day, occasional EtOH, smokes 1 pack/day (27 pack-year hx). Would like to exercise more, but is often too tired.

SOAP Note

<table>
<thead>
<tr>
<th>VS</th>
<th>Height: 65.0 in</th>
<th>Weight: 168.0 lb</th>
<th>BMI: 28.0</th>
<th>Blood Pressure: 180/104 mmHg</th>
<th>Temp: 98.6 F</th>
<th>Pulse: 62 pbm</th>
<th>Resp Rate: 12 rpm</th>
</tr>
</thead>
<tbody>
<tr>
<td>CC</td>
<td>F/u of physical exam</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>47 y.o. A.A. F presents to clinic for f/u of physical exam findings. Found to be hypertensive during physical exam 1 week ago. PCP ordered blood work.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>O</td>
<td>BUN 35, SCr 1.8, 24-hr urine: &gt;1 g/day proteinuria, glucose: 99mg/dL. Lipid panel: TC: 240mg/dL, TG: 170mg/dL, HDL: 34mg/dL, LDL: 144 mg/dL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| A  | - Pt has uncontrolled HTN  
- Smoking, caffeine, stress and poor diet increase BP and risk of CV disease. Lifestyle modifications and smoking cessation will help to reduce BP. |
| P  | TO BE DETERMINED |

PLAN
Next, you will see a list of drugs to be prescribed for this patient. Please select the medication list you view as most appropriate for this specific patient. Note that this list might not be comprehensive.
Calcium Channel Blocking Agents
- Norvasc
- Cardizem
- Diltzac
- Tiazac
- Cardene IV
- Adalat CC
- Nifediac CC
- Procardia
- Calan
- Isoptin
- Verelan

Cardiac Glycoside
- Cardoxin
- Lanoxicaps
- Lanoxin

Vasodilators
- Nitro-Bid
- Nitrostat
- Rectiv

Angiotensin Converting Enzyme Inhibitors
- Capoten
- Monopril
- Aceon
- Enalapril

Peripheral Vasodilators
- Cyclospasmol
- Voxsuprine
- Pavaco
- Papacon
- Pavagen

Angiotensin Receptor Blockers
- Edarbi
- Teveten
- Candesartan
- Cozaar
- Benicar
- Micardis
Statins
- Lipitor
- Lescol
- Mevacor
- Livalo
- Pravachol
- Crestor
- Zocor

Platelet Aggregation Inhibitors
- Ecotrin
- Fasprin
- Miniprin
- Clavix
- Clopirad
- Plavix
Appendix C: Sample Simulation Code - Warranty

```java
package sim.app.virus.Warranty;
import sim.field.network.*;
import sim.field.continuous.*;
import sim.engine.*;
import sim.util.*;

// Read from csv file */
import java.io.BufferedReader;
import java.util.StringTokenizer;
import java.io.*;
import java.sql.Timestamp;
import ec.util.MersenneTwisterFast;
import java.util.ArrayList;
import java.util.Collections;
import java.util.Comparator;
import java.util.List;
import java.util.Map;
import java.util.*;
import java.util.Map;
import java.util.*;

public class DeterrenceDiffusionDemoWar extends SimState //GUI
{

    //private static long state = 0xCAFEBABE; // initial non-zero value

    public Continuous2D field = null;
    public static Network providers = null;

    public static MersenneTwisterFast random = new MersenneTwisterFast();

    public static final double XMIN = 0;
    public static final double XMAX = 1200;
    public static final double YMIN = 0;
    public static final double VMAX = 900;

    public static final double DIAMETER = 0;

    public static int TASK_ID=0;
    public static boolean Sequential=false; /* Sequential Audit ON/OFF*/
    // public static boolean FINet = false; /*False: Random True:
    Power Law*/

    public static int DiffuseFlag = 0;
    public static int Klevel = 3;
    public static int MaxRounds = 16;
    public static int MaxAuditRounds = 1;
    public static int MaxRuns = 10000;

    "-1-"
```
package sim.app.VirusWarranty;
import sim.field.network.ks;
import sim.field.continuous ks;
import sim.engine ks;
import sim.util ks;

/* Read from csv file */
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.util.StringTokenizer;
import java.io.ks;
import java.sql.Timestamp;

import cc.util.MersenneTwisterFast;
import java.util.ArrayList ks;
import java.util.Collections ks;
import java.util.Comparator ks;
import java.util.List ks;
import java.util.ks;
import java.util.util ks;

public class DeterrenceDiffusionDemoWar extends SimState //GUI
{

    //private static long state = 0xCAFEBABE; // initial non-zero value

    public Continuous2D field = null;
    public Network providers = null;

    public static MersenneTwisterFast random = now MersenneTwisterFast();

    public static final double XMIN = 0;
    public static final double XMAX = 1200;
    public static final double YMIN = 0;
    public static final double YMAX = 900;

    public static final double DIAMETER = 8;

    public static int TASK_ID = 0;
    public static boolean Sequential = false; /* Sequential Audit ON/OFF*/
    //    public static boolean FLNet = false; /*False: Random True:
Power Lau*/
    public static int DiffuseFlag = 0;
    public static int klevel = 3;
    public static int MaxRounds = 16;
    public static int MaxAuditRounds = 1;
    public static int MaxRuns = 10000;

    */
public static int INFThreshold = 10;
public static int maxDegree = 7;  //Per Dealer
public static int minDegree=1;   //Per Dealer
public static double RandConnProb = 10;

/**< 300 providers in the network*/
public static int numProviders = 1000;
public static double maxIncome = 700;    //To be set up in CreateProviders
public static double meanIncome = 500;
public static double stdDev = 100;

/**<Weight representing the effectiveness of provider fraud prob algorithm*/
public static double weight = 0.8;

/**< Set Diffusion probabilities */
/**<By default, the provider has not been audited (Fa, Fr, Fc, Plintr don’t apply)
public static double Fc = 0.9;  //Intra-Clique
public static double Plintr = 0.2;  // Inter-Clique Diffusion Probability
public static double DfDecay = 0.5;  //High vs Low Decay

/**< Set Deviation Cut-off */
public static double DevCutOff = 1.5;

/**< Set Deterrence vars */
public static double Pd = 0.8;
public static double alpha1 = 0.95;  // Deterrence: In a clique of non-fraudulent dealers
public static double alpha2 = 0.7;   // Deterrence: In a clique of fraudulent dealers

public static List<String> FastSets = new ArrayList<String>(); //Optimality

public Continuous2D environment = null;
public final static double collisionDistance = 5.0;

/**< Creates a VirusInfectionDemo simulation with the given random number seed. */
public DeterrenceDiffusionDemoWar(long seed)
{
   super(seed);
}

public void start()
{
   super.start(); // clear out the schedule  GUI

   field = new Continuous2D(16.0, (XMAX-XMIN), (YMAX-YMIN));
   providers = new Network(); //new Network(false);
}
// Create Dealer Network
CreateDealerNetwork(numProviders, maxDegree, minDegree);

// Set Fraudulent Nodes
SetFraudNodes();
SetInfluentialNodes();

try {
    String outFilePath = "outStart." + TASK_ID;
    PrintWriter out = new PrintWriter(new FileWriter(outFilePath, true));
    // PrintWriter out = new PrintWriter(new OutputStreamWriter(System.out), true);

    // Set k-Level Neighbors
    Bag ProvObjs = providers.getAllNodes();
    for (int p = 0; p < ProvObjs.numObjs; p++)
    {
        Provider prov = ((Provider) ProvObjs.objs[p]);

        // Find the k-Level Neighbors
        List< List<Provider> > lvlNeighbors = KlvlNeighbors(prov, kLevel);

        // Set the estimated fraud value (deviation)
        CalcDevFraudVal(prov);

        //out.println("Prov Index \t" + prov.getIndex() + "");
        int lvlDegree = 0;
        for (int Level = 0; Level < kLevel; Level++)
        {
            List<Integer> NeighborsIndices = new ArrayList<Integer>();
            NeighborsIndices = prov.getKlvlNeighborsIndices(Lv);
            // For (int i = 0; i < prov.getKlvlDegree(Lv); i++)
            //     out.println("Level \t" + Level + ":\"Neibor \t" + i + ":\t"
            NeighborsIndices.get(i));
            klvlDegree += prov.getKlvlDegree(Lv);
        }
        prov.setKlvlDegree(klvlDegree);
    }

    out.close();
}
catch (IOException e){
    System.err.println("Caught Exception: " + e); // e.printStackTrace();
}

for (int Round = 0; Round < MaxRounds; Round++)
{
}
TASK_ID=Round;
ReadInputFile();

// Network Topology
DisplayNetwork();

System.out.println("DiffuseFlag: " + DiffuseFlag);

int K_AUDIT = 3;
for (int AuditRound = 0; AuditRound < MaxAuditRounds; AuditRound++)
{
    double[][] IncChange = new double[2][5];
    double[] NetIncChange = new double[2];
    double[] FraudAuditIncChange = new double[2];
    double[] FraudAwareIncChange = new double[2];
    double[] NonFraudAuditIncChange = new double[2];
    double[] NonFraudAwareIncChange = new double[2];

    double[] numFraud = new double[2];
    double[] numAware = new double[2];
    double[] numAwareFraudCluster = new double[2];
    double[] numAwareNonFraudCluster = new double[2];

    double[] TotalPrecision = new double[2];
    double[] TotalRecall = new double[2];
    int[] numWINS = new int[2];

    List<Provider> DevAuditSet = new ArrayList<Provider>();
    List<Provider> DetAuditSet = new ArrayList<Provider>();
    List<Provider> OptAuditSet = new ArrayList<Provider>();

    DevAuditSet = SelectAuditProviders(K_AUDIT, 0);
    clearInfo();
    DetAuditSet = SelectAuditProviders(K_AUDIT, 1);

    String DevSet = null;
    String DetSet = null;
    for (int i=0; i<DevAuditSet.size(); i++)
    {
        DevSet += Integer.toString(DevAuditSet.get(i).getIndex());
        DetSet += Integer.toString(DetAuditSet.get(i).getIndex());
    }

    OptAuditSet = FindOptimalSet();

    // Continue with the rest of the code...
}
for (int 1 = 0; 1 < MaxRuns; 1++)
{
  for (int Mode=0; Mode<4; Mode++) // 0: Deviation 1: Deterrence 2: Degree
  {

if (Mode == 0)
{
  clearInfo();
  DiffuseDeter(DevAuditSet, xLevel);
  /*
  System.out.println("Deviation");
  System.out.println("Audited EPR Index \t Fraudulent \t Actual Degree \t K-Level Degree \t Inc Markup \t Inc Inc \t Pnal Inc \t Fraud Prob \t Estimated Fraud Val \t Risk Tolerance \t Net Fraud Val");
  for(int i=0; i<DevAuditSet.size(); i++)
  {
    Provider prox =
      (Provider)DevAuditSet.get(i);
    System.out.println(prox.getEPRIndex() + " \t " + prox.FraudProb + " \t " + prox.getDegree() + " \t " + prox.getkLevelDegree() + " \t " + prox.getIncMarkup() + " \t " + prox.getIncInc() + " \t " + prox.getPnalInc() + " \t " + prox.getRiskTolerance() + " \t " + prox.getNetFraudValue());
  }
  */
}

if (Mode == 2)
{
  clearInfo();
  DiffuseDeter(SetAuditSet, xLevel);
  /*
  System.out.println("Deterrence");
  System.out.println("Audited EPR Index \t Fraudulent \t Actual Degree \t K-Level Degree \t Inc Markup \t Inc Inc \t Pnal Inc \t Fraud Prob \t Estimated Fraud Val \t Risk Tolerance \t Net Fraud Val");
  for(int i=0; i<SetAuditSet.size(); i++)
  {
    Provider prox =
      (Provider)SetAuditSet.get(i);
    System.out.println(prox.getEPRIndex() + " \t " + prox.FraudProb + " \t " + prox.getDegree() + " \t " + prox.getkLevelDegree() + " \t " + prox.getIncMarkup() + " \t " + prox.getIncInc() + " \t " + prox.getPnalInc() + " \t " + prox.getRiskTolerance() + " \t " + prox.getNetFraudValue());
  }
  */
}
IncChange[Mode] = calcNetIncChange();
NetIncChange[Mode] += IncChange[Mode][0];
FraudAuditIncChange[Mode] += IncChange[Mode][1];
FraudAwareIncChange[Mode] += IncChange[Mode][2];
NonFraudAuditIncChange[Mode] += IncChange[Mode][3];
NonFraudAwareIncChange[Mode] += IncChange[Mode][4];
FraudClusterIncChange[Mode] += IncChange[Mode][5];
NonFraudClusterIncChange[Mode] += IncChange[Mode][6];
TotalPrecision[Mode] = calcPrecision(K_AUDIT);
TotalRecall[Mode] = calcRecall();

Bag ProvObj = providers.getAllNodes();
for(int p=0; p<ProvObj.numObjs; p++)
{
  Provider prov = ((Provider)(ProvObj.objs[p]));

  if (prov.fraud == true)
    numFraud[Mode]++;

  if (prov.aware == true)
    numAware[Mode]++;

  if ((prov.aware == true) && (prov.fraudCluster == true))
    numAwareFraudCluster[Mode]++;

  if ((prov.aware == true) && (prov.fraudCluster == false))
    numAwareNonFraudCluster[Mode]++;

  //

  if (IncChange[1][0] < IncChange[0][0])
    numInc[0]++;
  else if (IncChange[1][0] > IncChange[0][0])
    numInc[1]++;
}
double Difference = IncChange[1][0] - IncChange[0][0];
// System.out.println("Deviation Amount \t + IncChange [0][0] + "
Deviation Amount \t + IncChange [1][0] + "+ Difference);
}

int Mode = -3;
OutputResults(K_AUDIT, Mode,NetIncChange[Mode]/MaxRuns, 
FraudAuditIncChange[Mode]/MaxRuns, FraudAwareIncChange[Mode]/MaxRuns, 
NonFraudAuditIncChange[Mode]/MaxRuns, NonFraudAwareIncChange[Mode]/ 
MaxRuns, FraudClusterIncChange[Mode]/MaxRuns, NonFraudClusterIncChange[ 
Mode]/MaxRuns, numFraud[Mode]/MaxRuns, numAware[Mode]/MaxRuns, 
numAwareFraudCluster[Mode]/MaxRuns, numAwareNonFraudCluster[Mode]/ 
MaxRuns, DevAuditSet, TotalPrecision[Mode]/MaxRuns, TotalRecall[Mode]/ 
MaxRuns, numWin[Mode], DevSet);
Model = 1;
OutputResults(K_AUDIT, Mode,NetIncChange[Mode]/MaxRuns, 
FraudAuditIncChange[Mode]/MaxRuns, FraudAwareIncChange[Mode]/MaxRuns, 
NonFraudAuditIncChange[Mode]/MaxRuns, NonFraudAwareIncChange[Mode]/ 
MaxRuns, FraudClusterIncChange[Mode]/MaxRuns, NonFraudClusterIncChange[ 
Mode]/MaxRuns, numFraud[Mode]/MaxRuns, numAware[Mode]/MaxRuns, 
numAwareFraudCluster[Mode]/MaxRuns, numAwareNonFraudCluster[Mode]/ 
MaxRuns, DevAuditSet, TotalPrecision[Mode]/MaxRuns, TotalRecall[Mode]/ 
MaxRuns, numWin[Mode], DevSet);

clearInf();
K_AUDIT = 3;
}

}

public static List<Provider> FindOptimalSet()
{
    List<Provider> OptAuditSet = new ArrayList<Provider>();
    String set = new String();

    int K_AUDIT = 3; // if this number is changed, need to change the way 
    OptAuditSet is constructed

    int OptSetsNum = 0;
    for (int k=0;k<NumProviders;k++)
    {
        for (int p=0;p<NumProviders;p++)
        {
            
            .7.

0

138
for (int q=j; q<numberProviders; q++)
{
    set = "xx" + Integer.toString(k) + "xx" + Integer.toString(p) + "xx" +
    Integer.toString(q) + "xx";

    if (x==p || x==q || p==q)
        continue;

    OptAuditSet = new ArrayList<Provider>();
    OptAuditSet.add(FindNode(k));
    OptAuditSet.add(FindNode(p));
    OptAuditSet.add(FindNode(q));

    double[] IncChange = new double[5];
    double netIncChange = 0;
    double FraudAuditIncChange = 0;
    double FraudAwareIncChange = 0;
    double NonFraudAuditIncChange = 0;
    double NonFraudAwareIncChange = 0;
    double FraudClusterIncChange = 0;
    double NonFraudClusterIncChange = 0;

    double numFraud = 0;
    double numAware = 0;
    double numAwareFraudCluster = 0;
    double numAwareNonFraudCluster = 0;

    double totalPrecision = 0;
    double totalRecall = 0;
    for (int i = 0; i < MaxRuns; i++)
    {
        clearInfo();
        DiffuseDeter(OptAuditSet, kLevel);

        IncChange = calcNetIncChange();
        netIncChange += IncChange[0];
        FraudAuditIncChange += IncChange[1];
        FraudAwareIncChange += IncChange[2];
        NonFraudAuditIncChange += IncChange[3];
        NonFraudAwareIncChange += IncChange[4];
        FraudClusterIncChange += IncChange[5];
        NonFraudClusterIncChange += IncChange[6];
        totalPrecision += calcPrecision(K_AUDIT);
        totalRecall += calcRecall();

        Bag ProvGuys = providers.getAllNodes();

}
for(int j=0;j<ProvObj.numObj;j++)
{
    Provider prov = ((Provider)(ProvObj.objs[j]));

    if (prov.Fraud == true)
        numFraud++;

    if (prov.Aware == true)
        numAware++;

    if ((prov.Aware == true) && (prov.FraudCluster == true))
        numAwareFraudCluster++;

    if ((prov.Aware == true) && (prov.FraudCluster == false))
        numAwareNonFraudCluster++;  
}

set = Integer.toString(k) + Integer.toString(p) + Integer.toString(q);

OutputResults(F_AUDIT, k,NetIncChange/MaxRuns , FraudAuditIncChange/MaxRuns , FraudAwareIncChange/MaxRuns , NonFraudAuditIncChange/MaxRuns , NonFraudAwareIncChange/MaxRuns, FraudClusterIncChange/MaxRuns, NonFraudClusterIncChange/MaxRuns, numFraud/MaxRuns, numAware/MaxRuns, numAwareFraudCluster/MaxRuns, numAwareNonFraudCluster/MaxRuns, OptAuditSet, TotalPrecision/MaxRuns, TotalRecall/MaxRuns, 0, set);

clearInfo();
String InputFile = "input." + TASK_ID;

BufferedReader br = new BufferedReader(new FileReader(InputFile));
String line = "";
String[] linearr = null;
StringTokenizer st = null;

int lineNumber = 0;

//read file line by line
while ((line = br.readLine()) != null)
{
    linearr = line.split("=");

    if (linearr[0].equals("KLevel"))
        KLevel = Integer.parseInt(linearr[1]);
    if (linearr[0].equals("MaxRounds"))
        MaxRounds = Integer.parseInt(linearr[1]);
    if (linearr[0].equals("MaxRun"))
        MaxRun = Integer.parseInt(linearr[1]);
    if (linearr[0].equals("numProviders"))
        numProviders = Integer.parseInt(linearr[1]);
    if (linearr[0].equals("maxincome"))
        maxIncome = Integer.parseInt(linearr[1]);
    if (linearr[0].equals("meanIncome"))
        meanIncome = Integer.parseInt(linearr[1]);
    if (linearr[0].equals("stdDev"))
        stdDev = Integer.parseInt(linearr[1]);
    if (linearr[0].equals("weight") + Weight representing the effectiveness of provider fraud prob algorithm)
        weight = Double.parseDouble(linearr[1]);
    if (linearr[0].equals("Pp"))
        Pp = Double.parseDouble(linearr[1]);
    if (linearr[0].equals("PInter") + PInter)
        PInter = Double.parseDouble(linearr[1]);
    if (linearr[0].equals("DifDecay"))
        DifDecay = Double.parseDouble(linearr[1]);
    if (linearr[0].equals("DevCutOff") + DevCutOff)
        DevCutOff = Double.parseDouble(linearr[1]);
    if (linearr[0].equals("Pd") + Pd)
        Pd = Double.parseDouble(linearr[1]);
    if (linearr[0].equals("alpha") + alpha)
        alpha = Double.parseDouble(linearr[1]);
    if (linearr[0].equals("alpha2") + alpha2)
        alpha2 = Double.parseDouble(linearr[1]);
}

br.close();
catch (Exception e) {
    System.out.println("Input file cannot be read : " + e);
}

public static void clearInfo()
{
    Bag ProvOobjs = providers.getAllNodes();
    for(int p=0; p<ProvOobjs.size(); p++)
    {
        Provider prov = (Provider)(ProvOobjs.objs[p]);
        prov.Audit = false;
        prov.Aware = false;
        prov.E2ware = false;
        prov.Diffused = false;
        prov.setSegMarkUp(prov.getInitSegMarkUp());
        prov.setSegInc(prov.getInitSegInc());
        prov.setFinalInc(prov.getInitInc());
        prov.setSegDev(prov.getInitSegDev());
        prov.setFinalDev(prov.getInitDev());
    }
}

displayNetwork()
{
    try
    {
        System.out.println("level=" + xLevel);
        System.out.println("MaxRounds=" + MaxRounds);
        System.out.println("MaxRuns=" + MaxRuns);
        System.out.println("numProviders=" + numProviders);
        System.out.println("maxIncome=" + maxIncome);
        System.out.println("meanIncome=" + meanIncome);
        System.out.println("stdDev=" + stdDev);
        System.out.println("weight=" + weight);
        System.out.println("Pa=" + Pa);
        System.out.println("PInter=" + PInter);
        System.out.println("DifDecay=" + DifDecay);
        System.out.println("DevCutOff=" + DevCutOff);
        System.out.println("Pd=" + Pd);
        System.out.println("alpha1=" + alpha1);
        System.out.println("alpha2=" + alpha2);
    }
}
String outFile = "outNetwork." + TASK_ID;
PrintWriter out = new PrintWriter(new FileWriter(outFile, true));

//PrintWriter out = new PrintWriter(new OutputStreamWriter(System.out), true);

Bag ProvObjs = providers.getAllNodes();

out.println ("Prov Index\tCluster Num\tFraud Prob\tFraud Status\n\tFraud Cluster\tDegree\tInit Income\tRisk Tolerance\tMulti level degree ");
for(int i=0;i<ProvObjs.getNumObjs();i++)
{
    Provider prov = ((Provider)(ProvObjs.objs[i]));
    
    int TotalNumNeighbors = 0;
    for(int Level = 0; Level < Level; Level++)
    
    TotalNumNeighbors += prov.getLevelDegree(Level);
    
    out.println (prov.getIndex() + "\t" + prov.ClusterNum + "\t" + prov.getFraudProb() + "\t" + prov.Fraud + "\t" + prov.FraudCluster + "\t" + prov.getDegree() + "\t" + prov.getInitInc() + "\t" + prov.getRiskTolerance() + "\t" + TotalNumNeighbors );
    
    for(int i=0;i<prov.getDegree();i++)
    
    out.println ( "\t" + prov.NeighborsIndices.get(i));
}
//out.println ( StatisticsUtil.getMean(NetCentralities) + "\n");
//out.println ( StatisticsUtil.getStandardDeviation(NetCentralities) + "\t");
out.close();

} catch (IOException e){
    e.printStackTrace();
}

public static void OutputResults(int K_AUDIT, int Mode, double NetIncChange, double FraudAudITIncChange, double FraudAwareIncChange, double NonFraudAudITIncChange, double FraudClusterIncChange, double NonFraudClusterIncChange, double NumFraud, double NumAware, double numAwareFraudCluster, double numAwareNonFraudCluster, List <Provider> AuditedSet, double Precision, double Recall, int NumWins, String set )
{
}
```java
double[] NetDegrees = new double[numProviders];
List<Integer> audited = new ArrayList<Integer>();

StringBuffer strBuf = new StringBuffer();
strBuf.append("Mode \t Sequential \t Num Fraudulent \t Num Audited \t numAware \t AvgNetIncChange \t AvgFraudAuditIncChange \t AvgNonFraudAwareIncChange \t AvgNonFraudAwareIncChange \t FraudClusterIncChange \t NonFraudClusterIncChange\t numAwareFraudCluster \t numAwareNonFraudCluster \t Precision \t Recall \t Number of Wins \t Audited Set \n" );

if (Mode == 0)
    strBuf.append("Deviation \t \n");
else if (Mode == 1)
    strBuf.append("Deterrence \t \n");
else if (Mode == 10)
    strBuf.append("Brute Force \t \n");
strBuf.append(Sequential + "\t");
strBuf.append(numFraud + "\t");
strBuf.append(AuditedSet.size() + "\t");
strBuf.append(numAware + "\t");

strBuf.append( NetIncChange + "\t");
strBuf.append( FraudAuditIncChange + "\t");
strBuf.append( FraudAwareIncChange + "\t");
strBuf.append( NonFraudAuditIncChange + "\t");
strBuf.append( NonFraudAwareIncChange + "\t");
strBuf.append( FraudClusterIncChange + "\t");
strBuf.append( NonFraudClusterIncChange + "\t");
strBuf.append( numAwareFraudCluster + "\t");
strBuf.append( numAwareNonFraudCluster + "\t");
strBuf.append( Precision + "\t");
strBuf.append( Recall + "\t");
strBuf.append( NumWins + "\t");
strBuf.append( set + "\t\n");
strBuf.append("Audited Providers \t Detection \t Deterrence\n");
strBuf.append(AuditedSet.size() + "\t");
if (Mode == 0)
    strBuf.append(NetIncChange + "\n");
else if (Mode == 1)
    strBuf.append("\t + NetIncChange + "\n");
```
strBuf.append("Audited Prov Index \t Fraudulent \t Degree \t K-Level Degree \t Init Markup \t Init Inc \t Final Inc \t Fraud Prob \t Estimated Dev Fraud Val \t Risk Tolerance \t Net Fraud Val (THIS VALUE CHANGES AT EVERY ITERATION AFTER EACH AUDIT PROV SELECTION!!!! (EMWARE FLAGS) \t Fraud Cluster \n");

for(int j=0; j<AuditedSet.size(); j++)
{
    Provider prov = (Provider)AuditedSet.get(j);
    strBuf.append(prov.getIncDec() + "\t" + prov.getFraud + "\t" + prov.getDegree() + "\t" + prov.getk1Degree() + "\t" + prov.getInitMarkUp() + "\t" + prov.getInitInc() + "\t" + prov.getFinalInc() + "\t" + prov.getFraudProb() + "\t" + prov.getDevFraudValue() + "\t" + prov.getRiskTolerance() + "\t" + prov.getNetFraudValue() + "\t" + prov.getFraudCluster();
}
System.out.println(strBuf);

public Provider CreateNode(int id, double x, double y)
{
    ZipfGenerator zipfGen = new ZipfGenerator(100,1);
    double income = (random.nextGaussian() * stdDev) + meanIncome;
    Provider prov = new Provider(id);
    double FraudProb = 0;
    FraudProb = (double) zipfGen.next() / 100;
    prov.setFraudProb(FraudProb);
    for(int z=0; z<5; z++)
    {
        prov.HospPriv[z] = random.nextInt(S0);
    }
    prov.setRiskTolerance(random.nextDouble() * 0.5);
    prov.setInitInc(income);
    prov.setSeqInc(prov.getInitInc());
}
public static void SetFraudNodes()
{
    // Shuffle Fraud probabilities
    for(int i = 0; i < 2000; i++)
    {
        Provider prov1 = ((Provider)(ProvObjs.objs[random.nextInt(numProviders)]));
        Provider prov2 = ((Provider)(ProvObjs.objs[random.nextInt(numProviders)]));

        // Swap double temp = prov1.getFraudProb();
        prov2.setFraudProb(prov1.getFraudProb());
        prov1.setFraudProb(temp);
    }

    // Set provider as fraudulent based on prior fraud prob + shock
    for(int i = 0; i < ProvObjs.size(); i++)
    {
        Provider prov = ((Provider)(ProvObjs.objs[i]));
        double d = random.nextDouble();
        if (d < prov.getFraudProb())
            prov.Fraud = true;
        else
            prov.Fraud = false;

        // Shock the prior fraud probabilities
    }
}
double RandProb = random.nextDouble();
double ShockedProb = (weight * prov.getFraudProb()) + ((1-weight) *
                     RandProb);

prov.setFraudProb(ShockEdProb);
}

// Set Fraudulent Nodes Attributes
for(int i = 0; i < ProvObjs.size(); i++)
{
    Provider prov = ((Provider)(ProvObjs.objs[i]));
    if (prov.Fraud == true)
    {
        double income = prov.getInitInc();
        prov.setInitInc(random.nextDouble() * (maxIncome-income) +
                        income); // Increase income
        prov.setSegMarkUp(prov.getInitInc() - income);
        prov.setRiskTol(random.nextDouble() * (0.5) + 0.5);

        // Set Fraudulent Cluster Attributes
        prov.FraudCluster = true;
        for (int j=0; j<prov.NeighborsIndices.size(); j++)
        {
            int index = prov.NeighborsIndices.get(j);
            Provider neighbor = ((Provider)(ProvObjs.objs[index]));

            if ( (prov.ClusterNum > -1)
                && (prov.ClusterNum == neighbor.ClusterNum) )
            {
                neighbor.FraudCluster = true;
            }
        }
    }
}

public static void SetInfluentialNodes ()
{
    Bag ProvObjs = providers.getAllNodes();
    int Flag = 0;
    -16-
```java
int i = 0;
// Set Influential Nodes attributes
for (int i = 0; i < ProvObj.size(); i++)
{
    Provider prov = (Provider) ProvObj.obj[i];
    if (prov.Degree > InfThreshold)
    {
        ++i;
        prov.Influential = true;
    }
}

public void CreateDealerNetwork(int numProv, int maxDegree, int minDegree) //Cliques
{
    try
    {
        String outFileName = "outCreateDealerNetwork.0";
        PrintWriter out = new PrintWriter(new FileWriter(outFileName, true));
        out.println("in CreateDealerNetwork");
        Esa ProvObj = providers.getAllNodes();
        System.out.println("numProv: " + numProv + " ProvObj.size: " + ProvObj.size());
        int NodesNum = 0;
        int index = 0;
        int p = 0;
        int q = 0;
        int ClusterNum = 0;
        while (NodesNum < numProv)
        {
            double xa = random.nextDouble() * (XMAX - XMIN - DIAMETER) + XMIN + DIAMETER / 2 - XMIN;
            double ya = random.nextDouble() * (YMAX - YMIN - DIAMETER) + YMIN + DIAMETER / 2 - YMIN;
            Provider providerA = CreateNode(index, xa, ya);
            // Re-initialize Neighbors
            providerA.Degree = 0;
            providerA.NeighborsIndices = new ArrayList<Integer>();
            providerA.klvDegree = 0;
            NodesNum = NodesNum + 1;
            ++index;
        }
    }
    catch (IOException e)
    {
        e.printStackTrace();
    }
}
```
```java
providerA.lv1Degree = new int[1000];
providerA.lv1NeighborsIndices = new ArrayList<List<Integer>>();

int randDegree = random.nextInt(maxDegree - minDegree) + minDegree;

if ((NodesNum + randDegree + 1) > numProv)
    {
        cut.println("RandDegree \t" + randDegree + "\t (NodesNum + randDegree + 1) \t" + (NodesNum + randDegree + 1));
        randDegree = numProv - NodesNum - 1;
    }

cut.println("prov A index \t" + index + "\t Random out randDegree \t" + randDegree);
providerA.setDegree(randDegree);

providerA.ClusterNum = ClusterNum;

double xB = xA+(1.0/p)*(-providerA.getDegree());
double yB = yA+((-1.0/p)*Math.pow(-1, p%2))/10;

//Make the cluster
for (p = index+1; p < index+ providerA.getDegree()+1; p++)
    {
        xB = xB+(2.0*(p-1)%2);
        yB = yB+(2.0*(p%2));
        Provider providerB = CreateNode( p, xB, yB);

        providers.addEdge(providerA, providerB, null);
        providerA.NeighborsIndices.add(providerB.getIndex());
        providerB.NeighborsIndices.add(providerA.getIndex());

        providerA.ClusterNum = ClusterNum;
        providerB.ClusterNum = ClusterNum;

        for(q = p-1; q > index-1; q--)
            {
                Provider providerC = ((Provider)(ProvObjs.objs[q]));

                providers.addEdge(providerB, providerC, null);
                if (!providerB.NeighborsIndices.contains(providerC.getIndex()))
                    {
                        providerB.NeighborsIndices.add(providerC.getIndex());
                        providerB.setDegree (providerB.NeighborsIndices.size());
                    }
                }
        }
{  
  providerC.NeighborsIndices.add(providerB.indexOf());
  providerC.setDegree(providerC.NeighborsIndices.size());
}

providerB.setDegree(providerB.NeighborsIndices.size());

ProvObj.objs[p] = providerB;
out.println("prox B index \t" + p + \" \n providerB.NeighborsIndices.size() \t" + providerB.NeighborsIndices.size());
out.println("prox B index \t" + p + \" \t providerB.ClusterNum \t" + providerB.ClusterNum);

providerA.setDegree(providerA.NeighborsIndices.size());
ProvObj.objs[index] = providerA;
out.println("prox A index \t" + index + \" \t providerA.ClusterNum \t" + providerA.ClusterNum);
index += providerA.getDegree() + 1;
NodeNum += (randDegree + 1);
ClusterNum++;
}

for(int j = 0; j < numProv; j++)
{
  Provider providerB = ((Provider)ProvObj.objs[j]);
  out.println("prox B index \t" + j + \" \t providerB.ClusterNum \t" + providerB.ClusterNum);
}

List<Integer> generated = new ArrayList<Integer>();
List<Double> generatedCluster = new ArrayList<Double>();
int RandConnNum = 0;

for(int j = 0; j < numProv; j++)
{
  Provider providerD = ((Provider)ProvObj.objs[j]);
  int k = random.nextInt(numProv-1);

  Provider providerE = ((Provider)ProvObj.objs[k]);
  if (providerD.ClusterNum != providerE.ClusterNum && generatedCluster.contains(providerD.ClusterNum) && !generatedCluster.contains(k))
  {
    generatedCluster.add(providerD.ClusterNum);
    generated.add(k);
    RandConnNum++;
  }
}
```java
&6 { !generated.contains(k) }
&6 { !generated.contains(s) }

    providers.addEdge(providerD, providerE, null);
    providerDNeighborsIndices.add(providerD.getIndex());
    providerENeighborsIndices.add(providerD.getIndex());
    providerD.setDegree(providerDNeighborsIndices.size());
    providerE.setDegree(providerENeighborsIndices.size());

    out.println("prox D index \t + providerD.getIndex() + " \t providerDNeighborsIndices.size() \t + providerDNeighborsIndices.size());
    out.println("prox E index \t + providerE.getIndex() + " \t providerENeighborsIndices.size() \t + providerENeighborsIndices.size());

    generated.add(s);
    generated.add(c);

    generatedCluster.add(providerD.ClusterNum);

}

out.println("out of CreateDealerNetwork");
out.close();

} catch (IOException e){
    e.printStackTrace();
}

public static List<Integer> randomPick(Provider providerA, int startNumber, int endNumber, int numbersNeeded, int maxDegree)
{
    Random rng = new Random();
    List<Integer> generated = new ArrayList<Integer>();

    Bag ProvObjs = providers.getAllNodes();
    Bag SortedObjs = new Bag (SortNodes()); //Sort by degree

    int Flag = 0;
    int Max = 0;
```
```java
int t = 0;
while (t < numbersNeeded)
{
    for (int i = startNumber; i < endNumber; i++)
    {
        int next = random.nextInt(endNumber - startNumber) + startNumber;

        Provider providerB = ((Provider) ProvObj.obj[SortedObj.obj[next]].getIndex());

        if (Flag == 0)
            Max = providerB.getDegree();
        else if (Flag == 1)
            Max = maxDegree;

        if (!generated.contains(next))
            if (providerA.NeighborsIndices.contains(next))
                if (providerB.NeighborsIndices.size() > 0) //Dealer Network
                    if (providerA.getIndex() == providerB.getIndex())
                        generated.add(next);

        t++;
        if (t == numbersNeeded)
            break;
    }
    Flag = 1;
}

return generated;
}

public static List<Integer> randomPick(List<Integer> PrevNeighbors, int startNumber, int endNumber, int numbersNeeded)
{
    Random rng = new Random();
    List<Integer> generated = new ArrayList<Integer>();

    for (int t = 0; t < numbersNeeded; t++)
    {
        while (true)
        {
            int next = rng.nextInt(endNumber - startNumber);

            if (!generated.contains(next) && (!PrevNeighbors.contains(next)))
```
public static List<Provider> SelectAuditProviders(int numAudited, int mode) {

    List<Provider> audit_set = new ArrayList<Provider>();
    List<Provider> provobj = providers.getAllNodes();
    try {
        String outfile = "outAudit." + TASK_ID;
        PrintWriter out = new PrintWriter(new FileWriter(outfile, true));

        if (mode == 0) // Deviation * Prior Probability of fraud
        {

            // Sort Nodes based on Mode
            List<Provider> sortedobj = new ArrayList<Provider>();
            int q = 0;
            while (q < numAudited)
            {
                // FOR (int i = 0; i < sortedobj.numObj; i++)
                for (int i = 0; i < numProviders; i++)
                {
                    Provider prov = ((Provider) provobj.get(i));
                    audit_set.add(prov);
                    Audit prc;
                    prc = prov;$
                    if (q == numAudited)
                    {
                        break;
                    }
                    q++;
                }
            }
        }
    }
}
else if (Mode == 1)
   {
      while (numAudited > 0)
      {

         // Calculate the Net Fraud value after setting some EAware flags
         for(int p=0;p<numProviders;p++)
         {
            Provider provider = ((Provider)(ProvObj.objs[p]));
            if (DiffuseFlag -- 0)
               CalcNetFraudValue(provider);
            else
               CalcNetFraudValue2(provider);
            // out.println("prow \t" + provider.getIndex() + "\nNetFraudVal\t" + provider.getNetFraudValue() + "\n");
         }

         // Sort nodes based on Mode.
         Bag SortedObj = SortNodes(Mode);

         for(int i=0;i<numProviders;i++)
         {
            int provIndex = ((Provider)(SortedObj.objs[i])).getIndex();
            Provider prov = ((Provider)(ProvObj.objs[provIndex]));
            System.out.println("To be Audited Provider Index \t" + provIndex + "\nprov.Audit\t" + prov.Audit + "\t Net Fraud Value\t" + prov.getNetFraudValue() + "\n");

            if ( prov.Audit == true)
               continue;
            prov.EAware = true;

            for(int q=0;q<prov.getDegree();q++)
            {
               Provider Neighbor = ((Provider)(ProvObj.objs[(prov.
               NeighborIndices.get(q))]));
               Neighbor.EAware = true;
            }
         }

         AUDIT_SET.add(prov);
         Audit(prov);
         numAudited --;
   }
break;
}

/* else if (Mode == 2) //Brute Force */
{

Bag provObjs = providers.getAllNodes();

boolean found = false;
while (found == false)
{
    AUDIT_SET = new ArrayList<Provider>();
    String set = new String();
    int IntSet = 0;
    int q=0;
    while (q<numAudited)
    {
        //for(int i=0;i<SortedObjs.numObjs;i++)
        for(int i=0;i<numProviders;i++)
        {
            Provider prov1 = ((Provider) provObjs.objs[i]);
            if (prov1.Audit == false)
            {
                AUDIT_SET.add(prov1);
                set += Integer.toString(prov1.getIndex());
                Audit(prov1);
                q++;
                if (q==numAudited)
                    break;
            }
            if (q==numAudited)
                break;
        }
        for(int i=i+1;i<numProviders;i++)
        {
            Provider prov2 = ((Provider) provObjs.objs[i]);
            if (prov2.Audit == false)
            {
                AUDIT_SET.add(prov2);
                set += Integer.toString(prov2.getIndex());
                Audit(prov2);
                q++;
                if (q==numAudited)
                    break;
            }
            if (q==numAudited)
break;
    for (int k=j+1;k<numProviders;k++)
    {
        Provider prov3 = ((Provider)provObjs.objs[k]);
        //cout.println(prov3.getObjId() + "\t" +
        prov3.getDevFraudValue());
        if (prov3.Audit == false)
        {
            AUDIT_SET.add(prov3);
            set += Integer.toString(prov3.getObjId());
            Audit(prov3);
            q++;
            if (q==numAudited)
                break;
        }
    }
    if (q==numAudited)
        break;

    }
    //IntSet = Integer.parseInt(set);
    if (IntSet.contains(set))
        found = true;
    }

    }/*
    cout.println("out of SelectAuditProviders\t" + new Timestamp(System.
    currentTimeMillis()));
    cout.close();
}
    catch (IOException e){
    e.printStackTrace();
}
    return AUDIT_SET ;
}

public static Provider FindNode (int index)
{
    Provider prov = new Provider();
    Bag provObjs = providers.getAllNodes();
    for (int p=0;p<provObjs.numObjs;p++)
    {
        prov = ((Provider)(provObjs.objs[p]));
        if (prov.getIndex() == index)
            break;

    }
public static List<Provider> IndicesToNodes (List<Integer> Indices, int ArrSize)
{
    List<Provider> Nodes = new ArrayList<Provider>();
    Bag provObjs = providers.getAllNodes();
    for (int i=0; i<ArrSize; i++)
    {
        Provider provider = ((Provider)provObjs.objs[[Indices.get(1)]]);
        Nodes.add(provider);
    }
    return Nodes;
}

public static List<Integer> NodesToIndices (int size, List<Provider> Neighbors)
{
    List<Integer> NodesIndices = new ArrayList<Integer>();
    Bag provObjs = providers.getAllNodes();
    for (int i=0; i<size; i++)
    {
        NodesIndices.add( ((Provider)Neighbors.get(i)).getIndex() );
    }
    return NodesIndices;
}

public static void Audit (Provider prov)
{
    prov.Audit = true;
    prov.Aware = true;
}

public static void CalcDevFraudVal (Provider prov)
{
    prov.setDevFraudValue( (prov.getInitInc() * prov.getFraudProb()) );
}

public static void DiffuseDeter(List<Provider> AuditedSet, int kLevel)
{
try {
    String outFile = "outDiffuseDeter." + TASK_ID;
    PrintWriter out = new PrintWriter(new FileWriter(outFile, true));
    // PrintWriter out = new PrintWriter(new OutputStreamWriter(System.out, true));

    Bag provObjs = providers.getAllNodes();
    for (int p = 0; p < provObjs.numObj; p++) {
        Provider prov = ((Provider) provObjs.obj[p]);
        prov.Aware = false;
        prov.EAware = false;
        prov.Diffused = false;
    }

    for (int j = 0; j < AuditedSet.size(); j++) {
        Provider prov = (Provider) AuditedSet.get(j);
        prov.Audit = true;
        kLevelDiffuse(prov, kLevel);
    }

    for (int p = 0; p < provObjs.numObj; p++) {
        Provider prov = ((Provider) provObjs.obj[p]);
        if (prov.Aware == true) {
            Deter (prov);
        }
    }

    out.println("Out of DiffuseDeter \t" + new Timestamp(System.currentTimeMillis()));
    out.close();
}

} catch (IOException e) {
    e.printStackTrace();
}

} public static void Deter (Provider prov) {
    double RealInc = 0;
    double PrevInc = 0;
if (prov.Fraud — true)
{
    if (prov.Audit — true)
        prov.setFinalInc(prov.getInitInc() — prov.getInitMarkUp());
    else // aware
        prov.setFinalInc(prov.getInitInc() — (prov.getRiskTolerance() * prov.getInitMarkUp()));

    double dl=random.nextDouble();
    if (dl<Fd)
        {
            if (prov.FraudCluster — true)
                prov.setFinalInc(prov.getFinalInc() * alpha2);
            else
                prov.setFinalInc(prov.getFinalInc() * alphal);
        }
    else if (prov.Fraud — false) // aware
        {
            double d=random.nextDouble();
            if (d<fd)
                {
                    if (prov.FraudCluster — true)
                        prov.setFinalInc(prov.getFinalInc() * alpha2);
                    else
                        prov.setFinalInc(prov.getFinalInc() * alphal);
                }
        }

    prov.setFinalDev((prov.getFinalInc() — meanIncome)/stdDev); // need to use the new mean
}

public static void diffuseFraud(Provider prov, int level)
{
    double IntraDiffProb = 0;
    double InterDiffProb = 0;

    InterDiffProb = Fd;
    IntraDiffProb = 0;

    prov.Aware = true;

    Bag ProvObjs = providers.getAllNodes();
```java
int Degree = prov.getDegree();
for (int q = 0; q < Degree; q++)
{
    Provider Neighbor = ((Provider)(ProvObjs.objs[prov.NeighborsIndices.get(q)]));
    int rand = random.nextInt(100);
    if (prov.ClusterNum > -1)
        if (prov.ClusterNum == Neighbor.ClusterNum)
            if (rand < IntraDifProb * Math.pow(DifDecay, Level) * 100))
                Neighbor.Aware = true;
    else if (rand < InterDifProb * Math.pow(DifDecay, Level) * 100)
        Neighbor.Aware = true;
}
prov.Difuses = true;

public static void DiffuseNonFraud(Provider prov, int Level)
{
    double IntraDifProb = IntraDifProb;
    double InterDifProb = InterDifProb;
    int rand = 0;
    prov.Aware = true;
    int Degree = prov.getDegree();
    Bag ProvObjs = providers.getAllNodes();
    for (int q = 0; q < Degree; q++)
    {
        Provider Neighbor = ((Provider)(ProvObjs.objs[prov.NeighborsIndices.get(q)]));
        int rand = random.nextInt(100);
        if (prov.ClusterNum > -1)
            if (prov.ClusterNum == Neighbor.ClusterNum)
                if (rand < IntraDifProb * Math.pow(DifDecay, Level) * 100))
```

---

160
{  
    Neighbor Aware = true;
}
else if (rand < InterDisProp * Math.pow(DiffDecay, Level) * 100)  
{  
    Neighbor Aware = true;
}
}
prov Diffused = true;


public static void kLevelDiffuse(Provider prov, int kLevel)  
{
    try
    {
        String outFile = "outLevelDiffuse." + TASK_ID;
        PrintWriter out = new PrintWriter(new FileWriter(outFile, true));
        prov Aware = true;
        Bag provObj = providers.getAllNodes();

        for (int DiffuseLevel = 0; DiffuseLevel < kLevel; DiffuseLevel++)
        {
            if (DiffuseLevel == 0)
            {
                int rand = random.nextInt(100);
                if (rand < prov.getFraudProb() * 100)
                    DiffuseFraud(prov, DiffuseLevel);
                else
                    DiffuseNonFraud(prov, DiffuseLevel);
                continue;
            }

            int NeighborLevel = DiffuseLevel - 1;
            List<Integer> NeighborsIndices = new ArrayList<Integer>();
            NeighborsIndices = prov.getLV1NeighborsIndices(NeighborLevel);
            for (int i = 0; i < prov.getLV1Degree(NeighborLevel); i++)
            {
                Provider provider = (Provider) provObj.objs[NeighborsIndices.get(i)];
                if (provider Aware == false || provider Diffused == true)
                    continue;
            }

        }
    }
    catch (Exception e)
    {
        e.printStackTrace();
    }

}


int rand = random.nextInt(100);
    if (rand < provider.getFraudProb() + 100)
        DiffuseFraud(provider, DiffuseLevel);
    else
        DiffuseNonFraud(provider, DiffuseLevel);
}
}

catch (IOException e)
    e.printStackTrace();
}

public static List<List<Provider>> kLvlNeighbors(Provider prov, int K)
{
    List<Provider> ProvSet = new ArrayList<Provider>();
    List<Provider> ProvNeighbors = new ArrayList<Provider>();
    List<Provider> Neighbors = new ArrayList<Provider>();
    List<List<Provider>> LvlNeighbors = new ArrayList<List<Provider>>();

    ProvSet.add(prov);

    for (int Level=0; Level<K; Level++)
    {
        Neighbors = new ArrayList<Provider>();
        for(int i=0; i<ProvSet.size(); i++)
        {
            Provider provider = (Provider)ProvSet.get(i);
            ProvNeighbors = IndicesToNodes(provider.NeighborsIndices, provider.
                getDegree());
            Neighbors.addAll(ProvNeighbors);
        }
        LvlNeighbors.add(Neighbors);
        prov.setLvlDegree(Level, Neighbors.size());
        prov.setLvlNeighborsIndices(IndicesToIndices(Neighbors.size(), Neighbors));
        ProvSet = new ArrayList<Provider>(Neighbors);
    }
    return LvlNeighbors;
}
public static double CalcNetFraudValue(Provider prov) //Without Diffuse
{

double NetFraudVal = 0;
try
{
String outFileName = "outCalcNetFraudVal." + TASK_ID;
PrintWriter out = new PrintWriter(new FileWriter(outFile, true));
//PrintWriter out = new PrintWriter(new OutputStreamWriter(System.out), true);

List<ProvObj> ProvObjs = prov.getProvObjs();
//out.println("Prov Index " + prov.getIndex() + "\nProv Dev Fraud Val " + prov.getDevFraudValue());
NetFraudVal = prov.getDevFraudValue();

double[] FraudVal = new double[kLevel];
for (int Level = 0; Level < kLevel; Level++)
{
List<Integer> NeighborsIndices = new ArrayList<Integer>();
NeighborsIndices = prov.getkVlNeighborhoods[Index];
//out.println("Level " + Level + "\nLevelDegree " +
prov.getVlDegree(Index));
for (int l = 0; l < prov.getVlDegree[Index]; l++)
{
Provider provider = ((Provider)(ProvObj.objs[NeighborsIndices.get(i)]));
//out.println("Vendor " + provider.EAware + "\nDev Fraud Value " + provider.getDevFraudValue());
if (provider.EAware == false)
{
//out.println("Prov Index " + prov.getIndex() + "\nNeighbor Index " + provider.getDevValue() + "\nNeighbor Dev Val " +
provider.getDevFraudValue());
FraudVal[Level] += provider.getDevFraudValue();
//out.println(" FraudVal[Level] " + FraudVal[Level]);
}
//out.println("Prov Dif prob " + prov.getDevProb() + "\nDecay " +
Math.pow(DifDecay, Level));
NetFraudVal += prov.getDevProb() * Math.pow(DifDecay, Level) * FraudVal[Level];
//out.println("Level " + Level + "\nNetFraudVal " + NetFraudVal);
}
//out.println("Prov Index " + prov.getIndex() + "\nNetFraudVal " + NetFraudVal);
prov.setNetFraudValue(NetFraudVal);
}
close();
}
public static double CalcNetFraudValue(Provider prov)  //With Diffuse
{
  double NetFraudVal = 0;
  double TotalNetFraudVal = 0;
  try {
    String outFile = "outCalcNetFraudVal." + TASK_ID;
    PrintWriter out = new PrintWriter(new FileWriter(outFile, true));
    //PrintWriter out = new PrintWriter(new OutputStreamWriter(System.out),
    //true);

    List<Provider> provObjs = prov.getProIssuers();
    for(int p=0;p<provObjs.size();p++)
    {
      Provider provider = ((Provider)(provObjs.get(p)));
      provider.Aware = false;
      provider.Diffused = false;
    }

    kLevelDiffuse(prov, kLevel);
    //out.println("FRow Index \t" + prov.getFaxIndex()
    );

    for(int p=0;p<provObjs.size();p++)
    {
      Provider provider = ((Provider)(provObjs.get(p)));
      if ((provider.Aware == true) && (provider.Diffused == false) )
      {
        //out.println("FRow Index \t" + prov.getFaxIndex() + "\t Neighbor
        Index \t" + provider.getFaxIndex() + "\tNeighbor Dev val\t" +
        provider.getDevFraudValue() + "\tNetFraudVal\t" +
        NetFraudVal += provider.getDevFraudValue();
      }
    }

    //out.println("FRow Index \t" + prov.getFaxIndex() + "\tnetwork fraud val\t" +
    +NetFraudVal);
    NetFraudVal += NetFraudVal;
  }
  return NetFraudVal;
}
public class CustomComparator implements Comparator<Provider>
{
    @Override
    public int compare(Provider prov1, Provider prov2)
    {
        int Diff = 0;
        double d1 = prov1.getDevFRAUDValue();
        double d2 = prov2.getDevFRAUDValue();
        Diff = -1*(Double.compare(prov1.getNetFRAUDValue(), prov2.getNetFRAUDValue()));
        if (Diff == 0)
            Diff = -1*(Double.compare(d1, d2));
        return Diff;
    }
}

public static Bag SortNodes(final int Mode)
{
    Bag SortedObjs = new Bag(providers.getAllNodes());
    SortedObjs.sort(new Comparator<Provider>()
    {
        @Override
        public int compare(Provider prov1, Provider prov2)
        {
            int Diff = 0;
            double d1 = prov1.getDevFRAUDValue();
            double d2 = prov2.getDevFRAUDValue();
            if (Mode == 0)
                Diff = -1*(Double.compare(d1, d2));
            if (Mode == 1)
                Diff = -1*(Double.compare(prov1.getNetFRAUDValue(), prov2.getNetFRAUDValue()));
            return Diff;
        }
    });
}
```java
Diff = -1*(Double.compare(prov1.getNetFraudValue(), prov2.
getNetFraudValue()));
if (Diff == 0 )
    Diff = -1*(Double.compare(d1,d2));

if (Mode == 2)
    Diff = -1*(Double.compare(prov1.getKlv1Degree(), prov2.
        getKlv1Degree()));

if (Mode == 3)
    Diff = (Double.compare(prov1.PrefProb, prov2.PrefProb));

if (Mode == 4)
    Diff = -1*(Double.compare(prov1.PrefProb, prov2.PrefProb));

return Diff;

}

return SortedObjs;

}

public static double calcFFI()
{
    double count1=0;
    double count2=0;
    
    Bag provObjs = providers.getAllNodes();
    for(int p=0;p<provObjs.getNumObjs;p++)
    {
        Provider prov = ((Provider)(provObjs.objs[p]));
        if (prov.Fraud == true && prov.Influential == true)
            count1++;
        if (prov.Audit == true && prov.Fraud == true && prov.Influential == true)
            count2 ++;
    }
    return (count2 /count1);
}

public static double calcFFNI()
{
    double count1=0;
    
```
double count2=0;

for(int p=0;p<provObj3.numObjs;p++)
{
    Provider prov = ((Provider)(provObj3.objs[p]));
    if (prov.Fraud == true && prov.Influential == false)
        count1++;
    if (prov.Audit == true && prov.Fraud == true && prov.Influential == false)
        count2++;}
return (count2 /count1);
}

public static double calcPMFNI()
{
    double count2=0;
    double count1=0;

    for(int p=0;p<provObj3.numObjs;p++)
    {
        Provider prov = ((Provider)(provObj3.objs[p]));
        if (prov.Fraud == false && prov.Influential == true)
            count1++;
        if (prov.Audit == true && prov.Fraud == false && prov.Influential == true)
            count2++;
    }
return (count2 /count1);
}

public static double calcFNNMI()
{
    double count2=0;
    double count1=0;

    for(int p=0;p<provObj3.numObjs;p++)
    {
        Provider prov = ((Provider)(provObj3.objs[p]));
        if (prov.Fraud == false && prov.Influential == false)
            count1++;
        if (prov.Audit == true && prov.Fraud == false && prov.Influential == false)
            count2++;
    }
return (count2 /count1);
}

public static double calcRecall()
{
double count1 = 0;
double count2 = 0;

Bag provObjs = providers.getAllNodes();
for (int p = 0; p < provObjs.numObjs; p++)
{
    Provider prov = ((Provider) provObjs.objs[p]);
    if (prov.Fraud == true)
        count1++;
    if (prov.Audit == true && prov.Fraud == true)
        count2++;
}
return (count2 / count1);
}

public static double calcPrecision(int K_AUDIT)
{
    double count = 0;
    int num_audit = 0;

    Bag provObjs = providers.getAllNodes();
    for (int p = 0; p < provObjs.numObjs; p++)
    {
        Provider prov = ((Provider) provObjs.objs[p]);
        if (prov.Audit == true && prov.Fraud == true)
            count++;
    }
    return count / K_AUDIT;
}

public static double calcNetDevChange()
{
    double change = 0;

    Bag provObjs = providers.getAllNodes();

    //Calculate deviation change
    for (int p = 0; p < provObjs.numObjs; p++)
    {
        Provider prov = ((Provider) provObjs.objs[p]);
        change += prov.getInitDev() - prov.getFinalDev();
    }
    return change;
}

public static double[] calcNetIncChange()