Investigating Insurance Fraud using Social Media

Manuel Diaz-Granados, Javier Diaz-Montes, and Manish Parashar
Rutgers Discovery Informatics Institute, Rutgers University, USA
contact author: javier.diazmontes@gmail.com

***Abstract*—Since the social media hype started in the early 2000s, the Internet has bloomed with user-generated data. The content generated by users in social media varies from blogs, forums, social network platforms, and video sharing communities. This data has a special emphasis on the relationships among users of the community. As a consequence, social media data contains significant information about their creators and people around them. For this reason, law enforcement and insurance companies, among others, are starting to explore ways of using this data to identify unlawful or fraudulent activities. In this work, we present a solution to extract and analyze social media data in pursuit of identifying insurance fraud. We describe and evaluate an initial prototype of our solution that has been implemented on top of the CometCloud framework. We show how our solution is driven by the insights obtained from the data and it is able to extract data relevant to the investigators.**

***Keywords*-Fraud detection, CometCloud, Social media, Workflows, Cloud**

I. Introduction

Insurance fraud is a major crime that is costing insurance companies and insurance customers billions of dollars every year. According to the Coalition Against Insurance Fraud, nearly $80 billion in fraudulent claims are made in the U.S. [5] every year – this includes all types of insurances. For this reason, insurance companies are actively working to prevent fraud and looking for new technological opportunities to improve their procedures. For example, according to the Coalition Against Insurance Fraud, the property and casualty insurance industry alone spends more than $1 billion each year on anti-fraud programs [5].

In this context, the Internet and, specifically, social media are becoming a valuable source of information that insurance companies are starting to exploit to unveil fraud. However, extracting significant and valuable insights from this data is a challenging Big Data problem. For example, a person involved in a car accident submitted a claim to his insurance company alleging to be (physically) disabled. However, the details from the accident and resulting severity of damages did not seem to align. Hence, the insurance company manually combined private insurance data with social media data, and eventually found through his wife's Facebook page evidence of potential fraud. This process involved obtaining and analyzing large amounts of data spanning diverse categories such as social networks, Twitter streams, Facebook postings, public blogs, etc. [1] in combination with private data, e.g., the Nexis database [9]. Therefore, efficient and effective exploration of this data is a non-trivial task, owing to data volume and complexity, and highly convoluted nature of the latent information that investigators, or end-users, seek. It is not uncommon that to validate given hypothesis the end-user has to "patch together" seemingly disparate and often unstructured pieces of information. This may involve a tedious manual search across different Internet sites, which is severely limited by the rate at which individual sources can be inspected and complex patterns uncovered. Moreover, the process is inherently sequential and ultimately error-prone, and can be technologically limited (consider for example browsing several web pages simultaneously, where each page does not fit on a screen).

In this paper we present the design of an end-to-end framework that can automate large-scale data analytics and empower special fraud investigators with big data capability. Our framework extracts, aggregates, analyzes, organizes, and visualizes information relevant to the input queries generated by the end-users. By providing hierarchically structured and annotated overview of the Internet content relevant to the query, complemented with provenance records, the system can accelerate investigations and can significantly improve their accuracy. This framework can be offered as-a-Service and it is designed to account for interactivity and human-in-the-loop processes to guide audio and video streams analysis, which remain challenging for machines. To manage computational complexity of the underlying data processing tasks the system leverages parallel and distributed computing techniques.

The rest of the paper is organized as follows. In Section II we present our approach for fraud detection, followed by its implementation in Section III and evaluation in Section IV. We finalize with related work in Section V and our conclusions in Section VI.

II. Fraud Detection Approach

In this work we propose an approach to enable fraud detection though large scale exploration of social media. Our approach is data-driven to dynamically explore different social media platforms following insights obtained from the analysis of prior data as well as data obtained at runtime. There are two parts: (i) Identifying claimant on social media (e.g., Facebook, Twitter); and (ii) Obtaining, analyzing, and organizing relevant information from different social media platforms.

A. Identifying claimant on social media

According to our collaborators from a large insurance company, in most of the cases there is some prior knowledge about a claimant. This knowledge includes information such as, data birth, email address, phone number, current and former physical addresses, relatives, neighbors, employers, etc. For example, in Facebook this information can refer to the "About" section plus the list of friends. We propose to use a the Bag-of-words model [8] to use prior information to find potential matching social network profiles. Specifically, we consider the bag-of-words representation for social network profiles, with each profile being represented by a collection of local descriptors $D = \{d_1, d_2, ...d_k\}$. These descriptors can be name, friends, phone number, city of residence, etc. Additionally, we consider that different descriptors have different weights, represented by a set $W = \{w_1, w_2, ..., w_k\}$. These weights determine the relevance of each descriptor in the model. For example, the name of a sibling or wife has more weight that the city where our claimant lives.

We consider we have one generic profile S that contains all prior knowledge about our claimant. Moreover, we consider that in each social network, we have a set of N candidate profiles. These N candidates can be obtain by searching profiles with matching first name and/or surname of our claimant. Another approach to obtain these candidates could be using known relatives and friends. Once we have a list of candidate profiles, our approach evaluates them, identifying matching words from the generic profile S in such profile candidate. Once all properties are evaluated we quantify the similarity of our candidate using equation 1.

$$\sum_{i}^{k} d_i * w_i * x_i \tag{1}$$

where x_i represents the frequency of each descriptor d_i and w_i is the weight of such descriptor in our model. The similarity value allows us to rank profiles in a descending order, such that higher ranked profiles are more similar to the person we are looking for than lower ranked. Next we use a human-in-the-loop approach to present this information to the investigators, which allows them selecting one of the profiles for each social network of interest. This step could be automatized using machine learning techniques [3].

Once a profile is selected, we can go over the second part of our approach and extract relevant data related to our claimant.

B. Extracting relevant information

Using the Bag-of-words model, we extract keywords from our prior knowledge and claim information. We consider N sets of keywords $K = \{K_1, K2, ..., K_N\}$, where each set K_i has t keywords, i.e. $K_i = \{k_{i1}, k_{i2}, ..., k_{it}\}$. Each set of keywords has a weight for its keywords defined by $\{w_1, w_2, ..., w_N\}$. For example, we could have the following groups of keywords weighted by importance: (i) specific keywords related to the claim under investigation – e.g., in a claim about a physical disability, we could include activities that he/she cannot supposedly do; (ii) keywords extracted from our prior knowledge (e.g., name of insurance company or name of employer); and (iii) generic keywords that are relevant regardless the type of claim – e.g., claim, insurance, bankrupt, suit. These keywords require strong domain knowledge and, in our case, they have been generated by our insurance industry partners.

Furthermore, we have messages $M = \{m_1, m_2, ...m_t\}$ that can contain text, media (e.g., image, video), links to websites, etc. These messages are called differently depending on the social network platform we are using, for example in Facebook they are called *post*, or in Twitter they called *tweets*. In cases like Facebook or Instagram, each message has additional comments on it which may also contain relevant information.

Algorithm 1: Search relevant information

```
 1: for each profile in PROFILES do
 2:     Extract data from social network
 3:     Find messages with matching keywords in the message and/or
        comments
 4:     for each message found do
 5:         Evaluate message relevance using equation 2
 6:         if Message has comments then
 7:             Identify and attach "active" friends to current profile
 8:             if Current profile depth level is <=
                MAX-DEPTH-LEVEL and not friend profile in
                PROFILES then
 9:                 Add friend profile to PROFILES
10:             end if
11:         end if
12:         if Message is submitted from an external social network
            then
13:             Identify and attach external network profile to current
                profile
14:             Add social network profile to PROFILES
15:         end if
16:     end for
17: end for
18: Sort information and visualize
```

Our approach consists on finding messages with relevant keywords across various social networks while keeping provenance. As we describe in Algorithm 1, we have a list called *PROFILES* that serves as an initial list of social network profiles to start our search. This list of profiles can be the result of Section II-A or can be discovered at runtime as we explain later. Each message with matching keywords is evaluated to determine its relevance (line 5). The relevance of a message, including its comments, is defined by equation 2.

$$\sum_{i}^{n} \sum_{j} weight(k_{ij}) * f_i \tag{2}$$

where f_i represents the frequency of a keyword and $weight(k_{ij})$ represents the weight of keyword $k_{ij} \in K_i$.

Additional information is extracted from the message, namely friends and other social network profiles. We create a list of "active" friends, who are people that actively participate in our claimant's live by commenting on relevant messages (lines 6-11). These people may have information about our claimant in their social network profiles and we can automatically expand our search to these profiles. However, we have a variable, $MAX - DEPTH - LEVEL$, that allows us to control how may levels of friendship we want to explore. For example, one level means friends of our claimant and two levels means friends of friends of our claimant.

Oftentimes, social networks allow users to connect their profiles such that they can publish information from one social network in another. For example, one can publish information on Facebook via Twitter. In these cases, there is special metadata identifying the profile of the external social network. In the previous example, a Facebook post has a specific field showing that this information was published from Twitter as well as the associated profile. We leverage this to expand our search to other social networks (lines 12-15). This data-driven approach allows performing a thorough exploration of all publicly available data.

Provenance metadata, from all relevant information, is kept to enable organizing the information in a hierarchical manner that clearly identifies people's social connection and how the data was obtained. Within this hierarchy, we sort all messages of each social network profile using scores obtained in line 5 of Algorithm 1 in a descending order (line 18). Various social network friends are also sorted in such a way that the most "active" ones are shown first at each level of our hierarchy.

After collecting and organizing relevant data, we present this information to the investigators in a structured manner. Investigators will then evaluate the information and decide if there are signals of potential fraud. In this way, investigators have a single point of access to structured and relevant information of various social networks. Note that evaluating the existence of insurance fraud goes beyond the scope of this work, but techniques like the one presented by Subelj et al. [4] could be used to analyze our results.

III. Implementation

We parallelized our approach and implemented an initial prototype on top of CometCloud. CometCloud is an autonomic framework designed to enable highly heterogeneous, dynamically federated computing and data platforms that can support end-to-end application workflows with diverse and dynamic changing requirements [2]. The resulting solution is able to perform large scale exploration of social media using geographically distributed resources driving the exploration by dynamically modifying and creating subworkflows using data insights, as shown in Figure 1. In this solution, the Social Media Fraud Explorer implements our fraud detection approach, which serves as a computational engine. On the other hand, CometCloud is responsible for orchestrating the entire execution allocating and deallocating resources to meet the application needs.

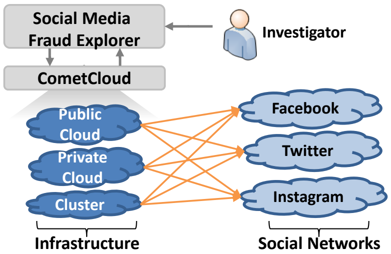


Figure 1: Fraud Detection Framework Architecture

Our solution exposes an API that allows the creation of plugins to interact with different social networks. Currently, our initial prototype supports Facebook, Twitter, and Instagram social networks. This prototype leverages Java Selenium Webdriver [11], Chromedriver [6], and the Xvfb virtual server [14] to extract and parse data from various social network portals. The reason for not using the official APIs is that oftentimes they impose constraints even on public data.

Next we detail how the approach presented in Section II has been implemented using data-driven dynamic ensembles of workflows in CometCloud.

A. *Identifying claimant on social media*

We implemented a dynamic workflow that allows us to autonomously search profiles matching our claimants or subjects under investigation on several social networks and organize those profiles by network and similarity to our subject. In the current implementation this initial search uses the name of our subject (first name and/or surname), name of family and friends, known employers, education, and phone number.

As a result an ensemble of workflows are generated and executed across distributed resources using CometCloud. Figure 2 describes the workflow that implements the approach described on Section II-A. This workflow allows us to concurrently explore different subjects and different social networks. Each white rectangle in Figure 2 represents a workflow that identifies a subject under investigation in a social network.

The first stage of the workflow (S1) filters profiles by searching basic information such as name, last name,

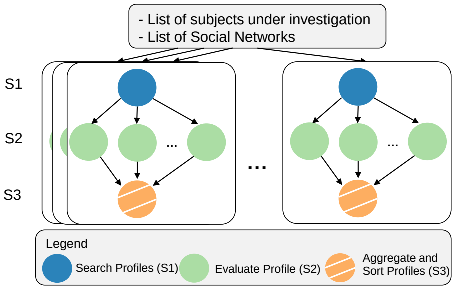

- List of subjects under investigation
- List of Social Networks
S1
S2
S3
...
...
...
Legend
Search Profiles (S1)
Evaluate Profile (S2)
Aggregate and Sort Profiles (S3)

Figure 2: Identify subject on social networks. Each workflow identifies a subject in a social network.

phone number, and email address. This gives us an initial list of profiles that are candidates to be our subject under investigation in a given social network. Next, in second stage (S2) each profile is independently evaluated using claimant's prior knowledge and information extracted from such social network profile. Finally, the last stage (S3) combines all the results and sorts them, placing at the top the profiles that are more similar to our claimant.

B. *Extracting relevant information*

Once we know the social network profiles of our claimants or subjects under investigation, we proceed to search relevant information in their profiles as well as in their social circle. The input file of this workflow is also a JSON file per subject, which includes the unique identifiers of our subject as well as time restrictions and relevant keywords with their respective weights.

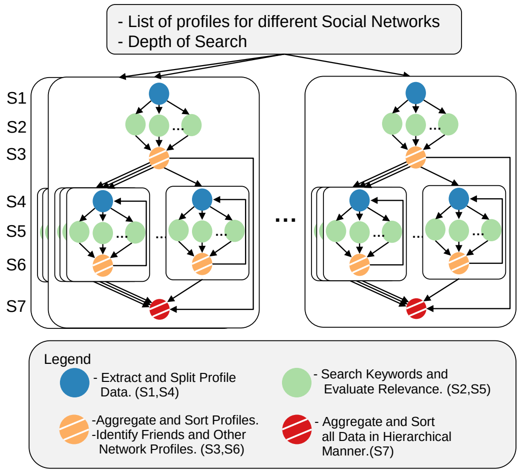

- List of profiles for different Social Networks
- Depth of Search
S1
S2
S3
S4
S5
S6
S7
...
Legend
- Extract and Split Profile Data. (S1,S4)
- Search Keywords and Evaluate Relevance. (S2,S5)
-Aggregate and Sort Profiles.
-Identify Friends and Other Network Profiles. (S3,S6)
- Aggregate and Sort all Data in Hierarchical Manner.(S7)

Figure 3: Search Relevant Data on varios social networks. This workflow shows a search that expands to friends of our subject's profile (depth=1). Each workflow search information of a subject and its social circle.

Figure 3 depicts the workflow or ensemble of workflows describing the implementation of our strategy. Each white rectangle in Figure 3 represents a workflow that search for relevant data on a subject and his/her social circle. First, we perform a search in our subject's profile – stages S1, S2, S3. Stage S1 identifies the amount of data available in the profile, e.g. in Facebook it identifies the number of months, or in Twitter the number of tweets. Next, this information is split to enable parallel processing of all data in S2. The processing of this data involves, searching keywords and evaluating relevance of a message, identifying "active" friends, and looking for other social media profiles. Next S3 collects all information generated in S2 and organizes it. Moreover, if depth of search is higher than one, multiple subworkflows will be created – as many as "active" friends – to perform the same operations over these friends (stages S4, S5, and S6). Note that we included a loop between S4 and S6 to indicate that we can go deeper in the social circle hierarchy, if depth of search is higher than one. If a loop occurs, CometCloud automatically injects new stages and expands the workflow. For example, if depth of search is equal two, then we will have new stages after S6, called S4_1, S5_1, S6_1, such that S6 connects to S4_1 and S6_1 is now connected to S7.

Finally, stage S7 collects all information from S3 and S6 (possibly stages S6_1,...,S6_N, if depth of search is higher than one). The system uses the recorded provenance to organize the data in an architectural way showing relationships and the way all data was obtained. In this way, investigators can access to all this information (text, pictures, links to videos, etc.) and assess if a potential fraud is present.

IV. Evaluation

In this work, we used two different subjects, with different social network activity, to evaluate our solution. One of the subjects has limited activity on social networks, labeled as *Subject-limited*, and the other subject has significantly more activity on social networks, labeled as *Subject-active*.We performed individual investigations to observe how our approach behaves following the insights obtained from the data. Although investigator can choose several social networks to perform the investigation, Facebook has shown to be the most interesting social network as it contains significant personal information. Therefore, in these experiments we used Facebook as a primary source of information. If the system finds Twitter or Instagram information during the data analysis, it would automatically expand the search to those social networks.

Oftentimes big institutions have different data centers distributed across the globe, or they rely in public clouds to complement their own resources. To show our ability of scaling across geographically distributed resources, we deployed our framework in a multi-cloud federation composed by three different geographically distributed clouds. One of the clouds is part of the FutureSystem project [7], located at Indiana University, Indiana and built using the OpenStack framework. The other two clouds are part of the Amazon Web Services (AWS), one is on the "us-west" region (located in Oregon) and the other one is on the "us-east" region (located in Northern Virginia). Note that each region in AWS are completely independent public clouds deployed in different data centers.

Table I: Resources available at each site and their characteristics.

VM type†	#Cores	Memory	Max. VMs‡
India_M1Medium	2	4 GB	8
AWS-East_M3Medium	1	3.75 GB	8
AWS-West_T2Small	1	2 GB	8
AWS-West_T2Medium	2	4 GB	4

Note: † – Name of the site followed by the type of VM.
‡ – Maximum number of available VMs per type

A. *Identify subject*

Given our two subjects under investigation we first identified them in the social network. This was done by simply including two JSON files, one per subject. In our experiments we used the Facebook social network. Table II summarizes the number of tasks generated for this experiment. S1 contains a single task per subject to identify candidates to be our subjects. Next S2 processes all these candidates in parallel (five for one subject and eight for the other) and finally S3 generates again one task per subject to aggregate and sort the results. Figure 4 collects the number of resources provision over time as well as the throughput measured as the number of tasks completed. We observe that the computational demands of this workflow are satisfied with the resources offered by a single cloud. Specifically, this workflow requires up to seven machines (14 cores) to complete in less than seven minutes

Table II: Number of tasks per stage.

Social Network	S1	S2	S3
Facebook	2	14	2

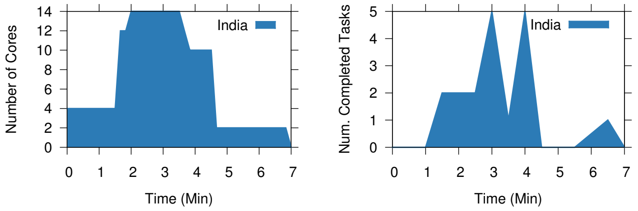


Figure 4: Summary of experimental results of identifying subject on social networks. Left column shows the number of resources provisioned over time to identify our two subjects and right column shows the Throughput measured as the number of tasks completed.

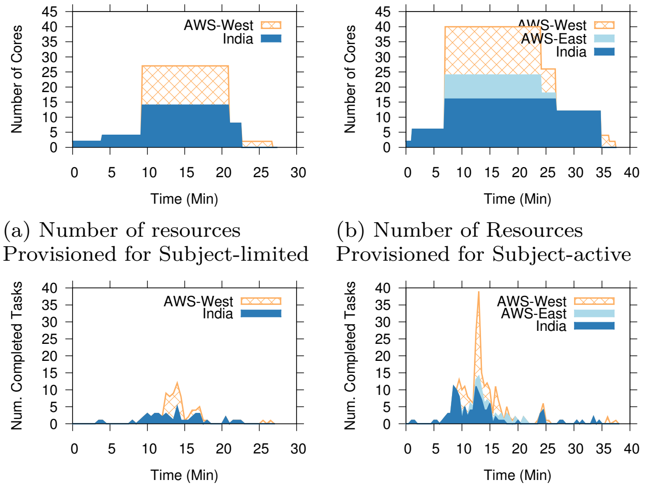
B. *Extracting relevant information*

In this workflow, we expected to have significant differences between our two subjects. Hence, we executed both investigations individually to show how our framework generates different number of tasks depending on insights obtained from the data and consequently allocate resources to satisfy the computational needs. We used a JSON file like the one described in Section III-B for each subject. In both cases we started with the Facebook social network. However, in the case of the Subject-active the system found a Twitter account during the analysis and automatically expanded the search to that social network as well.

Table III collects the number of tasks generated at each stage of the workflow for each one of the two subjects under investigation. Tasks are separated by social network. In both cases the three first states only have Facebook tasks as it is the only social network id provided as input data. However, for the Subject-active a Twitter account was found during the analysis of the Facebook information (stage S2). Specifically, this subject publishes information on Facebook from Twitter and the system was able to obtain the Twitter id from one of those posts. Hence an additional subworkflow was created to extract and analyze information of our subject in Twitter (subworkflow in stages S4, S5, and S6). This demonstrates that our approach is able to adapt to the available data by varying the amount of generated tasks at runtime. More importantly, this is transparent to investigators as it is automatically driven by data analysis at runtime.

Table III: Number of tasks per stage.

Subject-limited							
Social Network	S1	S2	S3	S4	S5	S6	S7
Facebook	1	2	1	30	58	1	1
Twitter	0	0	0	0	0	0	0
Subject-active							
Social Network	S1	S2	S3	S4	S5	S6	S7
Facebook	1	3	1	51	149	1	1
Twitter	0	0	0	1	99	1	0



(a) Number of resources Provisioned for Subject-limited
(b) Number of Resources Provisioned for Subject-active
(c) Throughput Subject-limited
(d) Throughput Subject-active

Figure 5: Summary of experimental results of searching relevant data on social networks. Left column shows Subject-limited and right column shows Subject-active.

Figure 5 summarizes the experimental results. On the left column we have the number of resources used and the number of tasks completed over time for the subject with limited activity on social networks (*Subject-limited*), Figures 5a and 5c. On the right column we have the same information for the subject that is more active (*Subject-active*), Figures 5b and 5d. We can clearly observe how the number of tasks as well as the number of resources significantly increases when there is more publicly available data. Moreover, the number of resources allocated to each workflow varies along the execution adapting to the number of tasks to compute. In the case of Subject-limited, the workflow requires up to 16 machines (27 cores) to satisfy the computational demands. These resources were allocated across two of the clouds as shown in Figure 5a. However, in the case of Subject-active the scheduler requested up to 28 machines, which totaled 40 cores from all three clouds as shown in Figure 5b.

Finally, Figure 6 shows the effectiveness of our solution for extracting relevant information from large amounts of analyzed data. The information is divided in relevant (contains our keywords) and non-relevant (does not contain any of our keywords). We show the number of messages (posts and tweets) that have been analyzed for each subject, labeled as *Posts-subject*. The figure also shows the number of messages of all active friends of each investigated subject (those friends that are tagged, commented, liked or retweeted relevant messages), labeled as *Posts-friends*. The number of friends is also included. We can observe that our analysis extracts relevant information significantly reducing the amount of data that an investigator has to study. Specifically, we found that in our use cases only between 1% and 15% of the analyzed data was relevant to the investigations. Additionally, we were able to identify relevant people in their social circle ("active friends") and extract information from them. In the case of the friends we only analyzed a few dozen from the several thousands of friends that our subjects had.

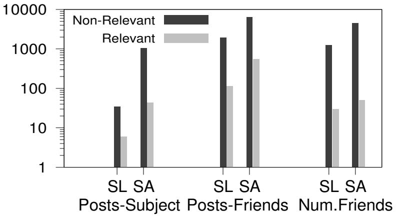


Figure 6: Summary of data analyzed in various social networks discretized by relevant and non-relevant. SL identifies the results of Subject-limited and SA identifies the results of Subject-active.

V. Related Work

Social media investigation is an area that is becoming increasingly popular among law enforcement and insurance companies. However we found very limited publicly available work on providing an end-to-end solution that can extract, aggregate, analyze and organize data from multiple social networks such as the one presented in this work. The closest to our solution is a commercial tool called Web Identity Search Tool (WIST) [13], which is only available to government, and insurance companies. This tool has some interesting visualization to obtain the social circle of a subject. However, it is a desktop tool and it cannot scale beyond a personal computer.

There are social media crawlers that can collect data from various social networks and blogs. These are typically focus on sentiment analysis, such as the services offered by PromptCloud [10], or to identify influence of a subject or brand in the Internet, such as the services offered by Social Crawlytics [12].

VI. Conclusion

In this paper we presented an approach to investigate fraud using social media. This solution is able to empower investigators with data analytics capabilities to explore various social networks from a single point of access. Provenance is kept during the data analysis to present extracted information in a structured way, helping investigators to identify insurance fraud. We performed some experiments and showed how the proposed solution is able to follow data identifying social circle and connections with other social networks. By simply specifying some input parameters, our approach was able to extract and analyze data large amounts of data in a short period of time by transparently scaling beyond institutional boundaries.

As a future work, we would like to explore the use of machine learning techniques to enable more complex searches, leveraging natural language processing, and to allow the system assessing the existence of fraud. Moreover, we would like to collectively analyze multiple claims to find patterns and unveil fraud rings.

Acknowledgments: The research in this work is supported in part by NSF via grants ACI 1339036, ACI 1310283, ACI 1441376.

References

[1] E. Agichtein, C. Castillo, D. Donato, A. Gionis, and G. Mishne. Finding high-quality content in social media. WSDM '08, 2008.

[2] J. Diaz-Montes, M. AbdelBaky, M. Zou, and M. Parashar. Cometcloud: Enabling software-defined federations for end-to-end application workflows. *Internet Computing, IEEE*, 19(1), 2015.

[3] S.-H. Park, S.-Y. Huh, W. Oh, and S. P. Han. A social network-based inference model for validating customer profile data. *MIS Q.*, 36(4):1217–1237, 2012.

[4] L. Subelj, S. Furlan, and M. Bajec. An expert system for detecting automobile insurance fraud using social network analysis. *Expert Systems with Applications*, 38(1), 2011.

[5] Coalition Against Insurance Fraud.http://www.insurancefraud.org.

[6] Chromedriver. https://sites.google.com/a/chromium.org/chromedriver/.

[7] FutureGrid. https://portal.futuregrid.org/.

[8] Y. Zhang, R. Jin, and Z.-H. Zhou. Understanding bag-of-words model: a statistical framework. *International Journal of Machine Learning and Cybernetics*, 1(1-4):43–52, 2010.

[9] Lexis Nexis. http://www.lexisnexis.com/.

[10] PromptCloud. https://www.promptcloud.com.

[11] Java Selenium WebDriver. http://docs.seleniumhq.org/.

[12] Social Crawlytics. https://socialcrawlytics.com/.

[13] Web Identity Search Tool (WIST). http://wist.harmari.com/.

[14] Xvfb. http://www.x.org/archive/x11r7.6/doc/man/man1/xvfb.1.xhtml.