Robust System for Identifying Procurement Fraud

Amit Dhurandhar, Rajesh Ravi
IBM T.J. Watson Research,
1101 Kitchawan Road,
Yorktown Heights, NY, USA

Bruce Graves
IBM GPS,
294 ROUTE 100,
Somers, NY, USA

Gopi Maniachari, Markus Ettl
IBM T.J. Watson Research,
1101 Kitchawan Road,
Yorktown Heights, NY, USA

Abstract

An accredited biennial 2012 study by the Association of Certified Fraud Examiners claims that on average 5% of a company's revenue is lost because of unchecked fraud every year. The reason for such heavy losses are that it takes around 18 months for a fraud to be caught and audits catch only 3% of the actual fraud. This begs the need for better tools and processes to be able to quickly and cheaply identify potential malefactors. In this paper, we describe a robust tool to identify procurement related fraud/risk, though the general design and the analytical components could be adapted to detecting fraud in other domains. Besides analyzing standard transactional data, our solution analyzes multiple public and private data sources leading to wider coverage of fraud types than what generally exists in the marketplace. Moreover, our approach is more principled in the sense that the learning component, which is based on investigation feedback has formal guarantees. Though such a tool is ever evolving, an initial deployment of this tool over the past 6 months has found many interesting cases from compliance risk and fraud point of view, increasing the number of true positives found by over 80% compared with other state-of-the-art tools that the domain experts were previously using.

Introduction

Given the subversive nature of fraud, such activities can be well hidden and difficult to identify and trace to the responsible parties. Routing out the cause, including identifying entities indicative of fraud, can be a difficult if not sometimes an insurmountable task. This is mirrored in an accredited 2014 survey conducted across 100+ countries, by the Association of Certified Fraud Examiners (ACFE) (ACFE 2014), who claim that on average 5% of a company's revenue is lost because of unchecked fraud every year. The reason for such heavy losses according to them is that it takes around 18 months for a fraud to be caught and audits catch only 3% of the actual fraud. A large portion of risky activity is caught through whistle blowers. This begs the need for better tools and processes to quickly and cheaply identify potential malefactors.

In the modern era, a phenomenal amount of digital data is involved in nearly every type of business. Modern developments in both software and hardware have allowed for data analysis techniques to be developed and directed to detecting and identifying fraud and its perpetrators. In the art of fraud detection and risk analysis, analytical systems are developed and relied upon to analyze data and make predictions as to the presence of risk/fraud. Despite considerable advances in fraud detection, the ways in which parties can commit fraud have also advanced and become more elusive. There is a persisting need for novel techniques and systems for the detection and identification of fraud and the conspirators responsible, that have a low false positive rate.

We describe one such system in this paper. Though our system architecture and analytics flow would be applicable in a wide variety of domains, we focus our attention on identifying procurement related risk or fraud. In large companies there are procurement groups, which buy goods and services from tens of thousands of vendors/suppliers all across the globe every year amounting to billions of dollars of spend. Given the scale of these operations, it is hard to enforce airtight compliance procedures in the interest of time and money, which makes it a breeding ground for nefarious activity. Here are a couple of real examples of procurement

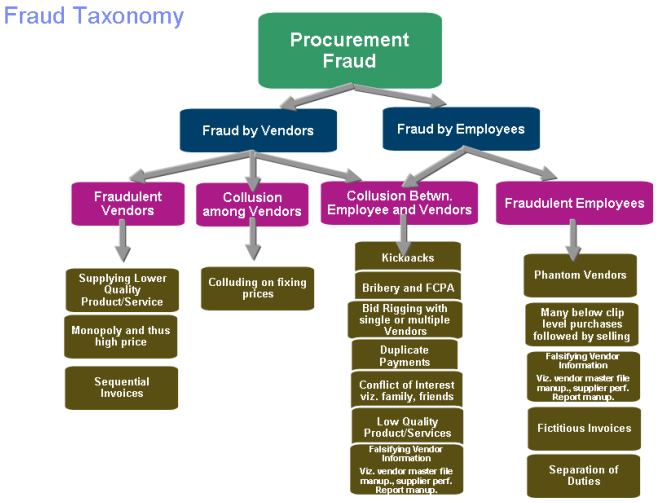

Figure 1: Above we see our taxonomy related to procurement fraud.

fraud. An employee of a large company bought a few USB drives every month at the company's expense over a couple of years and was selling them in the black market. Since, USB drives are inexpensive they were below the spend clip levels set by the company per purchase order and thus it went undetected for years. Another example was that of a company employee creating a company on his spouse's maiden name and then routing business to that company.

Studying hundreds of such cases of procurement fraud in the last few years, we created a taxonomy ellucidating the broad categories to which these different cases belonged to. This taxonomy is seen in figure 1. The taxonomy, we believe, gives more structure to the problem than just enumerating individual cases. Moreover, it assists us in better understanding the different types of fraud as well as the distribution of the cases across these categories. At a high level there are only two entities namely, company employees and vendors that the company buys from. Hence, fraud occurs through actions of any of the individual entities or through their interactions. In our review of prior fraudulent cases, we found that fraud based on collusion between employee and vendor had the highest occurance. Collusion essentially is secret or illegal cooperation or conspiracy, especially in order to cheat or deceive others. Relative to procurement, collusion involves at least two parties making an arrangement or agreement which provides at least one of the parties an unfair and illegal competitive advantage.

There are a few products that quantify procurement fraud. Based on our study they (Analytixs 2013; IBM-RCAT) mostly tackle just the leftmost column in figure 1, that is vendor fraud. Moreover, they are mainly based on business rules with mininmal analytics. Others are based on supervised learning (Alaric 2013) and thus require labelled data, which is often unavalaible in our setting. A few consulting firms also have products mainly relating to text analytics that scan emails and identify employees based on high risk words or phrases. Certain toolboxes in SPSS can identify aliasses (Jonas 2012a) of a person or direct relationships (Jonas 2012b) such as husband (employee)$\rightarrow$ wife (vendor), but not multihop relationships such as husband (employee) $\rightarrow$ wife $\rightarrow$ cousin(vendor) in an efficient manner. There are other tools (i2 IBM 2013), which are mainly used for investigative purposes but not for detection. All of these tools cover small portions of the taxonomy but none of them is even close to being comprehensive.

Our system is the most comprehensive that we know of, since we model collusion detection and all the different types of fraud, not being limited to only certain restricted types. As we will see later, to accomplish this we have the ability to analyze various public and private data sources including social network data to detect collusion. Moreover, our approach is more principled in the sense that our online updating scheme based on investigation feedback has theoretical guarantees, with the goal of reducing false positives and at the same time maintaining interpretability.

System Design

In this section, we provide details of our tool. We first discuss the different analytic components followed by an overview of the overall architecture.

Analytics Flow

As mentioned before, our system analyzes various public and private data. This data maybe structured or unstructured. It may contain independent instances or the instances maybe linked. To deal with this different types of data and to come up with a credible list of risky individuals (vendors/employees) we intelligently amalgamate various analytical tools such as text analytics, social network analytics, statistical outlier detection techniques, unsupervised learning and online learning with precious domain expertise comprising of business rules and assigning of importance/weights to different anomalous events. An overview of the analytics flow is depicted in figure 2.

Data Sources There are multiple data sources that feed our tool. Besides the usual transactional data, we access many private and public sources of data, some of which we now mention. This is by no means an exhaustive list but it outlines many important data sources and showcases the diversity of data used. Here are some private data sources other than transactional data.

- *Vendor Master File:* This document contains information of each of the vendors registered with the company. It has their name, addresses, account numbers, the date of registration and other relavant information.
- *RFx data:* This data source contains information about which bids for a particular product or service were sent out by whom in the company and to which vendors. It also has information about who won the bid and what the competitive prices proposed by the different vendors were. In some cases we may also have fair market value for the specific product or service.
- *Risky Commodity List:* Commodities are high level groupings of products or services. Certain commodities have less stringent checks (viz. higher clip levels) when buying items in that commodity and thus vendors or employees might indicate their item belongs to one of these commodities, when in fact it does not in an attempt to bypass the required checks. Such commodities are thus deemed as risky and it is important for a tool like ours to take into account this information.
- *Global Clip Levels:* Generally speaking clip levels are dollar amount cutoffs for a particular purchase below which the purchase goes through significantly less checks and approvals than a purchase that is above it. These vary from country to country as well as commodity to commodity. It is important for us to know what these clip levels are for catching potential bypasses.
- *Social Networking Data:* Company employee emails in terms of content, who they were sent to and how frequently two parties interacted could be useful information. In terms of external sources, certain businesses sell information about individuals regarding where all and with whom they lived in the last decade. Also information regarding their spouses and other close relatives is

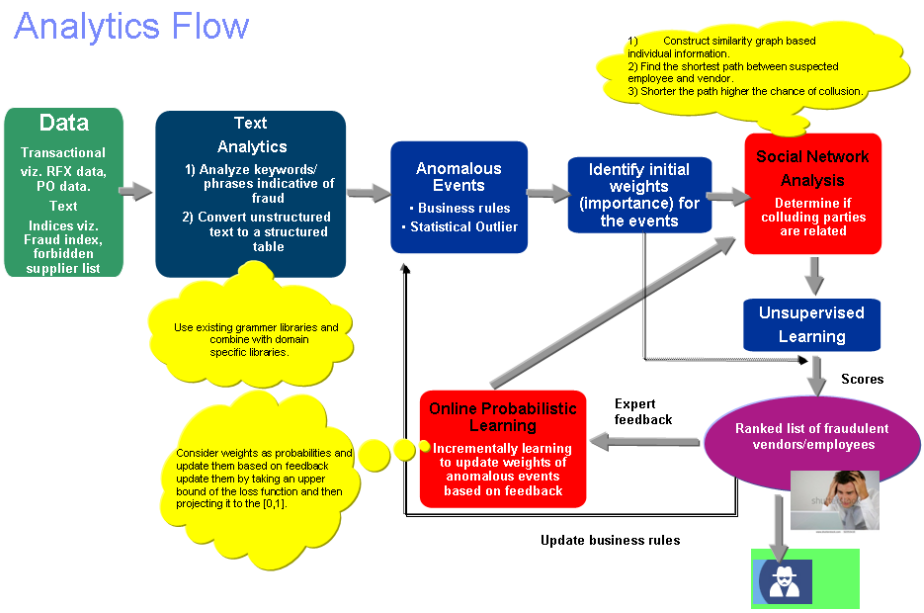

Figure 2: Above we see the different analytical components along with their interactions in our system.

available. Some times public profiles on social networking sites could also be accessed to reveal relations between individuals.

Besides, the proprietary data sources here are some examples of the public data we access:

- *Forbidden Parties Lists:* The US government every year releases lists of suspect businesses. The Denied Persons List (DPL) (Bureau 2013) and the Excluded Parties List (EPL) (System 2013) are two of the more widely used ones.
- *Country Perception Indices:* The corruption perception index (CPI) (Intl. 2013) is another public source, which ranks countries based on the levels of corruption faced by people in their daily lives in these countries. This list is created by different experts around the world such as credible analysts and businessmen.
- *Tax Haven Locations List:* This is again self explanatory. Having vendors located in tax haven locations or their bank accounts being present in such locations could be a red flag especially with other digressions.

Text Analytics There are multiple sources that we use which contain unstructured data. In fact, even the standard transactional data which has invoice and purchase order (PO) information, contains text fields which can serve as a rich source of information. In our system we mine this comments field to check if the work was not authorized by the company by matching certain keywords and phrases. We also try to extract the invoice date and compare it with the PO create date to verify that they occurred in the correct chronological order. Sometimes, no PO is created a priori and it is created as an after thought, which is not acceptable. We can also check if there are indications that the work started prior to PO creation or that the actual commodity code is different from what has been entered into the appropriate structured field indicating category.

Other unstructured sources include risk reports, which can be mined to get a feel for the political situation in a certain country or geography. Employee emails can be mined to see if high risk word or phrases have been used in interaction with vendors indicating possible malicious activity.

Anomalous Events Identification Anomalous events are a combination of expert provided business rules and analytical techniques such as statistical outlier detection techniques and insights derived from text analytics. Each event can be viewed as an independent feature, which gives us additional insight into the level of risk associated with a particular entity. We have 100s of such events and the list is of course ever growing.

It isn't possible to enumerate and explain all of these events so we provide a sneak peek into some of them to make lucid the kinds of events we are talking about.

- *Vendor Requestor Monopoly:* This rule checks to see if a high percentage (say >90%) of invoices from a particular vendor are approved by a single employee in the company. In this case, there is a higher chance of collusion as a single employee has quite some control in accepting or rejecting the vendor invoices.
- *Benfords Test:* Benfords law (Benford 1938) provides an expected distribution of the frequency of the most significant digit in a sequence of numbers given that they were generated from a real-world process. This distribution has been observed in varied domains such as, in surface areas of rivers, molecular weights, death rates, street addresses.

  We can perform statistical testing based on this by comparing the expected distribution with the actual distri-

bution of most significant digit of the invoice numbers/amounts for a vendor observed in our data. We can perform the chi-square test, where the null hypothesis states that the invoice numbers were generated from a real source. We trigger the event in this case, if the $p-value \leq 0.05$.

- *PO Create Date after Invoice Date:* A standard procedure is to first create a PO followed by submitting invoices for products sold or services rendered. Sometimes, this procedure may not be followed which is non-compliant behavior. The invoice dates may be indicated in a structured field, however it is sometimes different from the actual invoice date, which can be extracted from a comments field entered by an approver other than the requestor. We can thus check, if both of these invoice dates, which ideally should be the same, are at least after the PO create date, else we trigger the event.
- *Mix of PO and Non-PO Invoices:* Usually a vendor will provide goods or services with a PO corresponding for each such transaction. For some vendors the goods or services are so cheap and frequent that PO is not required to be created. However, having a mix of both is a red flag since, vendors would belong to one of these categories.
- *Invoices with Risky Commodity Code:* As described in the data sources subsection, certain commodity codes have higher clip levels after which they go through stringent checks. To bypass these checks an incorrect high clip level commodity may be entered for an invoice, which would trigger this event.
- *Country Corruption:* This event is triggered for countries with CPI below a certain value (viz. < 50). The confidence that is computed for this event is described in the next subsection.
- *Unfair Win:* Once a bid is thrown by a company, if the vendor that wins the bid demands significantly more than the historical/fair market price, then we trigger this event.

Initial Importance Weighting and Confidences of Events We mentioned before that we do not have labeled data. Hence, we cannot train a supervised model to rank entities based on a fraud score. To limit the number of false positives we come up with an initial weighting signifying the importance of the different events using a combination of domain expertise and the frequency of occurrence of the different events. In particular, we derive a weight in $[0, 1]$, where a higher weight indicates that the event is more important in identifying fraud. These weights are based on evaluation of the events with experts and us devaluing events that occur frequently based on our analysis of real data.

In the design, we insist on weights of individual events being normalized i.e., between $[0, 1]$, so that they are interpretable. The weight of an event can be viewed as the probability that a fraud has occurred given that the particular event was triggered. We believe, such semantics make it easier for the expert not only to be able to interpret the importance of events available in the system, but also in determining the relative importance of new events that may be added in the future. This is not the case if the events have unbounded weights, as is witnessed in some current tools.

Algorithm 1 Computing total confidence of collusion based on social networking given collusion confidence based on other (viz. transactional, Rfx) data.

Input: Social network $G_S = (V, E)$, suspect entities (V_1, V_2), confidence of collusion based on other data c_r, social confidence threshold t_s and importance of social network information $\alpha \in [0, 1]$.
Output: c_{tot} {Total confidence of collusion.}
Set $W = \phi$ {Setting list of weights to empty initially.}
for all $E_{ij} \in E$ **do**
  if V_i, V_j are same person **then**
    Set $p = 1$ {p signifies probability that V_i and V_j are colluding based on social network information.}
  else if V_i, V_j are close relatives **then**
    Set $p = 0.95$
  else if V_i, V_j are friends/acquaintences **then**
    Set $p = 0.9$
  end if
  Set $w_{ij} = -log(p)$ {Edge weight for E_{ij}}
  $W = W \cup (w_{ij}, E_{ij})$ {Storing weights with the corresponding edge.}
end for
Set $c_{social} = ETShortestPath(G_S, W, V_1, V_2, t_s)$ {Find the exponentiated shortest path between V_1 and V_2 thresholded by t_s, i.e. if the shortest path length at any intermediate step in Dykstra's algorithm starting at V_1 is $> t_s$ return 0 else return e^{-l_s}, where l_s is the length of the shortest path.}
Return $c_{tot} = min(c_r + \alpha c_{social}, 1)$

Confidences of events are complimentary to their weights. While weights indicate the importance of an event in detecting fraud, confidences signify our belief in the occurrence of the event. For example, if we consider the *vendor requestor monopoly* event, it either occurs or doesn't occur. If the event is triggered our confidence would be 1 else it would be 0. As we can see here, the confidence is different from the weight, which is a fixed number irrespective of if the event occurs or not. We use both of these indicators, that is, confidences and weights to determine the probability of fraud by an entity, which we will visit in detail in subsection .

Confidences for most events are either 1 or 0 depending on if they are triggered or not respectively. However, they are a real number between $[0, 1]$ for some events. A good example is the *country corruption* event. We calculate the confidence for this event as follows: $c_{\text{CPI}} = \frac{100-\text{CPI}}{100}$. CPI lies between $[0, 100]$, where a higher CPI indicates lesser risk. However, in our design we want the confidences to lie in $[0, 1]$, where a higher value indicates a stronger signal for the event.

Social Network Analysis We have described events on transactional data or RFx data that indicate the possibility of collusion. Analyzing social network data could significantly enhance our confidence in such findings.

In algorithm 1, we observe how information from such

data sources could be used to enhance our confidence in the fact that collusion has occurred. c_r can be obtained from transactional or Rfx data. For example, if we consider the event *Unfair Win* being triggered, we could set $c_r = 0.5$ as we are not hundred percent sure that collusion has occurred. Now accessing social network data and using algorithm 1, we can further strengthen our confidence in the finding that the two entities are colluding. t_s is a threshold used in the algorithm to improve efficiency as paths longer than a certain length will lead to very small c_{social} and are not worth calculating. t_s can be easily set by deciding paths lesser than what significance are not worth extending. For example, we could set $t_s = -3log(0.9) = 0.32$ indicating that if V_1 and V_2 have a weaker relationship than being friends of friends of friends, i.e. friends 3 hops away, then they are not socially related. α is just a discount factor for the social part signifying the importance we want to assign to it. Again since we want the confidences to be normalized we let c_{tot} have a maximum value of 1.

If social data is not available we can use the corporate email network as a weak proxy. Here the weights of the edges would be determined by the frequency of the communication between the various entities and checking the content for high risk words or phrases.

Unsupervised Learning From experience with dealing with domain experts across different industries, we believe that for them to gain faith in the system it is necessary to have high precision even if it is at the expense of some recall. In other words, many false positives can immediately inhibit interest amongst practitioners and experts for such a tool. Thus, the unsupervised component is of relatively low importance in the initial deployments of this application.

Nevertheless, methods from infrequent pattern mining (Haglin and Manning 2007) could be of considerable significance here, in identifying low support but high recall sequences in the data. These methods are of interest in anti-terrorism, where one monitoring suspicious money movements through different bank accounts.

If the patterns found by these methods are interesting they could lead to new rules that need to be checked in the future. This capability should not be ignored as humans are always adapting and hence, it is important to uncover suspicious behaviors not known to be risky by the experts.

Ranking Risky Entities As we have discussed in previous subsections, each event i is associated with a weight $w_i \in [0, 1]$ and given an entity $\mathcal{E}$ we have a corresponding confidence $c_i^{\mathcal{E}} \in [0, 1]$ of occurrence. Thus, the probability that an entity is fraudulent/risky is 1 minus the probability that it is not fraudulent. The probability that it isn't fraudulent is the probability that none of events that are triggered for it are fraudulent. Formally, given a total number of n possible events $I = \{1, ..., n\}$ the probability that entity $\mathcal{E}$ is fraudulent is given by,

$$P_f^{\mathcal{E}} = 1 - \prod_{i \in I} (1 - w_i c_i^{\mathcal{E}}) \tag{1}$$

where for any event i, $w_i c_i^{\mathcal{E}}$ is the probability that entity $\mathcal{E}$ is fraudulent given that the event was triggered with confidence $c_i^{\mathcal{E}}$. Notice that for events that are not triggered for entity $\mathcal{E}$, the corresponding confidences $c_i^{\mathcal{E}}$ would be 0, thus not contributing to the overall fraud probability.

Algorithm 2 A principled method for updating weights based on user feedback for a case/entity that maintains interpretability.

Input: $\{w_1, ..., w_n\}$, $\{c_1^{\mathcal{E}}, ..., c_n^{\mathcal{E}}\}$ and y {Inputs are weights, confidences and the feedback for the case.}
Output: $\{w_1, ..., w_n\}$ {Updated weights.}
Initialise $g_s = 0$ and $\eta = 0$ {η is the learning rate.}
if $y == 0$ {Feedback is entity is not fraudulent.} **then**
  $\eta = 0.001$
else if $y == 1$ {Feedback is entity is fraudulent.} **then**
  $\eta = 0.05$
end if
if $\eta \neq 0$ **then**
  for all $i \in I$ **do**
    $g_i = ln(1 - w_i c_i^{\mathcal{E}})$
    $g_s = g_s + g_i$
  end for
  for all $i \in I$ **do**
    if $c_i^{\mathcal{E}} \neq 0$ {Update weight only if event is triggered.} **then**
      $g_i = g_i - 2\eta(e^{2g_s} + y - 1)$
      $w_i = \frac{(1 - e^{g_i})}{c_i^{\mathcal{E}}}$
      $w_i = \mathcal{I}(w_i > 1) + w_i \mathcal{I}(w_i \in [0, 1])$ {Projecting to $[0, 1]$. $\mathcal{I}(.)$ is an indicator function.}
    end if
  end for
end if
Return $\{w_1, ..., w_n\}$

Online Probabilistic Learning On presenting the user with a ranked list of possibly fraudulent candidates, the user can further investigate entities that interest him/her. For each entity he can enter 3 possible outcomes into the system. He can indicate that, i) the entity was fraudulent or ii) the entity was not fraudulent and uninteresting or iii) the entity was not fraudulent but wants to keep monitoring it. Depending on the feedback (indicated in figure 2) we want to update the current weights of the events triggerred for that entity. A key aspect of the updating procedure is that we want to *maintain interpretability* of the individual weights. In other words, we want the weights to be in $[0, 1]$ after an update. Moreover, we want our efficient online learning procedure to have quality guarantees.

Let y denote the feedback variable, which is 1 for case i), 0 for case ii) and 2 for case iii). For case iii) we maintain the current weights as is seen in algorithm 2, since we neither want to emphasize or deemphasize the relevant events. Hence, we want $P_f^{\mathcal{E}}$ to be as close to 1 when the feedback is i) and to be close to 0 when the feedback is ii). With this we define the following least squares loss function that we want to optimize for our problem,

$$\mathcal{L} = (y - P_f^{\mathcal{E}})^2 \tag{2}$$

where if $\forall i \in \{1, ..., n\}\ p_i = 1 - w_i c_i^{\mathcal{E}}$, then from equation 1 we have,

$$P_f^{\mathcal{E}} = 1 - \prod_{i \in I} p_i \tag{3}$$

From equations 2 and 3 our loss function can be written as,

$$\mathcal{L} = (y - 1 + \prod_{i \in I} p_i)^2 \tag{4}$$

If we set $g_i = ln(p_i)$ and $b = ln(2(1-y))$ we can rewrite the above equation as,

$$\mathcal{L} = e^{2\sum_{i \in I} g_i} + \frac{1}{4} e^{2b} - e^{\sum_{i \in I} g_i + b} \tag{5}$$

The function in equation 5 is not convex. For convex loss functions however, it has been shown in (Zinkevich 2003) that online gradient descent with projections onto a convex set, which is $[0, 1]$ in our setting, achieves low regret. For this result to be applicable to us, we need to derive a convex approximation of our loss function. We accomplish this by linearizing the portion in equation 5 after the negative sign, which makes the following new function a convex upper bound of $\mathcal{L}$,

$$\mathcal{L}_u = e^{2\sum_{i \in I} g_i} + \frac{1}{4} e^{2b} - e^b(\sum_{i \in I} g_i + 1) \geq \mathcal{L} \tag{6}$$

Taking partial derivatives of $\mathcal{L}_u$ with respect to (w.r.t.) each of the non-zero g_i's and setting each such derivative to zero, leads to the update procedure described in algorithm 2. The weights are then projected onto $[0, 1]$, where $\mathcal{I}(.)$ is an indicator function, which is 1 when the parametrized condition is true and is 0 otherwise. Thus, our solution achieves low regret w.r.t. $\mathcal{L}_u$, which is a convex upper bound of $\mathcal{L}$.

η in algorithm 2 is the learning rate. We set a significantly higher learning rate for fraudulent cases than for non-fraudulent ones, since in the real world most cases are likely to be non-fraudulent. Given this, we do not want our weights to converge to 0 quickly when feedback on non-fraudulent cases is provided. At the same time we want to quickly emphasize events that lead to entities being fraudulent as they are likely to be far and few but critical.

Architecture

Given that we have a system that runs daily processing and analyzing billions of dollars of spend spread over millions of invoices and tens of thousands of vendors across the world, we need an architecture that is robust and scalable.

The solution architecture primarily comprises of three major components:

- Central data processing system and data warehouse
- Big data processing system
- Risk analytics engine

Firstly, the central data processing system, is implemented using IBM InfoSphere Information Server, which controls and regulates the flow of data in the system. Moreover, it orchestrates the triggering/execution of the different anomalous events and other analytical components. It also uses a number of data adaptors for the various public and private data sources that we have mentioned before, along with an extensible mechanism to add new adapters. One of the major challenges is to integrate the different data sources in a seamless manner so as to perform advanced analytics and report high risk entities. The central data warehouse bares this burden and is implemented using IBM DB2, which uses extensible high dimensional data modeling methods to support scalable data storage and retrieval.

Secondly, the big data processing system, implemented using IBM InfoSphere BigInsights and IBM InfoSphere Streams, complements the central data processing system for analyzing massive amounts of data. IBM InfoSphere BigInsights is based on an open source Apache Hadoop platform, which includes social media accelerators and text analysis toolkits that can be leveraged to process large structured/unstructured data sources such as social media, email, etc., using low cost commodity hardware. IBM InfoSphere Streams can be used to capture and process data in near real-time.

Finally, the risk analytics engine contains a library of anomalous events and analytic modules based on IBM SPSS, Python and Java that work in a co-ordinated fashion to score and rank risky entities.

User Acceptance Testing

Our first client for this system is IBM itself. The reasons for this are at least three-fold: First, if IBM itself is not using the system why would an external client be interested. Second, using IBM as a test bed we can enhance the system by removing kinks and testing it on large amounts of data consisting of billions of dollars of spend with hundreds of thousands of employees and tens of thousands of vendors. Third, a successful deployment within IBM will result in the relevant organizations providing support for the system in front of external clients. Saying this we have already demoed the system to potential external clients with positive feedback.

Our initial deployment has been for two critical organizations within IBM namely, a) Accounts Payable and b) Policies and Practices. The former tries to identify fraudulent entities, while the latter tries to identify entities posing a significant compliance risk. Based on their experience using the tool over the past 6 months, both of these organizations claim that it has increased their efficiency by approximately 80%. What this means in data mining terms is that our top 20 list, which they review every week, has on average **80% more true positives** than the state-of-the-art methods they had been previously using. This is a huge jump in performance and potentially already worth millions of dollars in savings. Moreover, they claimed to have found true positives some of which weren't in the last 15+ years, because of our text analysis coupled with our scoring scheme. Overall, the *main reasons for the improved efficacy are our scoring model, the specific implementations of some of the anomalous events (viz. fuzzy matching, statistical outlier detection) and the scalable user friendly architecture of our system.*

Summary:

Vendor	Location	Total Invoice Amount	Average Invoice Amount	No. of Invoices	No. of Events	Profile Risk Score	Transactio Risk Score	Perception Risk Score	Collusion Risk Score	Overall Risk Score
Vendor_3451	US - Illinois	$381,207.05	$8,865.28	43	7	0	82	0	30	87

Events:

Risk Category	Event Name	Risk Score
Collusion	Vendors with high percentage invoices from same requestor	30
Transaction	Consecutive Invoice Numbers	10
Transaction	Vendor with mix of invoices with PO and without PO	29
Transaction	Invoice Amount jumps by 50pct	10
Transaction	Vendor who created PO after Invoice	50
Transaction	Supplier with same material group changes	30
Transaction	Benford Law	10

Figure 3: Above we see the overall risk score (x100) for a potentially fraudulent vendor and the events that were triggered for it.

Summary:

Vendor	Location	Total Invoice Amount	Average Invoice Amount	No. of Invoices	No. of Events	Profile Risk Score	Transactio Risk Score	Perception Risk Score	Collusion Risk Score	Overall Risk Score
Vendor_7549	US - Illinois	$8,524,122.1	$66,594.7(	128	7	0	95	0	0	95

Events:

Risk Category	Event Name	Risk Score
Transaction	Consecutive Invoice Numbers	10
Transaction	Vendor with mix of invoices with PO and without PO	28
Transaction	Vendor who created PO after Invoice	50
Transaction	Vendor who created invoices with risky commodity	30
Transaction	Benford Law	10
Transaction	Invoices were not authorized by IBM (by Text Analysis)	50
Transaction	Work started prior to PO (by Text Analysis)	50

Figure 4: Above we see the overall risk score (x100) for a potentially non-compliant vendor and the events that were triggered for it.

Given the closely related but different focus of these two organizations, we provide one example belonging to each of these types of cases that our tool found and were considered high risk by the experts.

Before we describe these cases, we ellucidate the data analyzed. These results are based on analysis of one year of real IBM procurement data, which consists of millions of invoices and has around 35000 vendors. In addition to this, we also access the various public and private data sources mentioned before. *For aesthetic reasons we show the risk/fraud score multiplied by 100 and then rounded* in the screen shots.

In figures 3 and 4, we see a more detailed view of two vendors that potentially present high risk and were identified by our tool. In the summary tab in both these figures, we see the (anonymized) vendor name followed by more vendor specific information in the next four columns. The *No. of Events* column, denotes the number of events triggered for this vendor. The last column is the risk score (x100) computed by our method. The preceding four columns are risk scores based on a partition of the universal list of events. Essentially, they act as thumbnails providing a quick sneak peek into where the problem lies for the particular entity. So for example in figure 3, most of the risk for that vendor is associated with its invoice/transactional data. In addition, there is also a collusion risk. Analogously, for the vendor in figure 4 all his risk can be attributed to non-compliant and potentially fraudulent transactions. More details about the particular events that were triggered leading up to these risk scores can be found in the *Events* tab below[1]. The tool also allows the user to perform multiple other tasks such as see the transactional and other accessed data, filter invoices by triggered events, however due to limited space we do not show screen shots for those.

Further delving into the case in figure 3, we see that there is a risk of collusion as a single requestor/employee is approving a bulk of the invoices sent by the vendor. Coupled with this we see that there is more than a 50% jump in spending with this vendor from a six month to the next six month period. Moreover, five other events are triggered making this vendor high on the suspect list.

[1]The risk score for each event is its weight multiplied by confidence (x100).

The case in figure 4, showcases the usefulness of mining unstructured data. The three highest risk events are triggered due to text analysis. Other than these we have the event that checks for risky commodity codes being triggered, which indicates a potential for bypass. These along with triggering of three other lower risk events makes this vendor high risk.

Discussion

In this paper, we described a daily running system that is currently being used by various organizations within IBM. The plan is to continuously extend and improve the system and deploy it with other internal and external clients. The idea is to have a universal list of events with default initial weights that are customizable by the experts in the particular organization. Therefore, every new organization/client will have their own independent instantiation of our system. An effective system such as this will potentially not only identify fraud but will also serve as a deterrent for entities that are currently committing fraud or those that are considering of doing it in the future.

Deploying such a system has many challenges. One of the main challenges is being able to access sensitive private data. For example, some of our events require accessing employee bank account data, which is highly sensitive information. We tackle this problem by letting HR run the specific events in their firewall and returning only the result. Afterall, our system only needs to know if a particular event was triggered for an entity or not. For instance consider the event, where we check to see if a vendor bank account number matches an employee bank account number. This would require knowing the bank account numbers of both these entities. However, we could send HR a list of vendor bank accounts and they could match them against the employee database and return to us the list of vendors that matched. This completely absolves them from sending us the employee bank account numbers. An analogous strategy can be used for other events that require sensitive data, where only the list of entities for whom the event was triggered is returned to us by the responsible party.

Another challenge is the type of data that the client is ready to subscribe to. The client may not want to buy or pay for accessing certain paid data sources that may carry useful signals. In certain cases, a particular geography may have restrictions on the public data that may be used incriminate anyone. For example, in Europe using publicly available social data to accuse an individual is strongly discouraged.

Given these restrictions in regards to accessing different data sources, which could lead us to potentially miss important signals, we can either settle with what we have or try to be more creative. From an analytics perspective a possible way of mitigating the impact of the absence of these data sources is to implement an extrapolated scoring scheme. What we mean by this is that using available data from existing clients (viz. IBM) we can figure out through techniques such as association rule mining the likelihood of certain combination of events to co-occur. Based on these insights we can score entities of a new client with their limited data by seeing the events that have been triggered for each entity and appending those with events that are likely to be triggered if in fact they had the data. In essence, we are extrapolating the score based on our knowledge of other similar clients. This extension could of course be erroneous but its probably our best estimate given the missing data sources.

Many such improvements are being implemented and tested in the system as we speak. Moreover, we are involved in serious discussions about carrying over the design and adapting the various analytical components to detecting fraud in other related domains.

Acknowledgement

We would like to thank Anthony Mazzati from *Accounts Payable*, Pablo Varela and Kathy Moon from *Policies and Practices*, Brian Hopper from *Global Process Services* and many others from different organizations (FCPA, Auditing) within IBM who continue to help us in improving this tool and are instrumental in its success.

References

ACFE. 2014. Report to the nations. In *Global Fraud Study*. http://www.acfe.com/uploadedFiles/ACFE_Website/Content/rttn/2014-report-to-nations.pdf.

Alaric. 2013. Fraud detection. In *Payment Processing and Fraud Prevention*. http://www.alaric.com/.

Analytixs, A. 2013. Fraud detection. In *yourFirstStrike*. http://www.apexanalytix.com/firststrike/fraud-detection.aspx.

Benford, F. 1938. The law of anomalous numbers. *Proc. of the American Philosophy Soc.* 78:551–572.

Bureau, I. . S. 2013. Denied persons list. In *US Dept. of Commerce*. http://www.bis.doc.gov/index.php/policy-guidance/lists-of-parties-of-concern/denied-persons-list.

Haglin, D., and Manning, A. 2007. On minimal infrequent itemset mining. In *Proc. of Intl. Conference on Data Mining*.

i2 IBM. 2013. i2 fraud intelligence analysis. In *Planning and Mgmt. Services*. http://www-03.ibm.com/software/products/en/fraud-intelligence-analysis/.

IBM-RCAT. Rcat. In *ISC*.

Intl., T. 2013. Corruption perceptions index. http://www.transparency.org/research/cpi/overview.

Jonas, J. 2012a. Entity analytics. In *IBM SPSS*. ftp://public.dhe.ibm.com/software/analytics/spss/ documentation/modeler/15/en/EA_UserGuide.pdf.

Jonas, J. 2012b. Identity insight. In *IBM SPSS*. http://www-03.ibm.com/software/products/en/infosphere-identity-insight/.

System, A. M. 2013. Excluded parties list. In *US Govt.* https://www.sam.gov/portal/public/SAM/.

Zinkevich, M. 2003. Online convex programming and generalized infinitesimal gradient ascent. In *Proc. of the 20th Intl. Conference on Machine Learning*.