Explore Social Behavior Indicators in Telecom Fraud

Visar Mullafetah

IT4BI Master Thesis Report

Abstract

Fraud in telecommunication networks is causing huge profit losses for telecom operators. One way to try to tackle this is through data analysis. Emergence of intelligent data analysis tools coupled with huge amounts of data being generated is helping to address this complex business problem. Call data, which are time-series call transactions, are analyzed to infer network user's intentions and behavior. Most approaches extract subscriber call behavior from call data and mainly use probabilistic methods, neural networks and rule based systems to differentiate between fraudulent and regular usage. Subscriber call behavior models show their limits on international bypass fraud, where fraudsters use more advanced technologies to mimic basic call behavior. The goal of this work is to explore the call activity from a social behavior point of view and look for discriminatory features. Three different models are created based on subscriber neighborhood activity, importance, and relations with deeper degrees in the network, and Selective Naive Bayes is applied. Six social features show discriminatory capabilities, two of the models show around 0.77 accuracy rate in classification, with relatively high TP and FP rates. The results are promising to continue further work.

Keywords: simbox fraud, data mining, social graphs, neighborhood computation, social tree

Table of Contents

1. Introduction

According to IBM, every day we create 2.5 quintillion bytes of data. The number does not make much sense, but it may give an indication that industries and the scientific community are willing to invest in data, to solve more complex problems, and extract value. This is coupled with the explosion of big data technologies, of which many are open source, and data mining algorithms that perform very well on large amounts of data. In turn, this allows companies and organizations to explore their data at not much high initial cost and see if further investment is valuable. This work has been carried out in Orange Telecom in Lannion in a five month internship period. Data used is real-world call data from Orange affiliate country with anonymized users.

In this work we deal with the problem of fraud, specifically with international bypass fraud in telecommunications networks. Fraud can be defined as the illegal access to the network and the use of its services with no intention to pay service charges or making money by using these services [1] [2]. Telecom fraud causes high revenue losses to telecom operators. Worldwide revenue losses in telecom industry in 2001 were estimated around $2.8 billion, and have risen to $7 billion in 2014[3]. In international bypass fraud, fraudsters bypass the interconnect route using low-cost call termination techniques, whereupon the overseas operator receives no payment. Many operators consider international bypass fraud very complex to tackle in depth[10]. The complexity can be attributed to fraudsters being able to use more sophisticated tools and technologies.

Telecommunications industry generates and stores tremendous amounts of data. This work focuses on call detail data which describes the calls within telecommunication networks, and the source which should reflect subscribers' intentions. The task here is to transform this call data into summary features to learn on models of calling behavior so that these models make inferences about users' intentions. There are several interesting issues when it comes to mine telecom data. First concern is scale, since telecom databases contain billions of records [11]. Second, call detail data in raw format is not suitable for mining, as it is time-series data representing individual events. Before mining, it needs to be transformed into useful summary features. Last but not least, instances of telecom fraud involve predicting very rare events, so rarity is another issue that must be dealt with.

Most of fraud detection applications are intelligent data analysis systems that analyze and summarize subscriber calling activity, and then use rule-based systems, probabilistic models, neural networks or other techniques to find different patterns of behavior that discriminate between regular and fraudster instances. Subscriber calling activity features can be defined as a set of summary features of outgoing and incoming calls of a subscriber during an activity period, independently from other subscribers in the network. Network operators are exploring complementary or alternative approaches to detect more fraudsters. This work focuses on looking at subscribers' calling activity from a social point of view. We construct a social network and generate social features based on the calling activity of subscriber's direct neighborhood, importance of subscriber in the network and its local communities, and prior links with its direct neighbors before their first call. We apply Selective Naive Bayes to evaluate the features and learn models that can discriminate between regular and fraudster instances.

The structure of the document is as follows: Section 2 presents international bypass fraud, Section 3 summarizes related work, Section 4 presents approach in the work, Section 5 nature

of source call data, Section 6 social indicator exploration, and Section 7 and 8 learning models and evaluation.

2. International bypass fraud

International bypass fraud uses several low-cost call termination techniques such as SIMBOX, to bypass the legal interconnection and divert the international calls with VoIP or satellite gateway into local calls, thus avoiding payment of the interconnect toll termination fee, illustrated in Figure 1. The call is diverted by the interconnect partner who utilize SIMBOX devices which can hold up to 300 SIM cards that belong to various local mobile operators, and are used to replace the originating phone number. Bypass Fraud is generally frequent in developing countries that lack transparent telecom regulatory mechanisms, and rule of law, causing to have less reliable interconnect operators. The key motivation for fraudsters (individual or organization [3]) is the difference in calling rates, targeting the countries where there is a big difference between the national calling rates and international terminating rates [3], due to charges being paid upon call termination, ensuring good profit margins. The scenario requires fraudsters to have access to advanced technology [1].

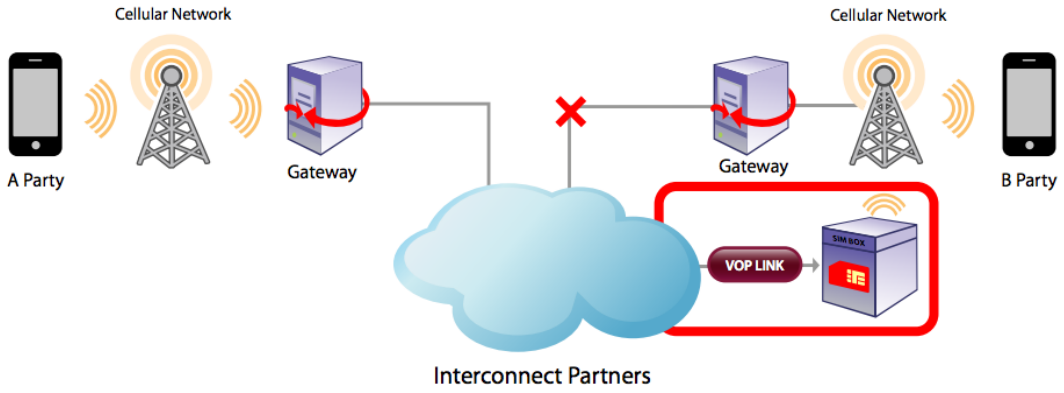


Figure 1. International Bypass Fraud. When Party B receives the international incoming call it appears as a local operator number which is used by SIMBOX to divert the call.

Fraudsters tend to use destination party's operators network to route the call, referred to as ONNET call/bypass, ensures the maximal charge profit margins for fraudulent parties. ONNET, calls within the same operators network, are expected to provide the least national calling rates.

According to a white paper published by Africa BroadBand Group on the topic of "Fraud in Mobile Services in Africa" in 2012, bypass fraud remains the largest problem to locate and eliminate. Study is based on responses of 11 mobile operators, out of 48 polled, in sub-Saharan Africa with more than 20% of the operators suspecting its use in their networks[10].

3. Related Work

There is no fixed set of deterministic rules that allow to identify a subscriber as a fraudster. The focus is on detection methodologies based on calling activity. One common method for identifying fraud is to build a profile of customer's calling behavior and compare recent activity against this behavior. Hollmen (2000) divides the detection methodologies into two categories: absolute analysis and differential analysis, illustrated in Figure 2. He states that in absolute analysis detection is based on the calling activity of normal behavior and fraudulent behavior, whereas the differential analysis approaches the problem by detecting sudden changes in behavior, by raising alarms on deviations from established patterns of usage. Hollmen concludes that in both cases, analysis methods are usually implemented using probabilistic models, neural networks or rule-based systems.

Figure 2. Absolute analysis and differential analysis, the two main approaches to fraud detection, are illustrated using a probabilistic view. In absolute analysis, illustrated left, models of both normal (C0) and fraudulent behavior (C1) must be formulated. In differential analysis, one model is built assuming normal behavior (C0) and any deviations from the established behavior are classified as fraudulent. [2]

Taniguchi, Hollmen et al. (2000) use three different methods for fraud detection. First, they propose a feedforward neural network which learns a discriminative function to classify subscribers with summary features such as maximum, average and standard deviation on the duration and number of calls made during a day. Second, they describe a probability density estimation using a Gaussian mixture model of subscribers' past behavior and then compute the probability density of the current usage. They used the daily number of calls and the length of the calls occurring during the office hours, the evening hours and the night hours as summary features. Third, they construct two Bayesian networks to describe the subscriber behavior. The first behavioral model is based on the assumption that the subscriber is a fraudster, while the second assumes the subscriber to be a legitimate user. By inserting evidence in these networks, the observed user behavior *x* derived from call data, they get a probability of the measurement of *x* under both assumptions. Their experiments using simulated call records show that their models detect around 85% of fraudsters. Performance assessment of their models is done using Receiver-Operating Characteristic (ROC) curve, as Ogwueleka (2010) and Aranuwa (2013) do in their work. Aranuwa uses both absolute and differential methods on subscriber call features summarized on various times of the day such as business time (8am-5pm), evening calls (5pm-9pm), and night period (after 10pm) at which they claim to find the majority of fraudsters.

Since fraudulent usage is rare compared to all usage, and number of detected fraudsters is relatively low, fraud detection applications have to deal predicting very rare events, with highly skewed class distribution making data mining algorithms prone to bias. Weiss (2003) and Ezawa (1995) increase the proportion of fraudulent cases in the training set to have a more balanced distribution. Ezawa states that even after increasing fraud cases from 1-2% to 9-12% the

systems have difficulties characterizing the minority class, whereas Weiss raises the issue of the use of a non-representative training set that can be problematic because it does not provide the data mining method with accurate information about the true class distribution.

We propose to create calling activity summary features from a social network point of view exploring the neighborhood, local communities and relations between subscribers. We use Khiops, which is an internal data mining tool in Orange, to evaluate the features and its Selective Naive Bayes which learn classification models. We use ROC curve to assess model performance and analyze predictor importance of features.

4. Fraud Detection

Fraud detection is referred to as a try to detect illegal usage of a communication network [2].Our approach is to explore the subscriber calling activity from a social point of view, and evaluate if summarized social features can help discriminate fraudulent usage. Data is represented in social graphs, and knowledge exploration is performed in three feature representations: 1. The call activity within the neighboring nodes of each subscriber, 2. The importance of subscriber in the social graph, and its local communities 3. The shortest distances between a subscriber and his direct neighbors before their first communication. We look if there exists a social tree structure. The exploration phase results with three representations of features. Feature, and model evaluation is performed by Khiops via its Selective Naive Bayes.

The Selective Naive Bayes (SNB) predictor performs a feature selection using a greedy heuristic. The selection consists in successive forward and backward passes, and is repeated several times. The Selective Naive Bayes predictor is then averaged among all the evaluated models. The weighting scheme on the models reduces to a weighting scheme on the features. The feature weights are reported in the evaluation report. The overall training time is $O(NK\log(NK))$ where N is the number of instances and K the number of features. [13]

5. Call Data

This section includes description of call data used in this work, data columns chosen to explore, and a previously detected list of fraudsters that will be used in the training set.

Call data represents call records that are transactions or events ordered in time, referred to as Call Detail Records (CDR). Each transaction in CDR can be of one of the following types: voice call, SMS, or mobile data. Voice and SMS transactions include two subscribers, consisting of the caller and the called. Mobile data transactions are associated with only one subscriber.We perform analysis on voice and SMS transactions only. For simplicity, both a voice call and an SMS are referred to as a call. In our approach mobile data is limited to a descriptive binary feature (use or not use) so we ignore it to reduce the data size, to process in time with the set of available resources.

Table 1 shows a summary of the available CDR for this work, real data from an Orange affiliate country. Subscribers in the network are represented by their MSISDN, the number uniquely identifying a subscription in a GSM network. Subscriber MSISDN is anonymized. A CDR contains two columns for identifying subscribers in a transaction: 1. Calling MSISDN, representing the originating or outgoing call, 2. Called MSISDN, representing the subscriber receiving the call. From the social network point of view we can look at each record from an outgoing or incoming call perspective.

Orange Affiliate Country CDR				
Number of subscribers	CDR per day	Total Data Size (compressed)	Data start/end day	Number of Detected Fraudsters
~3 million	~10 million records	81GB	12/10/2015 - 29/02/2016	12544

Table1. Summary of available call detail records from Orange affiliate country, and detected fraudster list

For each record the time of the originating call is saved, together with the duration of the call. SMS transactions do not have any duration, having a default value of 0.
Full CDR data consists of a set of tab-delimited compressed files, where one file contains generated transactions for one day, for the time period between October 12th, 2015 and February 28th 2016, specified in Table 1. The following data fields are used to generate summary features, which should include sufficient information to describe the important characteristics of each call:

1. transaction type (string) - record type (VOICE, SMS, DATA)
2. record type (string) – transaction direction; incoming or outgoing call (VOICE_IN, VOICE_OUT, SMS_IN, SMS_OUT)
3. timestamp (timestamp) – date and time of call origin
4. duration (integer) – duration of the call in seconds (default value for SMS)
5. calling MSISDN (long integer) – MSISDN of subscriber making outgoing call (calling party)
6. called MSISDN (long integer) - MSISDN of subscriber receiving call (called party)

Call detail records in raw format represent information in individual phone calls. Since the goal of data mining applications is to extract knowledge at the subscriber level, call detail records associated with a subscriber must be summarized in order to describe the calling behavior into a single record. Furthermore, this work aims to explore social behavior of subscribers and see if there are informative summary features that would more in-depth information about social structures.

The rule-based and probabilistic system in Orange has detected 12,544 fraudsters, using its data mining tool Khiops. This list of fraudsters is used for the classification task. The list of classified and verified subscribers as fraudsters is 0.0003 of all subscribers.

We represent CDR data as a social graph, where each node is a unique subscriber (MSISDN), and an edge is a link between two nodes representing a single record in CDR. Different graph algorithms such as PageRank[18], neighborhood aggregation, community detection are applied to generate models with social features. The following sections explain the knowledge exploration, analysis, and evaluation phases of the work.

6. Social Behavior Exploration

The analytics process is carried out in 7 phases, starting with analysis of the business problem, illustrated in Figure 3. Phase 2 and 3 represent the date range selection of call data to be processed and data import to Spark. Phase 4 performs horizontal and vertical slicing of CDR, keeping relevant data. Phase 5 builds the social graph in GraphX, and runs graph-parallel neighborhood computation, Page Rank, TriangleCount, Shortest Path algorithms to generate social indicator features. In phase 6 and 7 Khiops applies its Selective Naive Bayes to create learning models and generates four evaluation reports where we assess feature relevance and model performance. Tools used to perform this analysis process are Apache Spark for large-scale data processing, GraphX library for graph processing, and Khiops for data mining.

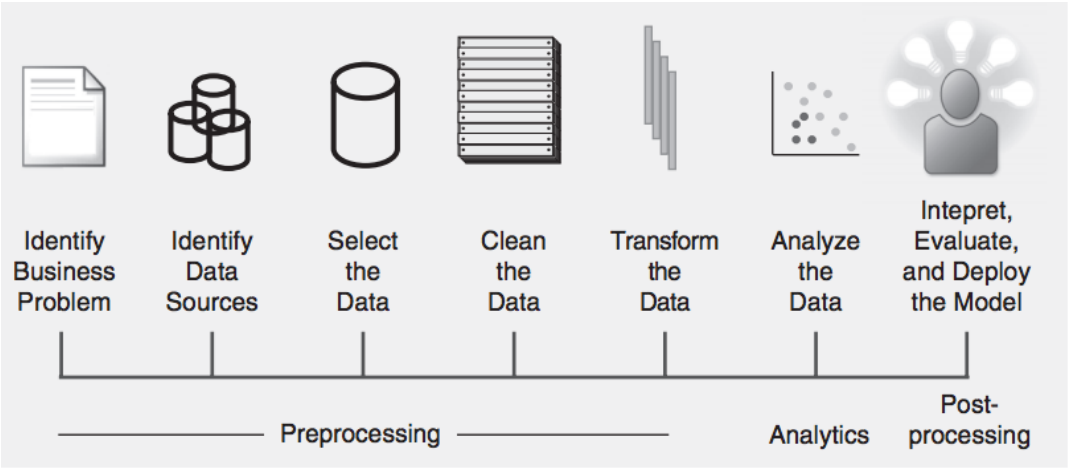


Figure 3. Exploration, analysis process model (change this figure). This process has been performed iteratively, to compare results with various data samples, various date ranges, and to re-evaluate features and models.

Social features are extracted based on calling activity of a subscriber, the calling activity of its neighboring nodes, and relation of a subscriber with other nodes in the graph.

We have chosen absolute analysis (see Section 3, Related Work) as detection methodology, by processing data in snapshots of time and formulate both normal and fraudulent social behavior models. It was observed that processing only one or two days of data does not provide any informative distinction of any irregular usage. Many experiments show us that we need to process at least 10 days of CDR to generate distinguishable patterns of behavior. This is partly due to different calling activity periods of subscribers, where longer periods of time can show more stable social activity. The majority of experiments were carried out using 20 days of CDR data. The data selection is the initial phase of the exploration processing.

Before constructing the social graph we perform data cleaning and slicing operations to reduce the data size and keep relevant records. Vertical slicing operations keep 6 relevant data columns out of 28, listed in section 5, whereas horizontal slicing filters out the irrelevant records such as: network subscribers which are devices (antennas, other devices), all subscribers that have more than 50 outgoing or incoming calls/day (telemarketing, other), mobile data and call forwarding records.

Resources available for computing include a SPARK shell on a 4-node Hadoop Yarn cluster over HDFS, and a remote server with 32 processing cores, and 100GB of memory. Indicator exploration is performed on the remote server where a SPARK standalone cluster is deployed.The SPARK shell is used to perform ad-hoc queries and transformations on data.

Section 6.1 introduces graph data representation of CDR and section 6.2 explains graph algorithms used to extract social features.

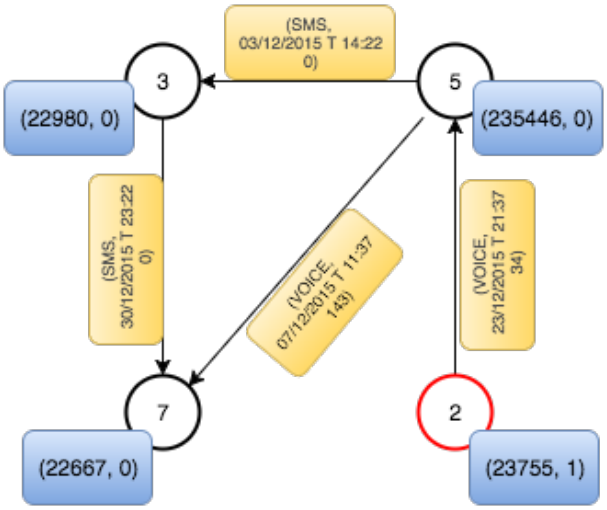
6.1 Graph Data Representation

GraphX is a Spark component for graphs and graph-parallel computation. At a high level, GraphX extends the Spark Resilient Distributed Dataset (RDD), a fault-tolerant collection of elements that can be operated on in parallel, by introducing a new graph abstraction: a directed multigraph with properties attached to each vertex and edge. To support graph computation, GraphX exposes a set of fundamental operators (e.g., subgraph, joinVertices, and aggregateMessages) as well as an optimized variant of the Pregel API[12].

6.1.1 Property Graph

A graph in GraphX, is defined to be a directed property graph that is parameterized over the vertex (VD) and edge (ED) property types denoted Graph[VD, ED]. Each vertex has a set of properties and is associated to a key in the form of a unique 64-bit long identifier (VertexID). Each edge consists of the source vertex identifier, the destination vertex identifier and some properties (Figure 4).

Vertices V represent the subscribers, whereas edges E directed links between them. A vertex v_i is denoted as `[vID, (VD)]` where `vID` is the key and `VD` is a set of two properties: the subscriber MSISDN, and the subscriber status - fraudster or not fraudster. An edge e_i is



id	VD (MSISDN, statuts)
3	(22980,0)
7	(22887, 0)
5	(235446, 0)
2	(23755, 1)

srcID	dstID	ED (type, time, duration)
3	7	(SMS, 30/12/2015 T 23:22, 0)
5	3	(SMS, 03/12/2015 T 14:22, 0)
2	5	(VOICE, 23/12/2015 T 12:37, 34)
5	7	(VOICE, 07/12/2015 T 11:37, 143)

Figure 4. CDR in property graph form. Table (top) shows vertex properties, table (bottom) edge properties. Vertex 2 is a fraudster (flag 1). Duration in seconds.

denoted as `[vsrc, vdst, (ED)]`, `ED` containing three properties: 1. transaction type (VOICE, SMS), 2. timestamp, 3. call duration. Vertices and edges form the social network.

Table 2 shows social network construction statistics in size and computation time.

Graph is kept in memory, and the scripts that generate the social indicator features are able to process up to 20 days of data, reaching memory limits. Spark, by default, allocates a fraction of 0.6 of the total memory to the Java heap to use for Spark's memory cache.

CDR snapshot	compressed source data size (GB)	graph data size before partitioning (GB)	graph data size after partitioning (GB)	computation time (min)
1 day	~0.7	~1.6	~1	~ 1
10 days	~7	~14	~10	~ 5
20 days	~14	20-25	15-18	~ 8

Table 2. Graph data size, computation time. Data slicing operations help reduce data size to compute, by keeping more relevant transactions. It also reduces computation time.

Social network size also depends on different snapshots taken. For example for 20 day snapshots, in November there is less communication with 3M subscribers present, where it peaks in January with 6M subscribers (including OFFNET numbers). Experiments were carried out with various time snapshots, where more data showed better results. Final experiments are carried out with snapshots of the first 20 days of each month in CDR, described in more detail in

Section 7.1. Working on different time snapshots is important to try cover more representation of calling activity, and try to find larger sets of fraudsters, as the proportion of fraudsters to regular vertices is very low, as shown in Table 3

CDR snapshot	number of vertices	number of edges	number of fraudster vertices
1 day	~1 million	2-4 million	~ 300
10 days	2-4 million	20-40 million	~ 1300
20 days	3-6 million	70-90 million	~ 2100

Table 3. Social network basic statistics.

6.1.2 Neighborhood

We have created three social feature representations. Each representation tries to describe some social indicators by expressing each social feature quantitatively, through aggregating communication data on different degrees of the graph.

1. Direct Neighborhood Representation (DNR) - features on the communication activity within the direct neighborhood of the subscriber.
2. Importance and Local Community Representation (ICLR) - features describing the importance of the subscriber within the social graph, and local communities.
3. Distance Representation (DR) - features on prior relationships of subscribers with their direct neighbors.

A key step in generating social indicators is aggregating information about the neighborhood of each vertex. Neighborhood computation is performed in all feature representations. We consider two vertices v_i, v_j to have a direct link, or be direct neighbors, if there is at least one edge e_i between them such as e_i is [v_{i_src}, v_j dst, (ED)] or [v_{j_src}, v_i dst, (ED)]. So, the neighborhood of a vertex v in the graph G[VD, ED] is the induced subgraph of G'[VD, ED] consisting of all vertices adjacent to v. In directed graphs we consider the out-neighborhood and the in-neighborhood of a vertex. That is, the out-neighborhood, $N+(v_i)$ is the set of vertices to which v_i has an edge, and the in-neighborhood $N-(v_i)$ is the set of vertices that have an edge to v_i. The DNR Representation extracts features from the out-neighborhood of a vertex.

aggregateMessages is a core neighborhood aggregation operation in GraphX, and provides an efficient neighborhood computation mechanism. It aggregates values from the neighboring edges and vertices of each vertex. The user-supplied sendMsg function is invoked on each edge of the graph, generating 0 or more messages to be sent to either the source or the destination

vertex in the edge. The `mergeMsg` function is then used to combine all messages destined to the same vertex. [12]

The following section describes each feature representation in detail, by summarizing its features and the extraction method.

6.2 Social Indicators

6.2.1 Direct Neighborhood Representation

Direct Neighborhood Representation (DNR) represents the communication activity of the direct neighborhood of the subscriber instance. The aim here is, to explore for each target user the calling behavior of its neighborhood by computing number of calls, number of distinct links, duration of calls, and ratio of voice over sms generates summary features.

Let's take v_t as the target vertex. The set of direct out-neighborhood vertices v_j from v_t have the distance `d=1`, and is denoted as $N_1^+(v_t)$. We extract the neighborhood of v_t which is an

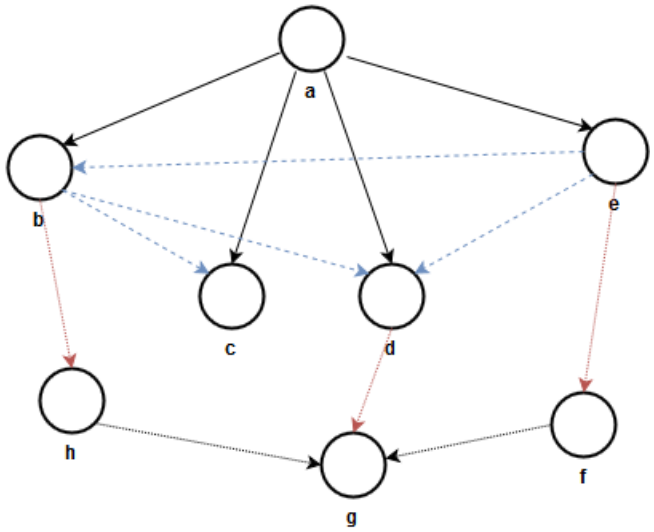


Figure 5. Subgraph `g` representing neighborhoods of vertex `a`. Direct, or `deg1` neighbors are all outgoing nodes one hop away from `a` denoted $N_1+(a)$ and consist of {`b,c,d,e`}. `deg2` neighbors are two hops away from `a` denoted $N_2+(a)$ and consist of {`f,g,h`}. In Direct Neighborhood Representation generated features are based solely on communication between $N_1+(a)$ nodes {`b,c,d,e`}, excluding `a`.

induced graph `G'[VD, ED]` where VD is the distinct set $N_1^+(v_0)$ vertices, and ED is the set of `ei` where `vj_src` and `vj_dst` of `ei` are a subset of $N_1^+(v_0)$, illustrated in Figure 5. `G'` does not include v_0.

For each vertex v_j in $N_1^+(v_0)$ within `G'` we generate the following call behavior features:

- v_j SPREAD - # of distinct vertices v_j called within G'
- v_j calls - # total outgoing calls v_j made within G'
- v_j duration - SUM, AVG of duration of outgoing calls within G'
- v_j type ratio - ratio of VOICE calls over SMS

The above generated features for each v_j in G' need to be summarized to have a single value for each feature type for the target vertex v_0. Target vertices make up the final feature matrix used for data mining. We summarize these features in two ways: 1. via basic arithmetic operations such as mean, sum for all features types, and 2. creating value groupings for SPREAD and calls features to get a better representation of calling activity. The final set of

Type 1 Features	Type 2 Features	
SUM ($N_1+(V)$ SPREAD)	G[1-3] ($N_1+(V)$ SPREAD)	# of subscribers making 1-3 calls
SUM ($N_1+(V)$ calls)	G[1-3] ($N_1+(V)$ calls)	
SUM ($N_1+(V)$ duration)	G[4-8] ($N_1+(V)$ SPREAD)	# of subscribers making 4-8 calls
AVG ($N_1+(V)$ SPREAD)	G[4-8] ($N_1+(V)$ calls)	
AVG ($N_1+(V)$ calls)	G[9-15] ($N_1+(V)$ SPREAD)	# of subscribers making 8-12 calls
AVG ($N_1+(V)$ duration)	G[9-15] ($N_1+(V)$ SPREAD)	
AVG ($N_1+(V)$ type ratio)	G[>15] ($N_1+(V)$ calls)	# of subscribers making more than 12 calls
	G[>15] ($N_1+(V)$ calls)	

Table 4. Direct Neighborhood Representation Type 1 and Type 2 features.

features are presented in Table 4. Type 2 features were generated after it was observed that Type 1 features do not provide any information to discriminate between fraudster and not fraudster classes.

The nature of the exploration task requires producing the models in a timely manner, in order to evaluate, reflect, and continue the next iteration of exploration. It is crucial to have optimized algorithms that perform well on graph-parallel processing tasks. To obtain DNM features we worked on two different approaches, using GraphX transformations. First, `subgraph` transformation, which restricts the graph to only vertices and edges satisfying the user defined predicates. This would give us the induced neighborhood graph G'. As predicate to slice the graph we provide the out-neighborhood vertices $N_1+(v_i)$ for each vertex v_i. Result of this operations is a tuple of (v_i, G_i'[VD, ED]). Map reduce transformations to G' generate the final features. The problem with this approach is the huge number of subgraphs that need to be generated (one per user), which becomes very time consuming, and after a certain point runs out of memory.

Second approach is to use `aggregateMessages` aggregation method explained in section 6.1.2. Before running aggregateMessages, we collect $N_1+(v_i)$, $N_1-(v_i)$ for each vertex v_t and broadcast to all nodes in cluster. `aggregateMessages` operates on each edge of the graph, sending computed messages to either source or destination vertex of the edge, and merging those messages for each vertex. Source vertex of edge e_i is defined as v_{e_src} and destination vertex as v_{e_dst}. Neighborhood extraction is performed in sendMsg function which has the following steps:

- collect $N_1-(v_{e\_src})$
- for each $v_j \in N_1-(v_{e\_src})$
  - collect $N_1+(v_j)$
  - if $v_{e\_dst} \in N_1+(v_j)$ then send message (v_j, v_{e_src}, v_{e_dst}) to v_{e_src}

`mergeMsg` then combines all messages at v_{e_src} to complete the process. Further map reduce operations transform data to type 1 and type 2 features, Table 4.

Table 5 shows computation capacity and time for both methods.

CDR snapshot	# of vertex to process `subgraph` method	computation time `subgraph` method	# of vertex to process `aggregateMessages` method	computation time `aggregateMessages` method
1 day	100k	2h	~1 million	< 1 min
10 days	50k	4h	2-4 million	3 min
20 days	X	X	3-6 million	5 min

Table 5. Computation time for DNM features using `subgraph` and `aggregateMessages` methods.

6.2.2 Importance and Local Community Representation

GraphX includes a set of graph-parallel efficient algorithms to simplify analytics tasks. We leverage two main graph operations, which produce the features of our Importance and Local Community Representation (ILCM): PageRank, and TriangleCount.

PageRank

PageRank measures the importance of each vertex in a graph, assuming an edge from vertex *u* to vertex *v* represents an endorsement of *v*'s importance by *u,* (Figure 6). For example, if a user is called by many others, the user will be ranked highly. PageRank algorithm recursively defines the rank of a vertex v:

$$\boldsymbol{Pr}(v) = 0.15 + 0.85 \sum_{u \text{ links to } v} w_{u,v} \times \boldsymbol{Pr}(u)$$

in terms of the weighted $w_{U,V}$ ranks *Pr(u)* of the vertices u that link to v [17].

In the static implementation the PageRank algorithm runs for a fixed number of iterations, whereas in the dynamic implementation PageRank uses a convergence tolerance (the smaller, the more accurate) and iterates until the ranks converge. Experimentally we verified that Dynamic PageRank with convergence tolerance of 0.001 turns out to give the best performance. We compute three social features with PageRank for each vertex v: 1. PageRank with all communication, 2. PageRank with voice calls only, 3. PageRank with SMS messages only (Table 6). Our experiments show that the first two features have discriminatory capabilities (Figure 8, Section 8.1.1), whereas SMS page rank is not an informative variable.

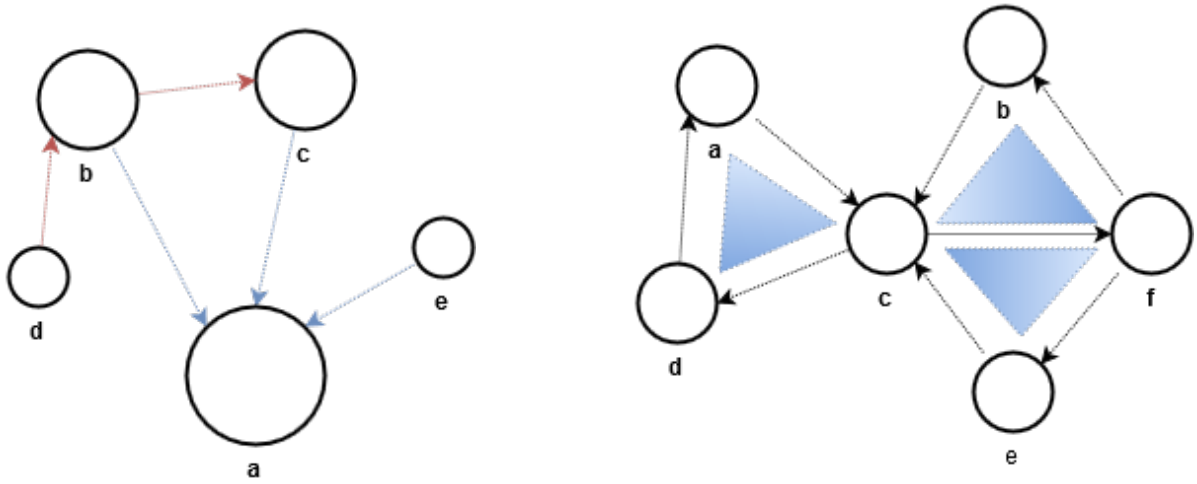


Figure 6. Subgraph g representing Page Rank (left) vertices with more incoming links have a higher ranking. Subgraph g' representing Triangle Count (right) number of triangles passing through each vertex.

Triangle Count

TriangleCount computes the number of triangles (e.g., cycles with three edges) that pass through each vertex. A vertex is part of a triangle when it has two adjacent vertices with an edge between them (Figure 6). Vertices that form the local community(triangle) are a subset of the direct neighborhood vertices of target vertex. TriangleCount computes the number these local communities of a subscriber.

The algorithm is computed in three steps:

1. Collect the set of neighbors for each vertex N(v).
2. For each edge e=(v_1, v_2), compute the intersection of N(v_1) and N(v_2). The size of the intersection gives the number of triangles passing through v_1 and v_2 and having e as an edge. The number of triangles is then updated for v_1 and v_2.
3. Compute the sum at each vertex and divide by 2 (since each triangle is counted twice)

Similar to PageRank we compute TriangleCount features for all communications, voice calls,

Importance and Local Community Representation features	
PageRank	Subscriber importance coefficient
PageRankVoice	Subscriber importance coefficient on Voice calls
PageRankSMS	Subscriber importance coefficient on SMS messages
TriangleCount	Number of local communities

Table 6. Importance and Local Community Representation features.

and SMS separately. In our experiments, we noted that voice and SMS features are not enough to differentiate between classes, which lead to their omission from the model.

6.2.3 Distance Representation

Distance Representation features look at target subscriber' shortest distance prior links with to their direct neighbors before their first call. We assume regular subscribers in the social network should have a social tree structure where each deeper level in the tree represents a neighborhood of closeness. Our intuition is that this social tree may not be the same for fraudsters.

We define the n-distance neighborhood P_n of a target vertex v_t as the set of vertices such that the shortest path between these vertices and v_t has exactly n edges, where $n \in \{1,2,3,4,5,6,7\}$. Therefore, P_1 is the set of vertices that are directly linked to v_t, P_2 is the set of vertices that are directly linked to at least one vertex in P_1 and so on. P_n for incoming links is denoted as P_n^-, whereas for outgoing links P_n^+. From a social point of view and from v_t's perspective we say that v_t is more closely related with 1-distance neighbors than 2-distance neighbors. This comparison assumption is applied to each increment of n, up to $n=7$. From a social perspective we consider that subscribers whose distance is more than 7 edges away would not have any relation whatsoever, where P_7 is the set of furthest vertices v_t can have a relation with. We consider this a simple social tree structure where v_t is the root node and each P_n makes a level in the tree. We look into shortest path link from v_t to each of its incoming neighbors (P_1^-) prior to their first communication. More specifically, we extract the incoming links P_1^- of v_t, and for each vertex v_j in P_1^- we find the shortest distance n from v_t disregarding their direct link. We say that v_t has a shortest prior relationship R_n with v_j with shortest distance n denoted $v_t R_n v_j$. This is the shortest distance between subscribers A and B, before B makes the first call to A. The intuition here is, does the subscriber know the caller at some level before

receiving the first call? The expectation is to see a difference between regular and fraudster calling patterns in any of the levels is this defined social tree. The final set of features of distance representation consists of the number of subscribers $v_t R_n v_j$ for each n-distance, shown in Table 7.

Distance Representation features	
D1	# of neighbors with no prior relationship
D2	# of neighbors with prior relationship of shortest distance two
D3	# of neighbors with prior relationship of shortest distance three
D4	# of neighbors with prior relationship of shortest distance four
D5	# of neighbors with prior relationship of shortest distance five
D6	# of neighbors with prior relationship of shortest distance six
D7	# of neighbors with prior relationship of shortest distance seven

Table 7. Distance Representation social features.

To find $v_t R_n v_j$ for each v_j in P_1 and for all n we extend the Pregel ShortestPaths implementation of GraphX to include the dimension for prior relationship R_n where $v_t R_n v_j$ = {0=no prior relationship,1=has prior relationship}. The algorithm takes two inputs: 1. A graph G - a 20 day snapshot of the social graph, 2. A set of target vertices v_t for which the n shortest distance prior relationship operation $v_t R_n v_j$ will be computed, for $n \in \{2,3,4,5,6,7\}$. The algorithm is implemented using the Pregel API. Pregel is a synchronous message passing graph-parallel abstraction in which a user-defined program $vprog$ is run in all source vertices v_j in G concurrently in a sequence of super-steps. In each of the steps $v_t R_n v_j$ is computed. A message m holds the information about the prior relationships between all v_j and v_t with their shortest distance. Super-step 0 initializes all v_j states to type m, and flags the target vertices v_t. Within each super-step the $vprog$ instance in v_j receives the messages sent by its neighbors in the previous super-step, computes the shortest path from v_t and checks if $v_t R_n v_j$ is found, mutates state of m in v_j, and then sends the next round of messages to its out-neighbors in the next super-step. Super-step 1 performs the process for P_1 , super-step 2 for P_2, and super-step n for P_n. The algorithm terminates when $n > 7$ or there are no more messages to send.

7. Khiops and General Model Evaluation

The following section describes performance of learning models, along with evaluation of informative features. It is important to mention the goal in this section is more to understand how the set of generated features perform than to have strong learning models.

7.1 Khiops

Khiops is internal data preparation and scoring tool of Orange for both supervised and unsupervised learning. It allows performing univariate and bivariate descriptive statistics, to evaluate the predictive importance of explanatory features, to discretize numerical features, to group the values of categorical features, to recode input data according to these discretization and value groupings. Khiops produces a scoring model for supervised learning tasks, according to a Selective Naive Bayes approach[13].

Khiops distinguishes between 4 feature types:

1. Native - initial representation features (Table 8)
2. Constructed - features derived from initial representation.
3. Evaluated - trained features.
4. Informative - features that can discriminate between instances.

Data dictionary is built by specifying the choice of feature types (all features are numerical, except the class), and creation of derivate variables using derivation rule language. 70/30 percentage split is used to create all training and test sets. Khiops evaluates the predictive importance of the numerical explanatory features to analyze their predictive values.

Khiops outputs 4 evaluation reports, which contain statistics of individual features and evaluation of the models consisting of training and test reports. Feature evaluation is performed only for informative features.

7.2 Models

The previous phase produced 3 separate feature representations: Direct Neighborhood Representation (DNR), Importance and Local Community Representation (ILCR), Distance Representation (DR). These representations are formed after many iterations of exploration with various time snapshots. We call these final feature representations (Table 8). The full set of features consists of 28 datasets: 4 DNR, 12 ILRC, 12 DR each of them representing social summary features of subscriber instances for 20-day calling data snapshots. An instance

	Sample 1 (Jan)		Sample 2 (Jan)		Sample 1 (Nov)		Sample 2 (Nov)	
	U	F	U	F	U	F	U	F
ILCR	5565	1455	5363	1448	4089	1306	5943	1351
DR	5575	1447	5370	1447	4069	1328	5965	1333

Table 8. Class distribution of for two(out of three) samples extracted for January 2016 and November 2015. Unknown (U) Fraudster (F). December and February have very similar representations, whereas November has ~130 less F instances with also more U instances.

represents one subscriber in the network. Each instance belongs to a binary class attribute of Fraudster (F) - instances previously detected as fraudsters, and Unknown (U) - instances not detected as fraudster. The number of fraudster instances is ~0.0004 of all instances. When we apply Selective Naive Bayes on feature sets with more than 1 million U instances (all subscribers in the network in a time snapshot), and 1200-2000 F instances (all detected fraudsters in a time snapshot) we do not get any informative variables, where all instances are classified as class U. To have a more balanced set we take a random sample for class U instances and keep all fraudster instances. After sampling the range of all instances is between 6,000-9,000 with class balance ~75%-25% with class Unknown majority. To have a better representation of U instances we get 3 samples of data for dates between 1 and 20 for November 2015, December 2015, January 2016, and February 2016. This creates 12 datasets for ILRC, and DR feature representations (Table 9). We summarize evaluation outputs of individual features and learning models for all datasets in Section 8.

DNR Feature Native Feature Set	ILCR Feature Native Feature Set	DR Native Feature Set
OutSpread	PageRank (PGR)	InSpread
OutCalls	PageRank Voice (PGR_VOICE)	D1
G[1-3] (N_1+(V) SPREAD)	Triangle Count (TRN)	D2
G[1-3] (N_1+(V) calls)		D3
G[4-8] (N_1+(V) SPREAD)		
G[4-8] (N_1+(V) calls)		
G[9-15] (N_1+(V) SPREAD)		
G[9-15] (N_1+(V) SPREAD)		
G[>15] (N_1+(V) calls)		
G[>15] (N_1+(V) calls)		

Table 9. Final feature sets of native DNR, ILRC, DR features.

Type 1 features (Table 4) in DNR did not give any information whatsoever and are not included. Two other subscriber level features are created: OutSpread - number of outgoing distinct links, OutCalls - total number of outgoing calls. They act as a very simple profile of subscriber's outgoing calling activity. These subscriber level features are used to create new derivate features which relate subscriber with its DNR social features (Table 9). Selective Naive Bayes evaluates the derivate features just as not derived ones. Though, neither original nor derivate features of DNR provide satisfactory result. All of DNR features are non-informative. Neighborhood calling activity patterns of fraudsters do not differ enough from regular users' patterns. This does not coincide with the intuition: *in the neighborhood of regular subscribers there should be some identifiable calling activity patterns of close groups such as family, close*

friends that have a more regular calling pattern between themselves than subscribers in the neighborhoods of fraudsters. The intuition should still have ground unless fraudsters mimic this behavior. Though, the quality of generated features of DNR model must be improved. We generate features on a 20-day snapshot of calling acitivty. During this 20-day snapshot subscriber presence in the network, and its calling activities may not well be comparable. One subscriber can be in the network from day 16, another other from day 3, each having different activity periods. Though, the learning algorithm treats all instances as equal. We propose an evolutionary approach in which each subscriber starts at day zero representing their first ever incoming or outgoing call in the network, then generate cumulative features for each consecutive day. The first set of features will be for day one, and each consecutive day will contain calling activity since day one. This way we can have a feature set as all subscribers that have been present in the network for 2, 5, 10, 15, 20, 30 days. This allows to see the evolution of behavior over time, and make new inferences. The approach also needs a more complete user profile. The user profile can include more in-depth calling activity features on daily calling activity: active hours during day, number of active periods, hibernation(inactivity) periods, sleep hours, and weekly calling activity: business days, weekends. User profile can also include localization features such as frequently calling and called locations, number of distinct calling locations. The evolutionary approach can be implemented in a streaming context and can serve as all feature representations.

We run each of these feature representations in Khiops, apply Selective Naive Bayes predictor, and evaluate the outputs.

Models on Importance and Local Community, and Distance features show discriminatory capabilities. Section 8 presents results obtained, and final inferences on these social features.

8. Evaluation of Results

Section 8.1 evaluates ILRC and DR features individually using metrics provided from Khiops. Section 8.2 evaluates learning models on ILRC and DR features using Accuracy, ROC, and Confusion Matrices.

8.1 Feature evaluation

We evaluate the individual features with the following metrics:

Level - Evaluation of the predictive importance of the features. The Level is a value between 0 (features without predictive interest) and 1 (features with optimal predictive importance). Features with level values over 0.1 are considered Level is the primary evaluation criteria for individual features.

Weight - evaluation of the predictive importance of the feature taken relatively to all the features used by the predictor.

Intervals - Number of intervals resulting from the discretization and grouping preprocessing of the feature. We analyze the coverage and distribution of instances that belong two both positive and negative classes, within these intervals.

Feature summary statistics are shown only if the features are informative.

8.1.1 Importance and local community features

This section analyzes the results of outputs for ILRC features.ILRC has three final features: TriangleCount (TRN), PageRank(PGR), PageRank Voice (PGR_VOICE), formally defined in section 6.1.2. It does not have any derived features. ILRC features perform relatively well, where each of TRN, PGR, PGR_VOICE show discriminatory capabilities

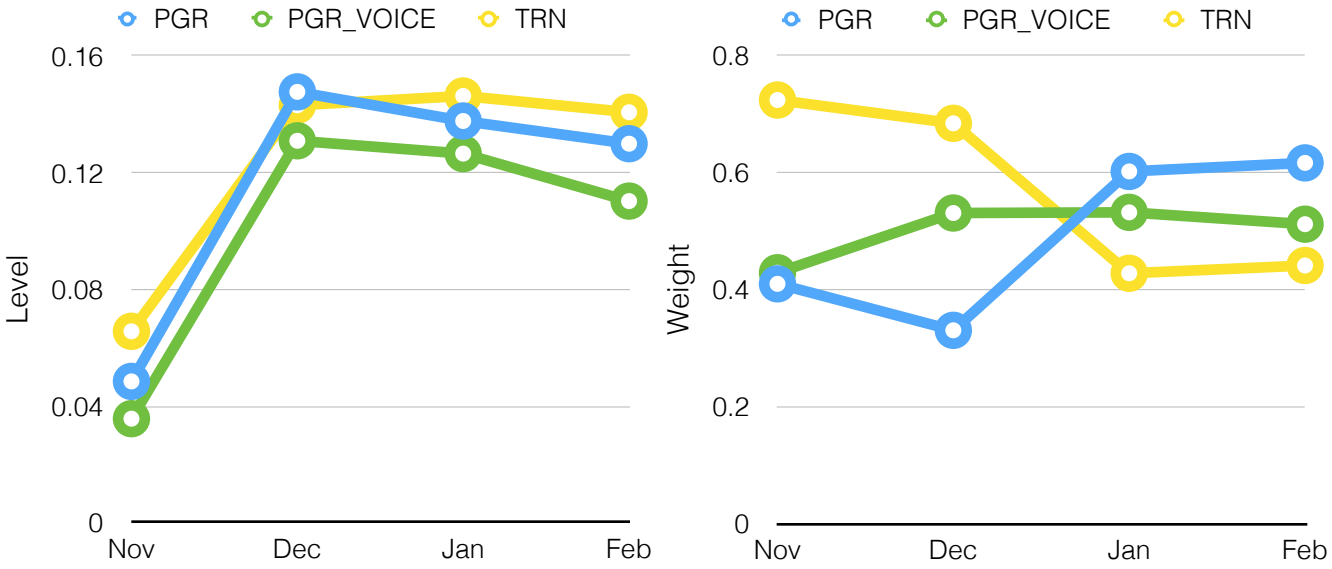


Figure 7. PageRan, PageRank Voice, TriangleCount evaluation for Nov, Dec, Jan, Feb. Feature levels on the left, weights on the right.

Figure 7 (left) shows the predictive importance of PGR, PGR_VOICE, TRN in each month. November has the least number of fraudsters compared to other months which reflects in the lowest level values of features in this month, and relatively higher in other months. TRN and PGR have stronger predictive importance than PGR_VOICE. PGR has the highest predictive importance in December with 0.148, PGR_VOICE in December with 0.131, and TRN in January with 0.146.

Figure on the right shows the weights of features for each month. We can see a different pattern in weights than that of levels. January has the most balanced weights with PGR the highest with 0.6, whereas in November TRN's weight peaks with 0.723 and PGR, PGR_VOICE are at ~0.41. PGR in December has the lowest weight, increasing in January and February. PGR_VOICE has the least fluctuations between months.

Khiops [13] states a feature (even with slight predictive interest and thus a small level) that is used by many good predictors can have a high weight. On the opposite, a feature with a high level can have a small weight in case of redundant variables: the weight is shared by the redundant variables. The results are in line with this claim.

Each selected feature for Selective Naive Bayes predictor is described by two main criteria: Level and Weights.

Next we evaluate features individually. We pick an output sample form January 2016. January contains more calling activity compared to other months with more fraudster instances. Class 1 refers to Fraudster instances.

TriangleCount (TRN) - number of triangles

Khiops created three intervals for TRN. Each of these intervals represents instances that fall under a value range of TRN. Interval 1 (I1) represents zero local communities and contains 66% of all instances, Interval 2 (I2) represents 1 to 3 local communities containing 14% of instances, Interval 2 (I3) represents more than 4 local communities (up to 886) containing 20% of instances Table 10.

	Interval 1	Interval 2	Interval 3
TRN interval ranges	0	1-3	4-886
Number of instances	4695	987	1338

Table 10. Distribution of instances in TRN feature intervals for January 2016

In Figure 8 we can see that only 9% of fraudsters have no local communities (I1), whereas 37% have one to three local communities, and 48% have more than three. This is not the case for regular users. Majority with 68% have no local communities, 19% more than three, and only 11% one to three local communities.

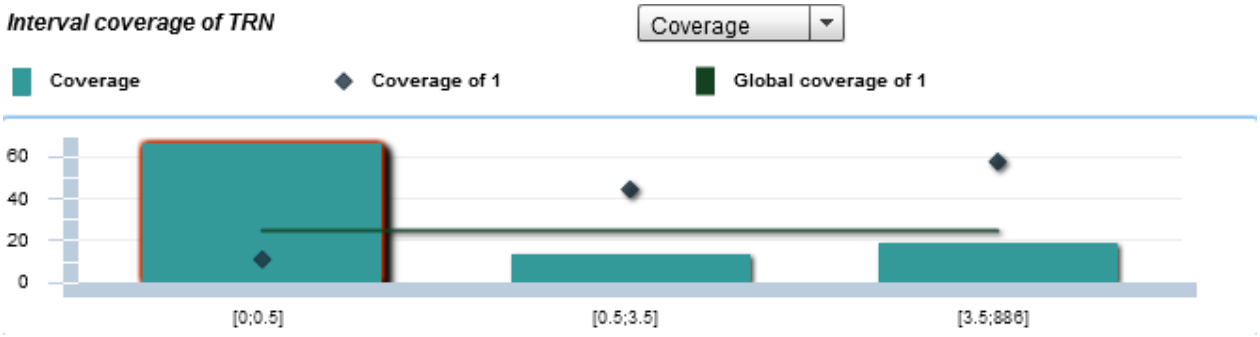


Figure 8. Coverage intervals for TRN with class U(green bar) and class F(black dot).

From these statistics we can infer that fraudsters tend to have more local communities than regular users. This is interesting to see. It can be attributed to fraudsters having a higher calling activity than regular users.

PageRank (PGR) - Importance in the network

Importance coefficient in the network is measured by incoming links a subscriber receives (Section 6.2.2). Minimum value of PGR is 0.15. PGR features have three intervals. I1 represents users with lowest importance in the network containing 44% of all instances, I2 22% of instances who have a higher importance than I1 instances though with a very low margin of increase, and

	Interval 1 low importance	Interval 2 (medium low importance)	Interval 3 (medium low to high importance)
PGR interval ranges	0.15-0.162	0.162-0.198	0.198 - 2.75
Number of instances	3158	1551	2311

Table 11. Distribution of instances in PGR feature intervals for January 2016

the rest of 32% reside in I3 which has a very high interval range compared to I1 and I2. Here we can see that interval ranges differ in size (Table 11).

In Figure 9 we see similar instance coverages as with local communities. A small minority of fraudster belong to I1, with majority in I3 (higher importance). On the other hand 47% of unknown instances have the least importance in the graph. Though there is a relatively high number of unknown instances (37%) in I3 high importance interval.

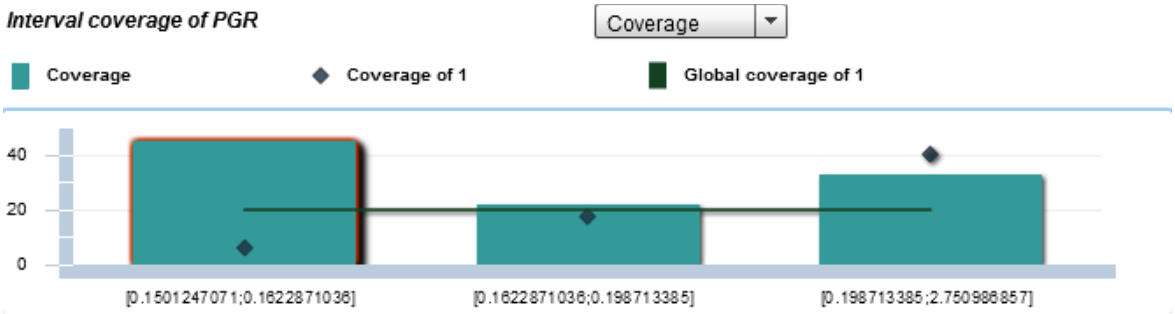


Figure 9. Coverage intervals for PGR with class U(green bar) and class F(black dot).

We can infer that majority of fraudsters do not have a low importance coefficient in the social network, and PageRank does not to distinguish well between fraudsters and regular users who have high importance coefficient.

PageRank Voice (PGR_VOICE) - Importance in the network considering only voice calls

Page Rank for voice calls feature provides five distinct intervals. Though, we can see that I1, I2, I3 contain instances with very low importance. Compared to PGR, PGR_VOICE has two intervals for value ranges between 0.167-2.661.

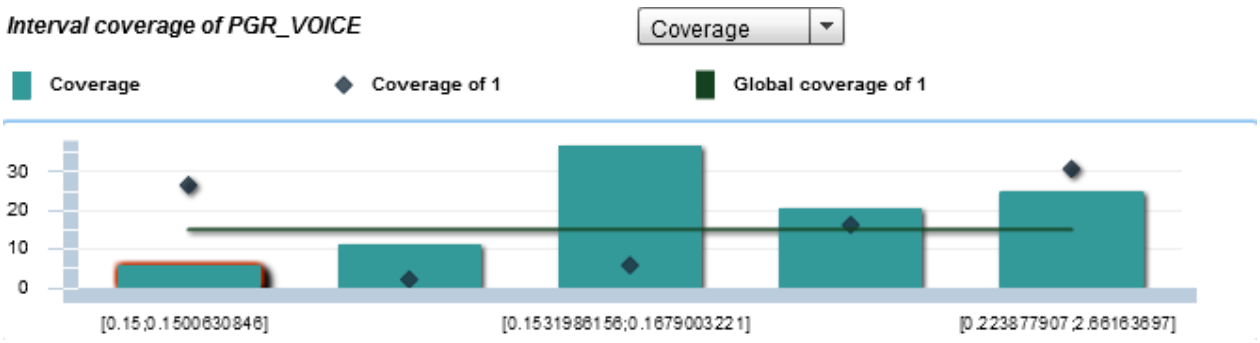


Figure 10. Coverage intervals for PGR_VOICE with class U(green bar) and class F(black dot).

Different to PGR, in PGR_VOICE 28% of fraudsters are in I1 with lowest importance coefficient. Higher importance intervals show a more balanced distribution.

	Interval 1	Interval 2	Interval 3	Interval 4	Interval 5
PGR_VOICE interval ranges	0.15	0.150–0.153	0.153 - 0.167	0.167-0.228	0.228-2.661
Number of instances	412	802	2311	1461	1751

Table 12. Distribution of instances for PGR_VOICE feature intervals for January 2016

8.1.2 Evaluation of distance features

This section evaluates final Distance Representation features. DR features consist of both native and derived features (Table 13).Native features represent number of neighbors with with prior relationship of shortest distance two and three, and number of neighbors with no prior relationship (formally defined in Section 6.1.3). Prior relationships of distances 4-7 (Table 7) are not considered as first experiments showed that they are not informative. Derived features contain the ratios of subscriber's prior relationships of his neighborhood to its number of incoming distinct link. A derived feature of a subscriber would answer to the questions: with how many of his/her neighboring subscribers he/she has a prior relation with for a single unique(subscriber) call he receives.

Feature	Description	Formula	Type
InSpread	number of unique incoming calls		Native
D1	# of no prior relationship neighbors		Native
D2	# of neighbors with prior relationship of shortest distance two		Native
D3	# of neighbors with prior relationship of shortest distance three		Native
PD1	D1 for each incoming unique(subscriber) call	D1/InSpread	Derived
PD2	D2 for each incoming unique(subscriber) call	D2/InSpread	Derived
PD3	D3 for each incoming unique(subscriber) call	D3/InSpread	Derived
DD1	D1 over all neighbors with prior relationship	D1 / SUM(D1, D2, D3)	Derived
DD2	D2 over all neighbors with prior relationship	D2 / SUM(D1, D2, D3)	Derived
DD3	D3 over all neighbors with prior relationship	D2 / SUM(D1, D2, D3)	Derived

Table 13. Native and derived features of DR. Final feature set.

Figure 11 shows predictive importance (level) of five selected DR features. Number neighbors with prior relationship of shortest distance 3, and its derivate features (D3, PD3, DD3) show good discriminatory capability. Number of incoming links alone shows has highest level. Features of neighbors with no prior relationships (D1, PD1, DD1) have low prediction importance. Similar to ILRC features all features have the lowest level values in November.

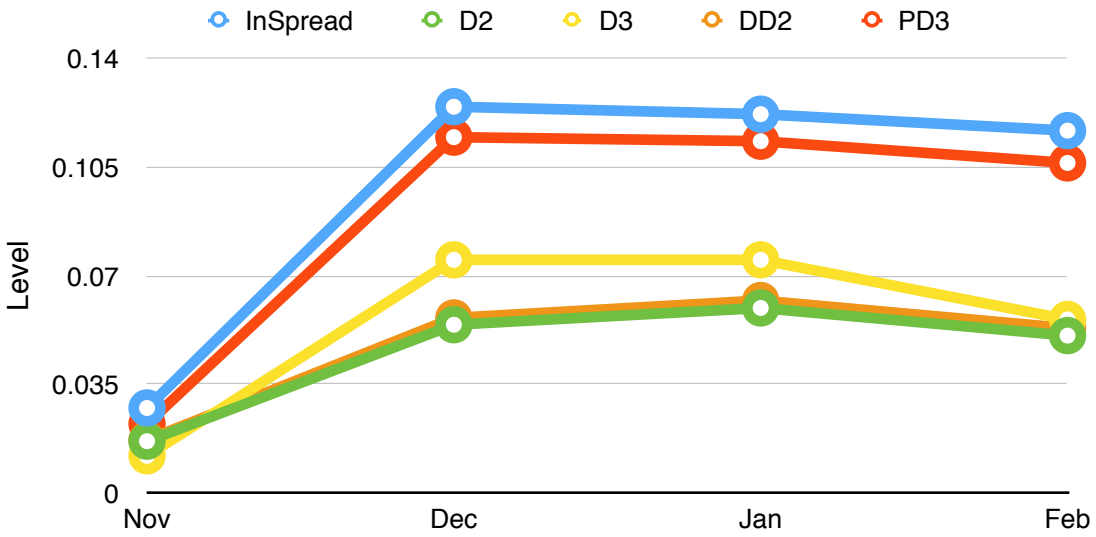


Figure 11. Level of five selected FR features: InSpread, D3, PD3, D2, DD2

Next we analyze PD3, and DD2 in more detail.

PD3 – Number of 3-distance neighbors with prior relationship per distinct incoming call

PD3 feature shows interesting results. It has evenly distributed 4 intervals of coverage, with Interval 4 showing a good distinction between regular subscribers and fraudsters (Figure 12). Table 14 describers the intervals.

	Description	Range
Interval 1	almost with none of his/her neighbors there is a prior relationship in distance 3, in relation to all his neighbors	0-0.05
Interval 2	with up to 37% of his/her neighbors there is a prior relationship in distance 3, in relation to all his neighbors	0.05 - 0.37
Interval 3	with 37%-84% of his/her neighbors there is a prior relationship in distance 3, in relation to all his neighbors	0.37 - 0.845
Interval 4	almost with all of his/her neighbors there is a prior relationship in distance 3, in relation to all his neighbors	0.0845 - 1

Table 14. Coverage intervals of PD3 feature with descriptions from target subscriber perspective.

Most of fraudsters have prior links with up to 37% of its neighbors in shortest distance 3, whereas this number is very high for regular users with above 85%. This validates the initial

assumption: regular users should have a more representative social social tree structure with each level in the tree representing a level of closeness. Fraudsters in principle should not have a

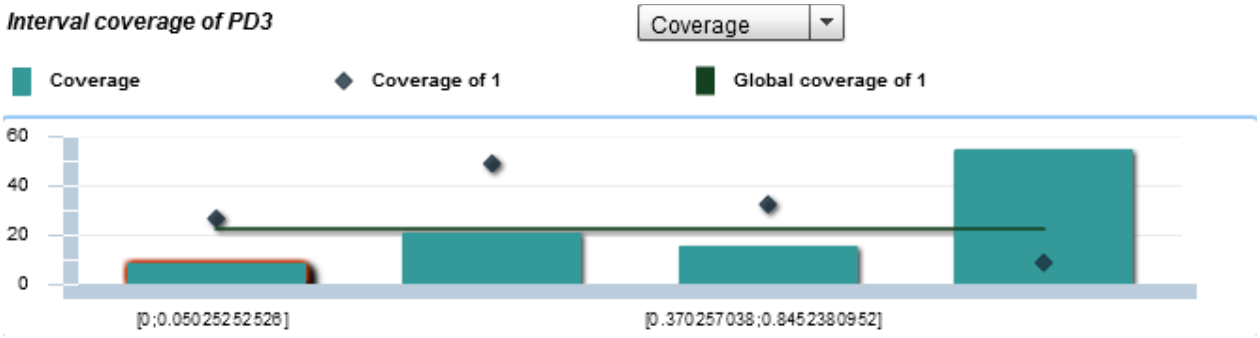


Figure 12. Coverage intervals of PD3 feature.

representative a social tree structure above its direct neighborhood. Here we can see that 57% of regular subscribers already have a link with a neighbor in distance 3, before becoming direct neighbors. In fraudsters this value is at 5%.

8.2 Evaluation of models

The evaluation is carried on two types of models: Importance and Local Community Representation model, Distance Representation model. Results represent basic summary statistics of 12 feature sets for ILCR model and 12 for DR model, which include various samples from November, December, January, and February. ILCR models the proportion of correct prediction (accuracy) in range 0.76-0.81, and area under the ROC curve measures between 0.67- 0.78 (Figure 13).

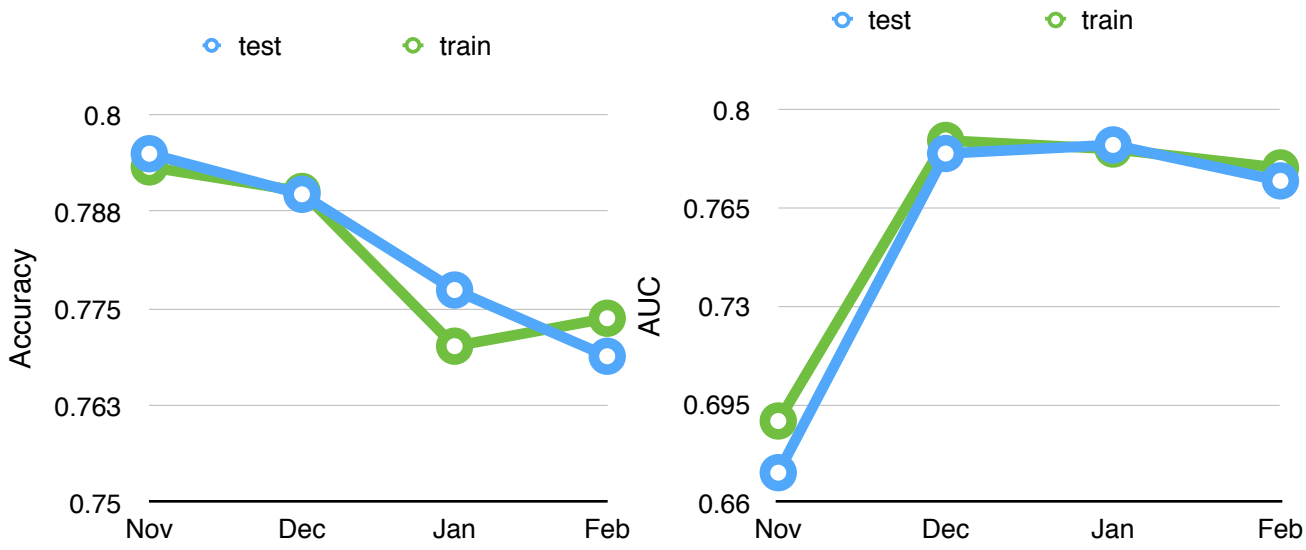


Figure 13. Accuracy (left), Area under the ROC curve(right) metrics for ILRC train and test models.

It is important to notice the highest accuracy is in November which also has the lowest AUC value. The lower discriminatory capability in November can also be seen in section 8.1.1 and 8.1.2 when we evaluate ILRC and DR features, which have the lowest discriminatory capability

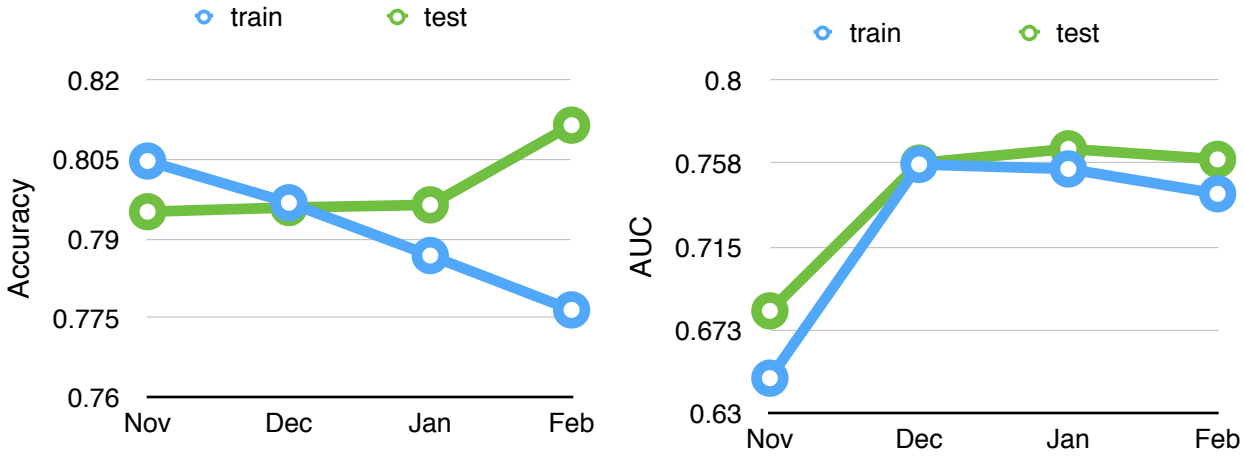


Figure 14. Accuracy (left), Area under the ROC curve(right) metrics for DR train and test models.

in November snapshot. Number of fraudster instances present in the network in November snapshot is less than in the other months. Models on December have the highest discriminatory capability with 0.789, and January and February very close. Similar results are obtained with DR models (Figure 14), with the exception of models in February that have both highest accuracy (0.812), and AUC rates (0.76).

	TP	FN	FP	TN	Precision	Recall
ILRC	4731	761	834	694	0.8501347708	0.86143481427531
DR	4967	893	608	554	0.8909417040	0.847610921501706

Table 15. Confusion Matrix for ILRC and DC Models.

Both ILRC and DR classifiers have a high FP rate (Table 15). This means there is a high number of regular subscribers classified as fraudsters. In fraud detection, a false negative error is usually more costly than a false positive error. Classifiers have high precision and recall rates, showing good performance in identifying class Unknown (U) instances . Though, in fraud detection the goal is to minimize false positives (number of regular subscribers identified as fraudster).

8. Future Work

A crucial improvement to this work is to introduce the time dimension by keeping track of the evolution of subscriber calling activity since he joins the social network on day one, and generate features when a time threshold is met (every 24h). This would allow to establish daily activity patterns by keeping track of active, hibernate, and sleep periods, and weekly activity patterns:

during business hours, weekend. This would create a more real and in-depth calling activity profile. The goal would be to create established patterns of behavior and raise flags on changes of behavior then to re-evaluate to update profile or report as alarm. The evolutionary approach would provide more comparable instances.

Another improvement would be to enrich the social structure properties with localization and closeness. Each call record has the location(antenna) of the caller. Generated features would include aggregation on city, state, country level. Look at frequently calling and called locations, number of distinct calling locations. Local communities feature is very simple with TriangleCount. We can create more representative communities by identifying closely connected nodes looking at their amount of communication, and localization. We can create a more closely related social network by keeping only bidirectional communication between subscriber.

9. Conclusion

In this work we transformed call data into a social network in a time snapshot. We evaluated calling patterns of subscribers from a social point of view. We observed the communication within each subscribers' neighborhood and realized with our current feature representation there is not enough difference between neighborhood calling patterns of fraudsters and regular users. When we observe the importance of a subscriber in the social network (by its incoming links) we see that fraudsters tend to have higher values. Here we can see a fraudsters have more communication then regular users. This seems reasonable as for each terminating call fraudsters earn money. The high communication of fraudsters with their neighborhood is also seen in the number higher local communities(triangles) compared to regular users.When we look at previous shortest links of a subscriber with its direct neighborhood (shortest link they have before their first call), we observe that majority of regular subscriber have a link at distance three, whereas less than 9% of fraudsters have this pattern. This encourages to create a social tree of subscribers and analyze the behavior in more detail in each level of the tree.

The results and observations encourage to continue the work on social representation of calling activity with approaches introduced in future work section. The approaches are also intended to lower the rates of false positives, which for the moment are high.

References

[1] Marah, Elrajubi, Abouda. Fraud Detection in International Calls Using Fuzzy Logic. Misurata University.

[2] Hollmén, Jaakko. User profiling and classification for fraud detection in mobile communications networks. Helsinki University of Technology, 2000.

[3] Subex Inc. White Paper Bypass Fraud. http://www.subex.com/pdf/bypass-fraud.pdf

[4] Aranuwa FO (2013). Hybridized intelligent data analysis model for fraud detection in mobile communication networks. Acad. J. Sci. Res. 1(5): 082-089.

[5] Ann Ratanamahatana, Gunopolus Feature Selection for the Naïve Bayesian Classifier using detection trees.

[6] Taniguchi, Haft, Hollmén , Tresp. Fraud detection in communications network using neural network and probabilistic methods.

[7] Phua, Lee, Smith. A Comprehensive Survey of Data Mining-based Fraud Detection Research. School of Business Systems, Faculty of Information Technology, Monash University.

[8] Bhowmik. Data Mining Techniques in Fraud Detection. Department of Computer Science Sam Houston State University.

[9] Mathieu Langlais, Orange.

[10] Africa BroadBand Group. Mobile/Cellular Services Frauds in Africa. A Revenue Assurance White Paper Whitepaper. 2012. http://www.4gafrica.co.za/wp-content/uploads/2012/07/Revenue-Assurance-White-Paper-BH081.pdf

[11] Weiss. Data Mining in Telecommunications. Department of Computer and Information Science Fordham University

[12] GraphX Documentation. http://spark.apache.org/docs/1.5.0/graphx-programming-guide.html

[13] Khiops guide 8.0, Nov 2014

[14] Weiss, G. M., Provost, F. Learning when training data are costly: The effect of class distribution on tree induction. Journal of Artificial Intelligence Research 2003; 19:315- 354.

[15] Ogwueleka FN (2010). Fraud Detection in Mobile Communications Using Rule-Based and Neural Network System. The IUP J. Sci. Technol. 6(4):21-34.

[16] Ezawa, K., Norton, S. Knowledge discovery in telecommunication services data using Bayesian network models. Proceedings of the First International Conference on Knowledge Discovery and Data Mining; 1995 August 20-21. Montreal Canada. AAAI Press: Menlo Park, CA, 1995.

[17] Xin, Gonzalez, Franklin, Stoica. GraphX: A Resilient Distributed Graph System on Spark

[18] Brin, Sergey, and Lawrence Page. "Reprint of: The anatomy of a large-scale hypertextual web search engine." Computer networks 56.18 (2012): 3825-3833.

Abbreviations

CDR – Call Detail Records

MSISDN - Mobile Subscriber Integrated Services for Digital Network Number

SNB - Selective Naive Bayes

DNR - Direct Neighbor Representation

ILCR - Importance and Local Community Representation

DR - Distance Representation

D1 - number of no prior relationship neighbors

D2 - number of neighbors with prior relationship of shortest distance two

D3 - number of neighbors with prior relationship of shortest distance three