Universidade do Minho
Escola de Engenharia

Eugénio Mendes Rosas

Network Associations – Prevenção de Fraude nas Telecomunicações

Outubro de 2009

Eugénio Mendes Rosas **Network Associations – Prevenção de Fraude nas Telecomunicações**

UMinho | 2009

Universidade do Minho
Escola de Engenharia

Eugénio Mendes Rosas

Network Associations – Prevenção de Fraude nas Telecomunicações

Tese de Mestrado em Informática

Trabalho efectuado sob a orientação do
Professor Doutor Cesar Analide

Outubro de 2009

Agradecimentos

Em primeiro lugar quero expressar a minha gratidão ao meu supervisor, o Professor Doutor Cesar Analide, pelo seu empenho e dedicação na realização desta tese. Sem a sua experiência, paciência e cuidadosa orientação este trabalho não seria possível.

Quero também agradecer à WeDo Technologies por ter facultado meios, recursos e tempo para a realização da tese. Este apoio por parte empresa onde trabalho foi vital para o sucesso desta tese.

Finalmente quero agradecer à minha família, amigos e à Rafaela pelo vosso apoio. Conciliar uma tese de mestrado com um emprego a tempo inteiro requer muita disponibilidade, vontade, e perseverança. O vosso apoio e motivação foram fundamentais para mim nos últimos meses.

A todos, o meu profundo agradecimento.

Resumo

Os operadores de telecomunicações enfrentam um grande desafio: a fraude nas telecomunicações é um facto que os operadores não consideram como um risco, mas como uma certeza e impossível de erradicar. A fraude tem um impacto nos operadores a vários níveis: financeiro, legal, recursos, imagem, relação com os clientes. Para combater a fraude os operadores implementaram sistemas de gestão de fraude, comummente designados de Fraud Management System (FMS).

A presente tese tem como objectivos: (1) estudar, analisar e detectar oportunidades de evolução das soluções de gestão de fraude actualmente disponíveis no mercado, (2) concretizar as oportunidades de evolução da solução, definindo um novo modelo de detecção de fraude e (3) implementar um protótipo.

Da análise de várias soluções de FMS resultou a identificação de duas oportunidades de evolução: (1) implementação de métodos orientados à detecção por identidade e (2) implementação de métodos de extracção de conhecimento. O novo modelo que implementa estas funcionalidades foi desenvolvido usando como suporte um Sistema Multi-Agente em que os seus agentes utilizam técnicas como o *profiling*, análise de redes sociais e extracção de conhecimento. Por fim, foi desenvolvido um protótipo, sobre o qual foram efectuados testes de *performance* qualitativa e quantitativa de forma a poder obter indicações acerca da solução definida para a nova abordagem ao problema.

Abstract

Telecommunication operators face a huge challenge: telecommunication fraud is not a fact that operators consider a risk, it is certain and impossible to eradicate. Fraud has an impact in telecommunication operators at several levels: financial, legal resources, image and customer relations. In order to fight fraud, operators implemented Fraud Management System (FMS).

This thesis has the following objectives: (1) study, analyze and detect opportunities of development for the fraud management solutions available in the market, (2) realize the opportunities of development of the solution, defining a new model for fraud detection and (3) implement a prototype.

The analysis of several FMS solutions resulted in the detection of two opportunities: (1) implement methods oriented to the detection by identity and (2) implement methods oriented to the knowledge discovery. The new model that implements these functionalities was developed using as support a Multi Agent System, where the agents use techniques such as profiling, social network analysis and knowledge discovery. In the last phase, a prototype was developed and performance tests were made in order to obtain information about the solution set for the new approach to the problem.

Lista de acrónimos

ARS Análise de Redes Sociais

CDR Call Detail Record

CFCA Communications Fraud Control Association

CRM Customer Relationship Management

DCBD Descoberta de Conhecimento em Base de Dados

DM Data Mining

EC Extracção de Conhecimento

FMS Fraud Management System

GSM Global System for Mobile communications

HNR Hidden Recharge Numbers

IA Inteligência Artificial

IAD Inteligência Artificial Distribuída

IMEI International Mobile Equipment Identity

IMSI International Mobile Subscriber Identity

KDD Knowledge Discovery in Database

PABX Private Automatic Branch Exchange

PRS Premium Rate Services

SIM Subscriber Identity Module

SMA Sistema Multi-Agente

SMS Short Message Service

SMSC Short Message Service Center

TUFF Telecommunications UK Fraud Forum

UMTS Universal Mobile Telecommunications System

VoIP Voice over Internet Protocol

XML eXtensible Markup Language

Conteúdo

Lista de Figuras

Lista de Tabelas

Lista de Algoritmos

1. Introdução

O primeiro capítulo da presente tese tem como objectivo fazer uma introdução ao trabalho desenvolvido ao longo da tese. Começa por apresentar um enquadramento e a motivação que levaram à realização deste trabalho, seguindo-se a definição de objectivos a atingir. Por fim, é apresentada a estrutura da tese.

1.1. Enquadramento

O final do século XX ficou marcado pelo crescimento exponencial do sector das telecomunicações. No seguimento deste crescimento e em simultâneo com a evolução tecnológica, os operadores de telecomunicações enfrentam um novo desafio: a fraude. A fraude nas telecomunicações pode ser definida em poucas palavras como o uso dos serviços ou dos produtos de um operador sem intenção de pagar. A fraude nas telecomunicações não é só um risco que os operadores correm, é um negócio extremamente organizado, à escala mundial, que afecta operadores por todo o mundo.

A Communications Fraud Control Association (CFCA) e a Telecommunications UK Fraud Forum (TUFF) são organizações que se dedicam ao combate à fraude nas telecomunicações.

"*CFCA is the Premier International Association for revenue assurance, loss prevention and fraud control through education and information.*" (1)

A CFCA publicou um estudo no ano de 2006 que alerta para a severidade do problema da fraude. De acordo com este estudo as perdas anuais ao nível mundial no sector das telecomunicações estavam entre os 54 e os 60 mil milhões US$ (dólares americanos). Nesse mesmo estudo era indicado um aumento de 52% em relação ao ano de 2003. O

presidente da CFCA, Clemmie A. Scott, interpretou estas estatísticas como um sinal da afirmação da fraude nas telecomunicações como um negócio ilegal e lucrativo.

"*TUFF – A forum for the exchange of information and the promotion of a united effort against telecommunications fraud*" (2)

No Reino Unido o TUFF estima que a perda anual devido à fraude ronde as 866 mil £ (libras inglesas). Em estudos mais recentes tanto a CFCA como o TUFF estimaram que a fraude causa perdas por volta do valor de 5% do valor da receita de um operador, podendo em alguns casos específicos, como no caso de novos operadores, chegar aos 15%.

Dado que as penas para este tipo de fraude são leves, quando comparadas com o roubo de montantes similares, combinado com a dificuldade de se obter uma condenação, defraudar operadores de telecomunicações é uma actividade de risco relativamente baixo e altamente lucrativo.

A fraude nas telecomunicações é uma realidade que os operadores de telecomunicações devem encarar de forma séria. É um negócio organizado, ilegal, lucrativo e apelativo para os criminosos, em contínua evolução e que causa sérios danos aos operadores de telecomunicações.

Ao longo da tese o conceito de "fraude nas telecomunicações" será referido como fraude e o conceito de "operador de telecomunicações" como operador. Um subscritor de um operador que usa os serviços do operador com intenção de cometer fraude será referido como fraudulento.

1.2. Motivação

A fraude tem um impacto nos operadores a vários níveis:

- Financeiro – a fraude pode resultar na perda directa de receita quando os serviços de um operador são usados sem serem pagos. Há ainda o custo de o operador ter de pagar por serviços usados não conseguindo mais tarde cobrar

estes mesmos custos. Explicando melhor este último ponto: um exemplo dos tipos mais comuns de fraude é quando um fraudulento cria os seus próprios Premium Rate Services (PRS), serviços de valor acrescentado, e depois conseguem de alguma forma usar os serviços de um operador para efectuar chamadas para esses PRS. O operador terá de pagar uma comissão à empresa a que pertence o PRS mas não consegue liquidar junto do subscritor fraudulento essa mesma dívida.

- Legal – o uso fraudulento dos serviços de um operador pode resultar no incumprimento de exigências legais e regulamentares, o que acarreta o risco de má publicidade e multas.
- Recursos – a fraude pode afectar o funcionamento normal da rede de um operador, causando consequentemente uma diminuição da qualidade do serviço prestado pelo operador, que pode levar também ao risco de má publicidade e multas.
- Relação com os consumidores – ao afectar a qualidade dos serviços prestados e ao afectar os valores das facturas dos consumidores, a fraude pode causar nos operadores a perda de clientes para outros operadores mais seguros e pode mesmo dar origem a acções legais por parte dos mesmos. Mais uma vez, pode levar também ao risco de má publicidade e multas.

O impacto da fraude causa sérios danos financeiros e na imagem da marca a um operador, podendo mesmo afectar a confiança dos accionistas e *shareholders* e, em casos extremos, afectar a cotação na bolsa.

Os operadores sabem que a fraude é um problema que não pode ser erradicado. Por isso adoptaram como estratégia a minimização de custos. Assim, os operadores desenvolveram ou adquiriram sistemas de informação direccionados para a detecção de fraude. Contudo, a fraude continua a ser uma das maiores (se não for mesmo a maior) causas de perda de receitas no sector das telecomunicações. Assim, surge a necessidade dos operadores investigarem novas estratégias a adoptar para fazer frente aos ataques de fraude a que são sujeitas.

1.3. Objectivos

Com o trabalho efectuado ao longo desta tese pretende-se, numa primeira fase, **estudar o problema** da fraude nas telecomunicações **e a solução** que os operadores adoptaram para resolver este problema. O objectivo desta primeira fase é analisar a solução actual e investigar **de que forma esta solução pode evoluir**, ou que outras soluções e novas abordagens inovadoras se poderão propor, de forma a auxiliar os operadores a lidar com o problema da fraude.

Numa segunda fase, pretende-se evoluir os resultados obtidos do trabalho efectuado na primeira fase, **desenvolvendo um novo modelo para a detecção e implementando um protótipo**. A implementação de um protótipo é importante de forma a determinar se o trabalho efectuado tem capacidade, tanto ao nível da *performance* qualitativa como ao nível da *performance* quantitativa, para ser incorporado na solução para detecção de fraude nos operadores. Por *performance* qualitativa entenda-se que é uma solução capaz de melhorar o processo de detecção de fraude detectando outros casos de fraude que não seriam possíveis de detectar com a solução actual. A *performance* quantitativa também é importante, uma vez que o volume de dados tratado nas soluções de detecção de fraude é enorme e a nova solução deve ser capaz de os tratar em tempo útil.

Sintetizando, os objectivos para esta tese são:

- Estudar, analisar e **detectar oportunidades de evolução** da actual solução de detecção de fraude;
- Concretizar as oportunidades de evolução da solução, **definindo um novo modelo de detecção de fraude**;
- **Implementar um protótipo** de forma a ser possível obter indicações acerca da *performance* (qualitativa e quantitativa) do novo modelo.

1.4. Estrutura da tese

A presente tese é composta por oito capítulos. Inicia-se com o presente capítulo de introdução onde é feito um enquadramento, apresentada a motivação que levou à realização da tese, definidos os objectivos a atingir e apresentada a estrutura da tese.

O capítulo 2 dedica-se ao estudo da fraude nas telecomunicações, das soluções existentes para resolver este problema e termina com a definição de uma nova abordagem. São apresentados e explicados vários tipos de fraude que afectam os operadores. Neste capítulo é ainda analisada a actual solução para detecção de fraude. Desta análise resulta a identificação dos pontos fortes e fracos da solução bem como a identificação de oportunidades de melhoria. Consequentemente é especificada uma nova abordagem ao problema, com base nas oportunidades previamente identificadas.

No capítulo 3 é efectuada uma pequena introdução à área dos Sistemas Multi-Agente, seguindo-se a definição do sistema que servirá de suporte ao desenvolvimento da nova abordagem estabelecida no capítulo anterior. Esta definição consiste em especificar os vários parâmetros que constituem o sistema (arquitectura, coordenação, comunicação e organização), os seus agentes que formarão o sistema e por fim os processos que serão implementados.

Os capítulos 4, 5 e 6 são dedicados aos agentes que compõem o Sistema Multi-Agente, nomeadamente, o agente de *profiling*, o agente de detecção e o agente de extracção de conhecimento. No início de cada capítulo são apresentadas as tecnologias usadas em cada agente, como a técnica de *profiling*, a análise de redes sociais e a extracção de conhecimento, seguindo-se a definição dos processos de cada agente.

No capítulo 7 é apresentado um protótipo da solução desenvolvida. A implementação deste protótipo e a realização de testes sobre o mesmo visa obter indicações acerca do desempenho, quantitativo e qualitativo, da solução.

A presente tese termina com o capítulo 8, o capítulo de conclusões, onde é efectuada uma revisão dos objectivos definidos inicialmente resultando numa reflexão crítica

sobre todo o trabalho desenvolvido. Ainda neste capítulo é exposto o trabalho futuro a desenvolver, no seguimento da conclusão da tese.

2. Análise ao problema da fraude

A GSM Association (3) identificou mais de 50 diferentes tipos de categorias de fraude. Devido à constante evolução tecnológica os diferentes tipos de fraude já existentes evoluem e novos tipos são criados com alguma regularidade.

Não faz parte do âmbito deste capítulo enumerar e explicar todas as categorias de fraude existentes. Apenas os principais tipos de fraude, que são de maior interesse para os operadores actualmente, serão explicados.

Após a introdução aos principais tipos de fraude, na segunda secção é feita uma pequena introdução à WeDo Technologies e apresentada a sua actual solução para gestão de fraude.

Na terceira secção é apresentada uma visão geral de outras soluções de gestão de fraude existentes bem como de investigação encontrada na internet acerca deste tema.

Na quarta secção é apresentada uma análise às soluções estudadas, identificando os pontos fortes e fracos com o objectivo de detectar uma oportunidade para evoluir a actual solução. Nesta secção é definida uma nova abordagem ao problema da fraude de forma a explorar a oportunidade detectada.

2.1. Tipos de fraude

Começando pelo tipo mais comum de fraude: **subscrição**. O fraudulento obtém uma subscrição dos serviços do operador sem nenhuma intenção de pagar, usando uma identidade falsa ou usando a identidade de outra pessoa (“*roubando*” os dados necessários para uma subscrição). Os danos que este tipo de fraude pode causar num operador variam consoante a intenção de fraudulento: num nível mais simples o fraudulento pode usar a subscrição para seu uso pessoal até ser detectado; num nível

mais sofisticado, um fraudulento pode usar a subscrição para efectuar chamadas ou enviar SMS (SMS é um acrónimo para Short Message Service, comummente apelidadas de mensagens de texto) para Premium Rate Services (PRS), serviços de valor acrescentado, de forma a obter "*lucro*" do uso dos serviços, expondo assim o operador a impactos financeiros maiores.

Um dos principais problemas da fraude por subscrição é distinguir esta fraude de um simples subscritor que por alguma razão, que não a de cometer fraude, não paga os serviços que subscreveu.

A fraude por subscrição é uma forma muito usual para os fraudulentos ganharem acesso aos serviços de um operador para depois praticarem diferentes tipos de fraude. Um estudo recente do TUFF (2) revela que **a fraude por subscrição é o tipo de fraude mais predominante** de todos, **representando cerca de 40% de todas as fraudes nas telecomunicações** no Reino Unido.

Um dos principais tipos de fraude é a **clonagem de cartões SIM** (o acrónimo SIM significa Subscriber Identity Module). Este tipo de fraude consiste na duplicação de cartões SIM de subscritores normais. O software para clonar estes cartões está disponível na internet, por isso, um fraudulento que tenha acesso físico a um cartão só precisa de um computador e um leitor de cartões para clonar o cartão original. Alguns operadores validam que apenas um cartão de cada vez pode estar activo, outros não têm este tipo de validação e ficam expostas a este tipo de fraude. De acordo com um artigo publicado pelo Federal Bereau of Investigation (FBI), a clonagem de cartões é uma das causas mais comuns de fraude (4).

Um tipo de fraude mais sofisticado é a fraude por **bypass**. Normalmente cometido usando tecnologia Voice over Internet Protocol (VoIP)[1], para fazer um desvio de tráfego internacional, impedindo assim os operadores de cobrar o tráfego de *interconnect* normalmente cobrado em comunicações internacionais.

Um tipo de fraude que os operadores não descuram é a **fraude interna**. Implica acção de funcionários do operador, que tipicamente desempenham funções em que têm

[1] Voz sobre IP, também chamado de VoIP, é o roteamento de comunicações usando a Internet ou qualquer outra rede de computadores baseada no Protocolo de Internet.

conhecimento e acesso a sistemas de informação que lhes permite manipular informação de forma a beneficiar terceiros como, por exemplo, atribuindo minutos extras, modificando dados de subscrição dos clientes (morada, número de identificação fiscal) e mesmo, a um nível muito mais perto do crime organizado, fornecer aos fraudulentos informações que lhes permitam não ser detectados pelos sistemas de detecção de fraude.

Estas são as principais causas de fraude nas telecomunicações. Existem muitas outras categorias de fraude, mas que normalmente têm origem em um dos tipos de fraude referidos anteriormente.

Um tipo de fraude que já foi bastante popular é o **Call Selling**. Este tipo de fraude diminuiu muito devido ao aparecimento do VoIP. O fraudulento adquire vários cartões SIM, quer através da fraude por subscrição quer através da clonagem de cartões, e depois vende chamadas de voz internacionais, normalmente em comunidades de imigrantes. O operador terá que pagar pelo tráfego internacional aos operadores estrangeiras e não conseguirá cobrar a dívida aos subscritores fraudulentos.

Na fraude de **PRS**, um fraudulento regista um PRS, serviço de valor acrescentado, num país estrangeiro, adquire vários cartões SIM, quer através da fraude por subscrição quer através da clonagem de cartões, efectua chamadas ou envia SMS em volume para o número do PRS que montou previamente. O operador tem que pagar ao fornecedor do PRS, neste caso o fraudulento, uma comissão por cada minuto da chamada ou por cada SMS enviada, mas no fim não consegue cobrar aos subscritores. Numa recente variação deste tipo de fraude, em vez de usarem cartões clonados ou subscrições falsas, o que os fraudulentos fazem é: aproveitam algumas vulnerabilidades dos Short Message Service Center (SMSC)[2] para enviar SMS em massa para os subscritores normais de um operador com mensagens do género: “Parabéns! Acabou de ganhar uma PlayStation3. Para saber como receber o seu prémio ligue para este número: 4433. Não perca esta oportunidade.” O número que aparece na SMS é o número do PRS do fraudulento. Neste caso, o operador consegue cobrar o valor da chamada aos seus subscritores normais e pagar ao fornecedor fraudulento do PRS, mas este tipo de

[2] Um SMSC é um elemento da rede do operador de telecomunicações, resposável por controlar o envio e a recepção de SMS.

acções pode levar à perda de clientes e ao risco de má publicidade, causando assim danos na imagem da empresa.

Outro tipo de fraude consiste nos ataques a um Private Automatic Branch Exchange (PABX)[3]. Geralmente os ataques a PABX são feitos usando a técnica de "War Dialing", tentado através de várias tentativas ganhar acesso a uma PABX, para depois redireccionar tráfego através desta, sejam chamadas internacionais ou chamadas e SMS para PRS. Tal como no caso anterior, o operador consegue cobrar o valor da chamada ao subscritor que detém a PABX e pagar ao operador de destino da chamadas internacionais ou ao fornecedor fraudulento do PRS, mas mais uma vez, este tipo de acções geralmente causa perda de clientes e má publicidade, consequentemente, danos na imagem da marca do operador.

A fraude nos cartões pré-pagos, conhecida por **Prepaid**, tem normalmente um das seguintes 3 causas. Pode ser causada por fraude interna de forma a manipular plataformas internas de informação, em que funcionários do operador alteram definições dos dados de subscrição, efectuam recargas nos cartões, dão créditos ou minutos grátis, etc. Uma outra forma de cometer fraude nos cartões pré-pagos é usar cartões de crédito roubados ou clonados para recarregar os cartões. Há ainda uma terceira forma de cometer fraude nos pré-pagos: manipulação dos Voucher Hidden Recharge Numbers (Voucher HNR). O conceito de *voucher* não existe em Portugal, mas em muitos países é usado: um subscritor quer carregar o seu cartão pré-pago, então dirige-se a um quiosque ou a um estabelecimento que venda estes *vouchers*. Compra um *voucher* de 10€, por exemplo, e recebe um papel que contém um código HNR, *Hidden Recharge Number*, que se envia para um número e automaticamente o seu saldo é incrementado com €10. Existe então a possibilidade de cometer fraude manipulando estes HRNs. Isto pode ser feito através da fraude interna, em que um funcionário do operador que tem acesso a estes HNRs os vende/oferece aos fraudulentos. A um nível mais sofisticado podem também ser desenvolvidos algoritmos e programas que tentam adivinhar HNRs e produzir *vouchers* falsos.

[3] Um PABX é uma central telefónica utilizada para fins particulares, dentro de uma empresa ou de um edifício por exemplo.

Alguns operadores usam o conceito de fraude de **Serviço/Equipamento**. Por vezes, em campanhas de marketing, os operadores vendem equipamentos por valores inferiores ao seu custo normal, ficando depois o subscritor comprometido a pagar uma mensalidade durante um certo período de tempo definido. Este tipo de campanha acontece muito com o iPhone hoje em dia, por exemplo, em que o subscritor apenas paga metade do valor do equipamento e compromete-se a pagar o resto do valor em mensalidades durante dois anos. Uma fraude por subscrição resulta num fraudulento obter um equipamento valioso por um preço inferior ao seu real valor pois ele não vai pagar as mensalidades durante os dois anos seguintes. O mesmo conceito aplica-se à venda de serviços pelo operador.

Por fim, há uma categoria de fraude que tem um impacto financeiro relevante nos operadores e que tem uma característica diferente dos tipos de fraude apresentados até agora: fraude por **Roaming**. Este tipo de fraude consiste em cometer alguns dos tipos de fraude anteriores, como por exemplo a venda de chamadas, mas com a diferença de ser cometida no estrangeiro, em Roaming. A diferença deste tipo de fraude para os outros é a seguinte: quando uma chamada é efectuada ou uma SMS é enviada em território nacional, o operador tem logo acesso a essa informação e os sistemas de detecção de fraude recebem logo informação relativa a esses eventos (chamadas ou SMS). Quando uma chamada é efectuada ou uma SMS é enviada em Roaming, ou seja, em território estrangeiro, o operador desse país estrangeiro que possibilita que o subscritor efectue essa chamada demora algum tempo até enviar os ficheiros TAP[4] (Transfer Account Procedure) que contêm informação relativa a esses eventos. Assim, o operador pode ficar horas sem receber esses ficheiros e abre uma janela de tempo muito grande para os subscritores fraudulentos usar.

[4] TAP é um mecanismo pelos quais operadores GSM trocam informação de facturação em Roaming de forma a poderem cobrar os custos da utilização de serviços.

2.2. Análise à solução para detecção de fraude

O objectivo desta secção não é analisar exaustivamente uma solução de fraude, mas sim compreender como esta funciona, de forma poder analisar como esta solução pode evoluir. Uma solução especificamente concebida para a gestão de fraude é normalmente designada por Fraud Management System (FMS). O FMS em estudo é a solução da empresa onde trabalho, a WeDo Technologies, e chama-se Fraud:RAID.

A WeDo Technologies é um fornecedor líder de soluções de software. Criada formalmente em Junho de 2000, a WeDo Technologies iniciou a sua actividade comercial em Fevereiro de 2001. Actualmente, conta com cerca de 400 consultores, e escritórios em todos os continentes.

O *know-how* da WeDo centra-se no Business Assurance para redes Telecom, com liderança nas áreas de Revenue Assurance, Roaming, Commissions e Credit & Collections Solutions. A nível nacional, a WeDo intervém em todos os mercados – Saúde, Financeiro, Indústria e Telecomunicações.

O Fraud:RAID é o novo sistema de gestão de fraude da WeDo Technologies, tendo sido especificamente desenhado para combater a fraude no sector das telecomunicações. É um sistema de gestão de fraude especificamente desenhado para ser completamente **configurável pelo operador**, permitindo assim à equipa de gestão de fraude a configuração dinâmica das técnicas de detecção de fraude do Fraud:RAID de forma a visarem novos casos de fraude à medida e quando estes forem surgindo.

Suporta técnicas padrão de correlação, definição de perfis e estatísticas. Estas técnicas são, regra geral, específicas e permitem ajudar as empresas a detectarem e rastrearem potenciais clientes fraudulentos a partir do momento em que estes subscrevem os serviços do operador e durante todo o seu ciclo de vida.

Incorpora um Graphical User Interface (GUI) baseado em Web com **controlo de acesso, dashboards e relatórios configuráveis pelo utilizador** que permitem conceder às pessoas certas da equipa e de toda a empresa uma visão do estado e êxito da investigação da sua equipa. Os controlos de acesso do Fraud:RAID permitem

determinar a informação que é apresentada a cada utilizador do sistema, para que todos tenham a informação que necessitam com o nível de detalhe adequado.

O Fraud:RAID permite que a equipa de gestão de fraude observe e rastreie todos os aspectos das actividades fraudulentas e análise de forma fácil, numa ferramenta completa de **gestão de casos**, proporcionando assim à equipa total visibilidade bem como um histórico completo das actividades fraudulentas. Esta visão única permite eliminar clientes fraudulentos e ajuda a aconselhar a equipa sobre como eliminar as lacunas técnicas de dos processos do negócio que são exploradas para fins fraudulentos que prejudicam o operador.

A Figura 1 representa a arquitectura da solução.

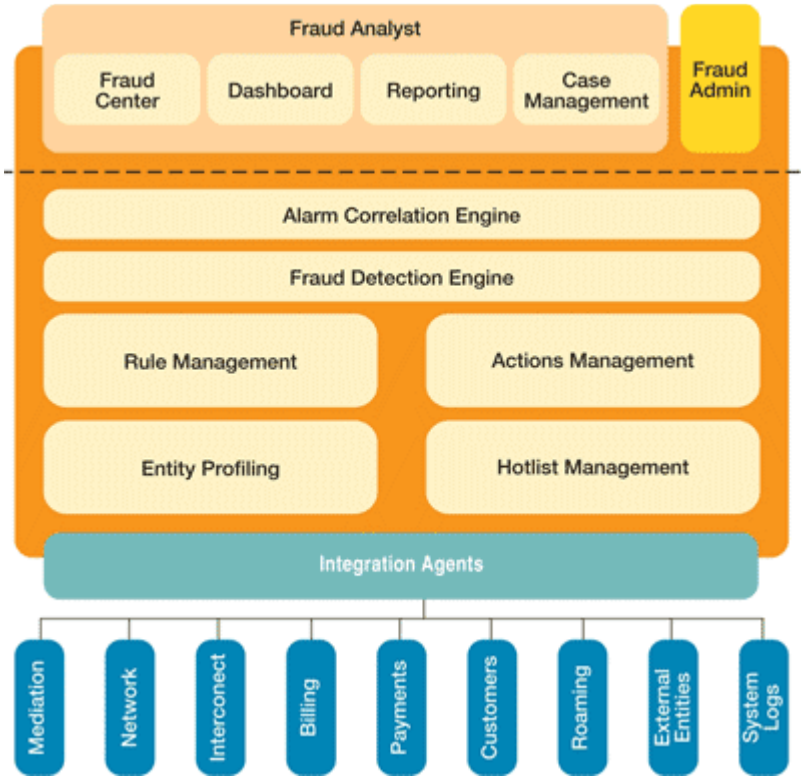


Figura 1 – Arquitectura da solução de fraude Fraud:RAID

A arquitectura da solução deve ser interpretada de baixo para cima. A primeira fase é a aquisição de dados efectuada pelos "**Integration Agents**". O objectivo desta fase é integrar todos os dados relevantes para o problema de gestão de fraude a partir de várias fontes, por exemplo: a mediação (Mediation), dados de rede (Network), facturação (Billing), pagamentos (Payments) ou catálogo de clientes (Customers).

Para ter a informação dos subscritores disponível durante os procedimentos de análise, é possível importar o catálogo de subscritores do sistema CRM (Customer Relationship Management), sistema de Billing & Customer Care, etc. Para ajudar o analista a classificar o potencial de fraude, o Fraud:RAID segmenta a base de subscritores ("**Entity Profiling**") de acordo com a idade de subscrição, histórico de pagamentos, retalho, pequenas empresas, corporate, perfil de risco, etc. Para visar segmentos de subscritores de risco mais elevado, as regras de detecção do Fraud:RAID ("**Rule Management**") são adaptadas de modo que estas apliquem a esses segmentos critérios de detecção mais restritos.

O Fraud:RAID possibilita a definição de Listas Negras ("**Hotlists Management**"), de forma a poder definir listas de entidades (subscritores, parceiros, funcionários) que devem ser supervisionados com especial atenção durante o processo de detecção de fraude.

Os métodos de detecção do Fraud:RAID são implementados em motores ("**Fraud Detection Engines**") que se baseiam na construção de estatísticas e na definição de limites, previamente definidos no Rule Management. Alguns exemplos desses métodos:

- High Usage – mede a quantidade de tráfego gerado por uma dada entidade. Compara com os limites definidos e gera alertas para as entidades que superem os limites.
- Colisão de chamadas – detecta chamadas efectuadas a partir do mesmo cartão SIM sobrepostas no tempo.
- Ratio – monitoriza os serviços (voz IN e voz OUT por exemplo) que as entidades usam. Lança alertas com base nos limites definidos, para entidades com um ratio desproporcional de uso de serviços.

O Fraud:RAID implementa uma técnica de detecção de duas fases: na primeira fase, a detecção referida anteriormente gera alertas com base nos dados de eventos; seguida da fase de correlação de alertas que gera casos que representem potenciais fraudes. O objectivo é gerar um caso de fraude, com base em vários alertas, gerados por um ou mais dos métodos de detecção da fase anterior. A correlação de alertas é controlada

por um motor ("**Alerts Correlation Engine**") baseado em regras totalmente configuráveis por analistas autorizados.

Tanto a fase de geração de alertas como a fase de correlação de alertas podem ser configuradas de modo a assumir acções automáticas caso detecte um conjunto específico de condições. As acções automáticas, definidas no "**Actions Management**", podem incluir o envio de SMS ou E-mails ou a aplicação de restrições à conta de um subscritor, etc.

Os casos gerados na fase de correlação de alertas podem ser atribuídos a equipas ou utilizadores específicos ("**Fraud Center**"). Os casos fornecem ao analista toda a informação relevante para a decisão. Além disso, através da GUI de gestão de casos, o analista pode requerer acções como pesquisas na base de conhecimentos, ir buscar dados a sistemas externos ou associar casos que o analista determine que possam estar relacionados ("**Case Management**").

O Fraud:RAID incorpora dashboard ("**Dashboard**") e relatórios ("**Reporting**") baseados na Web, configuráveis pelo utilizador, que permitem à equipa de gestão de fraude conceder às partes interessadas, acesso à informação relevante sobre dados financeiros, indicadores operacionais, estado e êxito de investigações de fraude, etc.

2.3. Outras soluções

Existem várias ofertas de soluções de FMS no mercado. As técnicas mais utilizadas são a detecção baseada em regras. Recentemente as empresas estão a apostar em técnicas baseadas em Data Mining (DM) e Inteligência Artificial (IA) de forma a tentarem retirar indicadores de fraude do elevado volume de dados que dispõem. Empresas como a Detica, Azure, Agilis ou Telbit oferecem soluções de FMS similares à da WeDo Technologies, no que diz respeito ao tipo de detecção de fraude baseado em regras.

A solução da Detica (5) NetReveal é focada na monitorização da actividade entre subscritores e no seu comportamento com o objectivo de determinar padrões de comportamento. A Azure (6) oferece uma solução de FMS, a Azure Fraud Control System (AFCS), baseada em regras, mas está actualmente a desenvolver uma solução baseada em Data Mining para complementar os métodos actuais. Entre outros métodos destaco a detecção baseada em desvio de comportamentos e a identificação de actividade criminal através da técnica de *fingerprinting*. Esta técnica tem por objectivo analisar o comportamento dos utilizadores fraudulentos de forma a retirar indicadores acerca da sua actividade na rede. A solução da Agilis (7), o NetMind, é um sistema de garantia de receita e de gestão de fraude que se foca no ciclo de vida de um subscritor identificando e gerindo pontos de risco do subscritor nesse mesmo ciclo. A Telbit (8) desenvolveu um FMS, o Centaur, que além detecção baseada em regras também inclui métodos de baseados em Inteligência Artificial usando técnicas de assinaturas, *fingerprinting* e *profiling*. Esta última técnica consiste na construção de perfis de comportamento dos utilizadores a partir da sua actividade na rede.

A detecção de fraude nas telecomunicações é um tema que está muito em voga. É possível encontrar na internet dezenas de artigos que contemplam este tema. Os próximos parágrafos consistem numa visão geral de vários artigos, basicamente um resumo dos objectivos de artigos relacionados com este tema, que foram analisados numa primeira fase da realização desta tese, uma fase de investigação.

Estevez (9) propôs um modelo baseado em *fuzzy rules* e redes neuronais para detectar fraude por subscrição, mas o seu modelo apresentava uma falha importante pois não contemplava a fraude por clonagem de cartões SIM.

De forma a tentar suprimir este problema, Fawcett e Provost (10) (11) desenharam uma solução baseada nas técnicas de *profiling* e Data Mining, com o objectivo de descobrir fraudulentos com padrões de actividade diferentes dos padrões considerados normais. A solução usa regras criadas para cada subscritor em vez de regras universais. O processo consistia em, para cada subscritor: utilizar 30 dias de tráfego livre de fraude e anexar de seguida um período de fraude, de seguida é analisada esta mudança no tráfego do subscritor e criadas regras que traduzissem essa

mudança. Assim cada subscritor tem um conjunto de regras que traduz a mudança na sua actividade caso fossem afectados por uma fraude.

Taniguchi (12) e Hollmen (13) desenvolveram uma solução que baseada no comportamento passado do utilizador, usando uma função de densidade probabilística, para depois calcular o provável comportamento actual do utilizador e detectar diferença entre este comportamento calculado e o comportamento real do utilizador.

Burge e Shawe-Taylor (14) (15) apresentaram uma solução baseada em redes neuronais recorrentes, que usa a técnica de *profiling*. Esta técnica calculava, para cada utilizador, dois perfis: Behavior Profile History (BPH) baseado no histórico da actividade do utilizador; Current Behavior Profile (CBP) baseado na actividade actual do utilizador. De seguida usavam a técnica *Hellinger distance*[5] para calcular a distância entre os dois perfis. Uma distância maior que um limite previamente definido significaria um caso de fraude.

A solução de Cortes (16) (17) (18) produz assinaturas para cada utilizador, baseadas em sumários estatísticos que geram um vector de valores, uma assinatura. Tal como Burge e Shawe-Taylor, Cortes produzia assinaturas com base no comportamento passado e no comportamento actual, a fraude é detectada através do cálculo da distância entre as assinaturas.

A solução de Ferreira, Alves, Belo, Lopes, Ribeiro, Cortesão e Martins (19) (20) é muito similar à anterior. Utiliza Data Mining para calcular as assinaturas, com a particularidade de não calcular a distância entre as assinaturas mas calcular a diferença entre os valores que formam a assinatura.

Weiss (21) desenvolveu um conjunto de métodos, baseados na técnica de Data Mining, direccionados para descobrir conhecimento "escondido" dentro das quantidades enormes de dados que os operadores têm acerca da actividade dos subscritores. Estes métodos tinham como principal objectivo a detecção de fraude, melhorar a eficiência do marketing e identificar falhas na rede do operador.

[5] Na teoria das probabilidades, a distância de Hellinger é usada para quantificar a similaridade entre duas distribuições de probabilidade.

Apesar de ser impossível rever todos os artigos existentes na internet, é possível conceber uma visão geral dos trabalhos publicados e da tendência das soluções apresentadas.

Data Mining, Inteligência Artificial, *profiling* e métodos estatísticos são as técnicas mais propostas e utilizadas para evoluir as soluções de FMS. Estas técnicas são orientadas ao estudo do comportamento do utilizador, com o objectivo de calcular indicadores/perfis/assinaturas que permitam identificar comportamentos fraudulentos, evoluindo as soluções de FMS tornando-as mais inteligentes, flexíveis e mais eficientes no que diz respeito à detecção de fraude de fraudulentos com base no seu comportamento.

2.4. Definição de uma nova abordagem

A solução de gestão de fraude Fraud:RAID permite à equipa de gestão de fraude a configuração dinâmica da monitorização dos dados e das técnicas de detecção de fraude de forma a detectarem novos suspeitos de fraude. Simplificando o processo: são definidos métodos de detecção e limites para esses métodos, que monitorizam o tráfego da rede do operador e mantém estatísticas que, quando atingem o limite, lançam alertas acerca de um suspeito de fraude. Estes alertas são agregados e um caso é gerado para que os analistas verifiquem se realmente se trata de um caso de fraude e tomar as devidas acções.

O **problema** é que esta solução foi desenhada para **detecção de fraude**, é uma **solução reactiva**, isto é, detecta a fraude depois de ela ter acontecido. Apesar de detectar um fraudulento e de terminar a sua actividade na rede evitando custos futuros, os operadores terão ainda que suportar os custos da fraude realizada até ao momento da detecção e bloqueio do fraudulento. Concluindo: para haver detecção tem de existir fraude. Isto traz grandes problemas para os operadores, uma vez que os fraudulentos estão continuamente a entrar na sua rede, usam os serviços até serem detectados e bloqueados, iniciando o processo de novo.

Se observarmos as soluções concorrentes de outras empresas e os trabalhos publicados na internet que foram objecto de estudo, verificamos que apresentam técnicas similares às técnicas presentes na solução de FMS que a WeDo oferece (detecção baseada em regras, modelos estatísticos, profiling) e às técnicas que a WeDo está actualmente a evoluir (Data Mining, Inteligência Artificial, Redes Neuronais). Para além da similaridade entre as técnicas utilizadas, também se verifica similaridade no objecto de estudo: o comportamento dos subscritores. Todas estas técnicas são orientadas para detectar fraude com base no comportamento do subscritor. O problema é que para ser detectado, um subscritor fraudulento tem que ter uma considerável actividade na rede do operador para que o seu perfil/assinatura/indicadores de comportamento sejam construídos. Uma vez detectados e banidos da rede, todo o processo reinicia, os fraudulentos voltam a entrar na rede do operador e os métodos de detecção de fraude voltam a construir perfis/assinaturas/indicadores para detectar se é um subscritor fraudulento ou não. O que falta a estas soluções é complementar estes métodos orientados ao comportamento com métodos orientados à identidade de subscritores, com o objectivo de, uma vez banidos da rede por comportamento fraudulento, seja possível identificar se o subscritor reentra na rede, o mais depressa possível, evitando assim o uso abusivo por parte do mesmo.

Aqui surge uma **oportunidade** para evoluir a solução de FMS para uma **solução proactiva**, que permita a **prevenção de fraude**. O objectivo é detectar fraudulentos que foram previamente detectados e banidos da rede o mais rápido possível assim que eles reentrem na rede do operador. Para tal, definiu-se uma **nova abordagem, orientada à detecção por identidade** do fraudulento em vez da detecção por comportamento. Esta abordagem detalhada de seguida **complementa a actual solução, não a substitui**. Usa os dados de tráfego por ela monitorizados e necessita dos métodos de detecção de fraude para primeiro detectar os fraudulentos. Só após, pode usar esta informação para detectar que o fraudulento reentra na rede, antes de ele usar abusivamente os serviços do operador, prevenindo os danos que este iria causar. Contudo, esta nova abordagem não será unicamente direccionada à detecção por identidade, apesar de este ser o objectivo principal, pois serão incluídos métodos

para estudar e extrair conhecimento acerca do comportamento do fraudulento. Segue a nova abordagem, explicada de forma mais detalhada e dividida em três fases:

- A **primeira fase** da nova abordagem ao problema consiste no armazenamento e na utilização dos dados de tráfego para **criar um perfil** para cada utilizador do operador. Este perfil é constituído por atributos de identidade e atributos de comportamento. Os atributos de identidade contêm informação que permitem, na segunda fase, identificar o fraudulento como indivíduo. Os atributos de comportamento contêm informação que identificam o fraudulento como pertencente a um grupo de risco, serão usados na terceira fase para extracção de conhecimento.
- O perfil resultante da fase anterior deve conter os atributos suficientes para que numa **segunda fase** do processo, quando um utilizador é identificado como fraudulento, seja possível **detectar que o fraudulento reentrou na rede** do operador, através da análise destes atributos. Esta análise será baseada nos no grupo de atributos do perfil de identidade.
- Complementarmente à segunda fase, uma **terceira fase** usa os atributos de comportamento do perfil para **extrair conhecimento** acerca do comportamento do fraudulento, de forma a providenciar aos analistas de fraude mais informação acerca dos perfis dos fraudulentos de forma a poderem usar esta informação para optimizar os processos de detecção (aplicando esta informação na configuração dos métodos tradicionais de detecção de fraude previamente apresentados) e prevenção de fraude (aplicando esta informação na configuração do processo da segunda fase desta solução).

Esta nova abordagem, dividida em três fases, tem as seguintes particularidades:

- Não substitui a solução actual, é **complementar** à mesma. Necessita dos dados da solução actual (dados da monitorização da rede e fraudulentos detectados) para conseguir atingir os seus objectivos;
- A primeira fase não produz qualquer tipo de *output* para os analistas de fraude. O seu propósito é construir informação para *"alimentar"* as fases seguintes.

Contudo é a fase que claramente trabalha um maior volume de informação, por isso é relevante que a *performance* a um nível **quantitativo** desta fase seja objecto de estudo quando for implementado um protótipo;

- A segunda fase permitirá evoluir a solução actual de reactiva para **proactiva**, pois ao detectar os fraudulentos quando eles reentram na rede do operador e antes de eles usarem abusivamente dos seus serviços previne os danos causados pelas fraudes que iria cometer. Esta é a fase cujos resultados (detecção fraudulentos que reentram na rede) serão mais visíveis para os analistas de fraude, por isso é relevante que a *performance* a um nível **qualitativo** desta fase seja objecto de estudo quando for implementado um protótipo;
- A terceira fase permitirá aos analistas de fraude que usam a solução com a nova abordagem **melhorarem o seu *know-how*** acerca do comportamento dos fraudulentos na rede, **potenciando os recursos** para detecção e prevenção de fraude.

3. Sistema Multi-Agente

A nova abordagem definida no capítulo anterior será desenvolvida usando como suporte um sistema multi-agente (SMA). Este capítulo começa com uma breve introdução aos Sistemas Multi Agente, seguindo-se a definição do SMA que sustentará a solução para a nova abordagem. Para completar a definição do SMA também a sua arquitectura e processo serão apresentados.

NOTA: Normalmente o acrónimo SMA é utilizado para referir Sistemas Multi-Agente, a subárea da Inteligência Artificial Distribuída. Nesta tese, SMA é utilizado para referir o sistema multi-agente que servirá de suporte à contrição da nova abordagem definida. Sempre que se referir à área Sistemas Multi-Agente, será feita por extenso, sem acrónimo.

3.1. Introdução aos Sistemas Multi-Agente

A Inteligência Artificial Distribuída (IAD) é um ramo da IA, dedicado ao estudo da resolução de problemas através de sistemas computacionais distribuídos (22). Actualmente este campo da IA foi claramente ultrapassado pelo campo dos Sistemas Multi-Agente.

O principal objectivo dos Sistemas Multi-Agente é o estudo, construção e aplicação de sistemas multi-agente, isto é, **sistemas compostos por vários agentes inteligentes**, que interagem entre si e com outros sistemas, **perseguindo um conjunto de objectivos** e/ou **executando um conjunto de tarefas** (23). Os Sistemas Multi-Agente compreendem um conjunto de entidades (agentes) que cooperam de forma a solucionar um dado problema, o que normalmente está para lá das suas capacidades individuais (24).

Um agente corporiza um sistema computacional capaz de revelar uma **acção autónoma e flexível**, desenvolvido num determinado universo de discurso. A

flexibilidade do agente está relacionada com as suas capacidades de reacção, iniciativa, aprendizagem e socialização (25).

A **noção fraca** de agente implica que o agente tenha um conjunto mínimo de características (26): autonomia, o agente opera sem intervenção de outros agentes e controla as suas acções e o seu estado de conhecimento; reactividade, o agente percepciona os eventos que ocorrem no seu universo de discurso e responde adequada e atempadamente a mudanças ocorridas nesse ambiente; pro-actividade, o agente toma a iniciativa, conduzindo as suas próprias acções mediante um comportamento dirigido por objectivos; sociabilidade, os agentes relacionam-se com outros agentes, comunicando, competindo ou cooperando na resolução de problemas que lhes sejam colocados.

A **noção forte** de agente implica que o agente tenha capacidades cognitivas, que seja passível de desenvolver a sua própria consciência e possua um conjunto de mais-valias como percepção, sentimentos e emoção (26). As características de um agente forte são: mobilidade, capacidade de se movimentar através da rede formada pelos seus pares, executando as tarefas de que foi incumbido; intencionalidade, capacidade que o agente apresenta para a definição de objectivos e das estratégias para os atingir; aprendizagem, capacidade que o agente ostenta de adquirir conhecimento, a actualização da base de conhecimento é feita através da assimilação de padrões de comportamento ou de preferências; competência, um agente é competente quando conduz com sucesso e eficiência as tarefas de que é incumbido, a competência está normalmente relacionada com a confiança depositada no agente por terceiros; veracidade, fala-se da veracidade de um agente quando este não fornece, de forma intencional, informação falsa; racionalidade, um agente racional não aceita realizar tarefas que lhe pareçam impossíveis de executar, contraditórias com os seus princípios o quando não são compensados em termos do risco, custo ou esforço; benevolência, um agente benevolente adopta como seus os objectivos de terceiros, desde que estes não entrem em conflito com os seus princípios de natureza ética e/ou deontológica, o que significa que não realizarão todas as tarefas que lhes sejam atribuídas; emotividade, certas características próprias do ser humano têm vindo a migrar e a estabelecer-se como parte constituinte dos agentes.

Alguns dos motivos que levaram ao desenvolvimento e evolução dos Sistemas Multi-Agente são: a complexidade e dimensão de alguns problemas, normalmente fora do alcance das capacidades individuais de um agente; o facto de alguns problemas estarem geograficamente e/ou funcionalmente distribuídos; a necessidade de ter informação e conhecimento dispersos; permite a interconexão de múltiplos sistemas (Legacy systems); permite paralelismo, robustez e escalabilidade.

Os Sistemas Multi-Agente são adequados para: construir sistemas para resolver problemas complexos, que não podem ser resolvidos por qualquer um agente individual; lidar com problemas que envolvem vários métodos de resolução de problemas, exigindo diferentes tipos de competências e conhecimentos, ou onde existem múltiplos pontos de vista; criar sistemas onde são necessárias formas dinâmicas de reorganização; tarefas em que a informação e os recursos são distribuídos.

Algumas das vantagens no uso dos Sistemas Multi-Agente são: a extensibilidade e flexibilidade, pois permite adicionar novos agentes, mover agentes para executar outras funcionalidades; a robustez e fiabilidade; a eficiência e rapidez do ponto de vista computacional, obtida através do paralelismo; normalmente são de mais fácil desenvolvimento e manutenção, devido à sua modularidade, pois permite desenvolver agente por agente; a reusabilidade dos agentes; normalmente comporta custos reduzidos, devido ao facto de serem do desenvolvimento e manutenção serem mais fáceis e devido à reusabilidade, flexibilidade e extensibilidade dos agentes.

Quando se constrói um Sistema Multi-Agente devem ser definidos um conjunto de características: arquitectura, coordenação, organização e comunicação (27) (28) (29).

Em termos de **arquitectura** um Sistema Multi-Agente pode ser aberto ou fechado. Num Sistema Multi-Agente aberto o sistema não tem um desenho/arquitectura pré-definido, apenas um conjunto de agentes no seu seio. Os agentes não têm necessariamente consciência da existência dos outros, logo, um mecanismo para identificar, localizar e procurar outros é uma exigência. Um Sistema Multi-Agente fechado apresenta uma arquitectura de desenho estático, com componentes e funcionalidades pré-definidos. As propriedades do sistema são conhecidas

antecipadamente: linguagem comum, cada agente pode ser desenvolvido como um perito, os agentes são (normalmente) cooperativos.

Ambas as arquitecturas apresentam vantagens e desvantagens. A arquitectura aberta tem como vantagens: a modularidade (agentes e/ou grupos de agentes são concebidos separadamente); a facilidade de manutenção e concepção ao longo do tempo; é mais flexível e tolerante a falhas; a sociedade de agentes é aberta e dinâmica. Por outro lado, a arquitectura aberta tem como desvantagens: a imprevisibilidade do comportamento geral do sistema; a variação de protocolos, linguagens e ontologias entre agentes; comportamentos maliciosos são mais difíceis de prever e evitar. Uma arquitectura fechada tem como vantagens: a possibilidade de distribuir carga e perícia; simplicidade e previsibilidade do sistema, uma vez que os agentes são conhecidos, a linguagem de interacção e protocolos são conhecidos, os agentes (normalmente) ostentam comportamentos cooperativos e partilham a arquitectura e *software*. Por outro lado, a arquitectura fechada tem como desvantagens: custos de manutenção mais elevados; menor tolerância a falhas; difícil interoperabilidade com outros sistemas.

A **coordenação** de um Sistema Multi-Agente pode ser: cooperativa, passa por um processo de tomada de decisão em que as partes envolvidas negoceiam, em termos de alcançarem um ou mais objectivos; competitiva, passa por um processo de decisão em que as partes envolvidas competem tendo em conta um único objectivo.

A **comunicação** de um Sistema Multi-Agente pode ser: directa, memória partilhada por todos os agentes, os agentes comunicam directamente entre si; assistida, passagem de mensagens entre agentes, existem agentes próprios para servir de meio de comunicação.

A **organização** de um Sistema Multi-Agente pode assumir várias formas: hierarquizada, existem agentes que têm autoridade sobre outros agentes e a comunicação é vertical; plana, os agentes têm a mesma importância, não existe nenhum agente com autoridade sobre outros e cada agente é visto como um especialista; organização baseada em mercados, os agentes competem pelos recursos através de leilões e de contratos; alocação por tarefas, alocação dinâmica dos agentes, redes de contratação.

Estas são as principais características (arquitectura, coordenação, comunicação e organização) para se definir um Sistema Multi-Agente.

3.2. Definição do SMA

O SMA que suporta a solução especificada no capítulo anterior é um Sistema Multi-Agente **fechado**, onde o desenho da **arquitectura** é estático, com todos os agentes e as suas respectivas funcionalidades pré-definidos. A **coordenação** entre os agentes é **cooperativa**, os agentes não competem entre si, cooperam de forma a atingir um objectivo comum. A **organização** é **plana**, sendo cada agente "visto" como um perito ou especialista numa funcionalidade e todos os agentes têm a mesma importância, nenhum agente tem autoridade sobre outro. A **comunicação** entre os agentes é **directa**, os agentes comunicam directamente entre si sem a intervenção de nenhum outro agente ou sistema e usam uma linguagem comum entre si.

3.2.1. Arquitectura

A Figura 2 traduz a arquitectura do SMA, representada ao lado da solução de FMS da WeDo Techonologies.

O SMA é composto por 3 agentes:

- Agente de ***profiling***: responsável por integrar os dados de entrada e construir o perfil (com atributos de identidade e atributos de comportamento) para cada subscritor;
- Agente de **detecção**: responsável por detectar fraudulentos previamente banidos que reentrem na rede, usando para isso os atributos de identidade calculados pelo agente de profiling;

- Agente de **KDD** (Knowledge Discovery in Database): responsável por, com base nos atributos de comportamento calculados pelo agente de profiling, tentar extrair conhecimento acerca do comportamento de utilizadores fraudulentos.

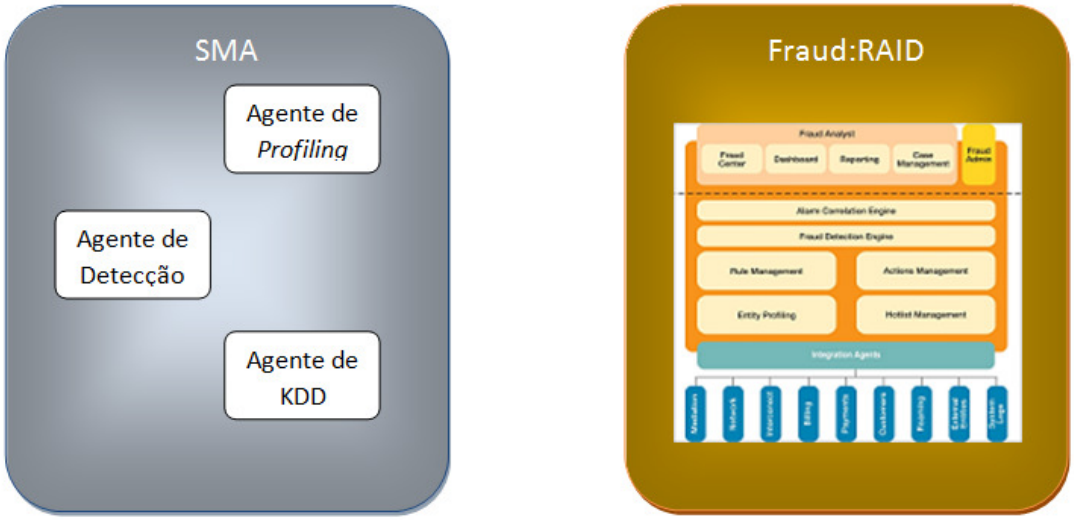


Figura 2 – Arquitectura do SMA

3.2.2. Processo

O processo entre o SMA e a solução de FMS, visto de uma forma global, é composto por cinco interacções, explicadas de seguida com imagens a complementar para melhor se perceber as interacções:

1) Numa primeira fase, a solução de FMS reencaminha dados do conteúdo dos CDR (Call Detail Record) para o agente de profiling. Este agente transforma e armazena estes dados necessários para construir o perfil (calculando atributos de identidade e comportamento) para cada subscritor.
2) Numa segunda fase, a solução de FMS deve indicar ao agente de detecção e de KDD todos os fraudulentos que detectou e baniu da rede.

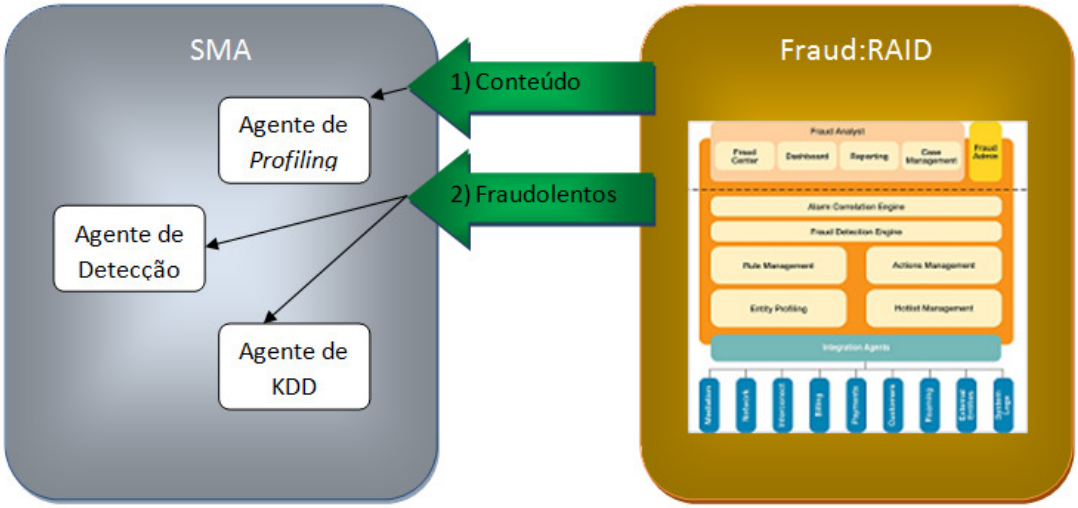


Figura 3 – Interacção SMA - Fraud:RAID, passos 1 e 2

3) Ambos os agentes usaram esta informação para enviar um pedido ao agente de profiling, ao qual este retorna o perfil construído até ao momento do subscritor fraudulento. Paralelamente, cada um dos agentes prossegue com as suas tarefas.

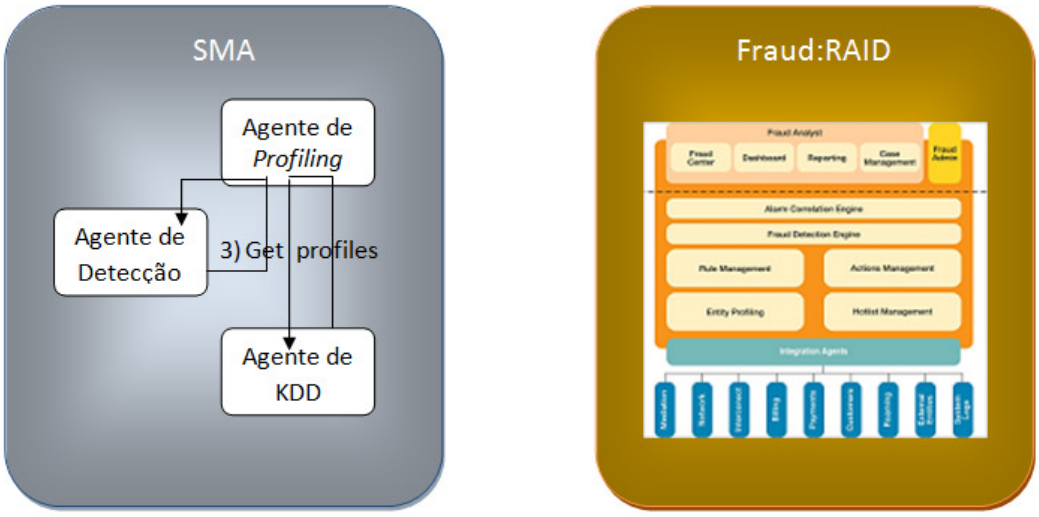


Figura 4 – Interacção SMA - Fraud:RAID, passo 3

4) O agente de detecção usará os atributos de identidade do perfil do fraudulento com o objectivo de detectar se o fraudulento reentra na rede, caso tal aconteça

deve construir um novo processo acerca do novo suspeito de fraude e enviar este processo para a solução de FMS.

5) O agente de KDD usará os atributos de comportamento do perfil do fraudulento para enriquecer uma base de conhecimento, com o objectivo de detectar padrões/comportamentos similares, sendo os resultados deste processo reencaminhados para a solução de FMS para que os analistas possam ter acesso a esta informação (sob a forma de texto, tabelas, gráficos).

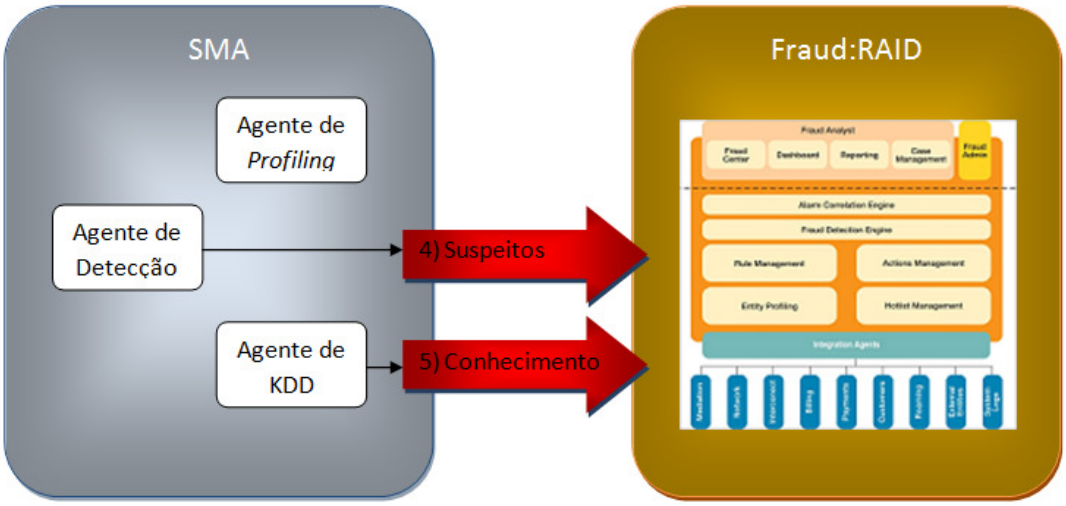


Figura 5 – Interacção SMA - Fraud:RAID, passos 4 e 5

4. Agente de *profiling*

A nova abordagem desenvolvida no capítulo anterior para resolver o problema em questão baseia-se na técnica de *profiling*. Este capítulo começa com uma introdução a esta técnica, seguindo-se as secções que enquadram esta técnica na solução, identificando quais os dados de entrada, os perfis que vão ser construídos, os campos que vão constituir cada perfil e por fim, como estes perfis vão ser construídos.

4.1. Introdução ao *profiling*

O profiling é uma técnica **auxiliar** da investigação criminal, que pertence ao domínio da Psicologia Forense. Consiste num processo de inferência das características de indivíduos responsáveis por actos criminais (30). A técnica de profiling deve ser usada como uma extensão da análise criminal, construindo perfis baseado num trabalho já realizado (31). A ideia base a manter sobre esta técnica é: o profiling é uma técnica auxiliar que deve ser usada para **complementar** trabalho prévio, não para o substituir.

Actualmente, o *profiling* é uma técnica muito usada, implementada em forças policiais por todo o mundo. Peritos nesta técnica, como McCrary (32) ou Wrightsman (33) , sublinham o facto de esta técnica ter excelentes resultados na previsão de factores e acontecimentos.

Apesar de esta técnica ter sido desenvolvida sobre o domínio da Psicologia Forense, o profiling é uma técnica que tem actualmente diversas aplicações. Como tal, é natural que seja um conceito que começou a ser usada com sucesso em vários sistemas de informação. Existem aplicações desta técnica em sistemas informáticos para determinar redes sociais (34) (35), análise de comportamento em grande escala (36), aplicações de segurança (37) (38) e modelos de previsão de transacção de dados (39).

4.2. Dados de *input*

Um CDR (Call Detail Record) é um registo informático produzido automaticamente por uma central telefónica, que contém informação acerca de eventos (chamada de voz, SMS) processados por essa central. Os CDR são transferidos electronicamente de várias centrais para um ponto central de processamento, onde software direccionado para o processamento destes eventos trata os CDR. Em termos de telecomunicações este software, geralmente chamado de BSS (Billing Support System), é responsável pela facturação dos eventos, calculando o custo de cada evento.

Apesar de inicialmente os CDR serem usados para efeitos de facturação, actualmente os operadores usam os CDR como *input* para aplicações com outros propósitos, como por exemplo, aplicações de garantia de receita, análise de desempenho de rotas ou mesmo para detecção de redes sociais (40).

Tipicamente, um CDR é um ficheiro de texto em que cada linha corresponde a um evento processado na central telefónica. Em alguns casos o CDR encontra-se num formato comprimido, para efeitos de optimização de espaço de armazenamento. Em termos de estrutura, normalmente um CDR tem uma estrutura definida: tem um número de campos definido, com uma ordem estabelecida e os campos são separados por um caracter específico.

A estrutura de um CDR varia consoante o operador, tanto ao nível do número de campos, como ao nível da ordenação dos mesmos e até o caracter separador varia. Contudo, há um conjunto de campos que, devido à sua importância para efeitos de facturação, estão normalmente presentes em qualquer estrutura de CDR:

- A_NUMBER – identifica o originador do evento;
- B_NUMBER – identifica o receptor do evento;
- EVENT_DATE – data do início do evento;
- EVENT_TYPE – identifica o tipo do evento, normalmente é um código: 1 (Voz), 2 (SMS), 3 (Dados);

- EVENT_AMOUNT – medida do evento: se o evento for uma chamada a medida será em segundos (123 segundos), se o evento for uma SMS a medida será o tamanho da SMS (43 caracteres).

Outros campos, não necessários para a facturação mas também comuns nas estruturas dos CDR, são:

- CELL_ID – identifica a célula de rede[6] que processou o evento;
- TELEPHONE_ID – identifica o aparelho (telemóvel) usado, através do seu IMEI.

Nas telecomunicações existem alguns conceitos que é necessário explicar, pois serão usados a partir deste momento:

- IMSI (International Mobile Subscriber Identity) – identificador único, a nível mundial, de todos os utilizadores de uma rede GSM ou UMTS[7]. Está guardado no cartão SIM.
- IMEI (International Mobile Equipment Identity) – identificador único, a nível mundial, de todos os telemóveis[8]. Normalmente está impresso na parte de trás da bateria do telemóvel. Também pode ser obtido marcando o seguinte código no telemóvel: *#06#.

A informação proveniente dos CDR será o *input* para todo o trabalho desenvolvido. Serão estes os dados que irão ser usados para construir os perfis dos subscritores.

4.3. Perfil: identidade e comportamento

[6] Normalmente é um código que permite identificar geograficamente (coordenadas de latitude e longitude) uma célula de rede.

[7] GSM (Global System for Mobile communications) e UMTS (Universal Mobile Telecommunications System) são tecnologias usadas nas telecomunicações móveis. GSM é normalmente conhecida por 2G e UMTS por 3G.

[8] O IMEI é um número único usado na rede GSM para identificar aparelhos válios. Pode ser usado para impeder que um telemóvel roubado tenha acesso à rede. Por exemplo, se um telemóvel é roubado, o proprietário pode contactar o seu operador e instruir o operador para restringir o acesso do telemóvel com o seu número IMEI. Isto torna o telemóvel inútil, mesmo que o cartão SIM seja trocado.

Conforme foi descrito na definição da nova abordagem, o perfil de um subscritor será constituído por atributos de identidade e de comportamento. Relembrando: os atributos de identidade contêm informação que permitem, na segunda fase, identificar o fraudulento como indivíduo; os atributos de comportamento contêm informação que identificam o fraudulento como pertencente a um grupo de risco, serão usados na terceira fase para extracção de conhecimento.

Olhando para os dados disponíveis para construir o perfil de um subscritor, o conteúdo dos CDRs, vemos que a informação contida nesses dados diz respeito a interacção entre subscritores do operador e outras entidades (que podem ser subscritores do mesmo operador ou subscritores de outra operador). Nesses dados temos o **IMEI**, que deve ser usado como um **atributo de identificação** pois identifica o aparelho que o fraudulento utilizou para cometer fraude. Pode-se dizer que é a arma do crime. O **CELL_ID** pode ser usado como elemento **identificador mas apenas como um indicador** auxiliar, pois identifica a área de acção do fraudulento. Contudo estes atributos seriam claramente insuficientes para detectar que um fraudulento reentra na rede do operador, pois normalmente o fraudulento livra-se do telemóvel com que foi detectado inicialmente porque já existe métodos nos FMS para usarem listas negras de IMEIs que foram associados a actividades fraudulentas. Então o que resta, que possa servir como atributo identificador, destes dados que registam a actividade de um subscritor fraudulento? Precisamente isso: **a actividade do fraudulento, isto é, a sua rede social**. Estes dados contêm todas as entidades com ele comunicou. Então daí é possível calcular qual a sua rede social e usa-la para detectar se o fraudulento reentra na rede.

A utilização da rede social como atributo de identificação tem uma grande vantagem: após um subscritor ser detectado como fraudulento e banido da rede do operador, não será preciso monitorizar todos os subscritores para tentar descobrir se é esse mesmo subscritor a tentar reentrar na rede, basta vigiar os subscritores que tenham actividade com os elementos que constituem a rede social do subscritor fraudulento banido.

Aplicando a ARS nesta nova abordagem permitirá reunir informação suficiente para a identificação de fraudulentos numa fase posterior e também para extrair conhecimento acerca do seu comportamento.

4.4. Redes Sociais

A Análise de Redes Sociais (ARS) tem vindo a emergir como um paradigma essencial em várias áreas como a sociologia, psicologia, tecnológica e mesmo criminal. O princípio deste paradigma assenta no facto de que considera que os atributos de um indivíduo numa rede social são menos importantes que as suas ligações ou relações com outros indivíduos da rede (41). Explorar a natureza e a força das ligações e a estrutura de uma rede social são um importante passo para compreender e explicar a dinâmica e fenómenos das redes sociais no mundo real.

Alguns conceitos muito básicos acerca do paradigma da ARS (42): Uma rede social consiste num conjunto finito de actores e das relações estabelecidas entre os mesmos. O enfoque da ARS constitui-se nos laços relacionais entre os actores que estão ligados uns aos outros através de vínculos sociais. Existem diferentes tipos de vínculos: biológico (família) e social (amizade); associação e afiliação (clubes e associações); interacção profissional (trabalho); física (cidade, bairro, rua, prédio). O método mais comum de representação de uma rede social é através do uso de um grafo, em que os nós dos grafos representam os actores e as ligações entre os nós representam uma relação entre os mesmos. Esta ligação pode ser não direccional, unidireccional ou bidireccional, e normalmente tem um peso associado que pretende representar a força da relação entre os actores.

Existem aplicações bem sucedidas da ARS em várias áreas com várias finalidades. A descoberta de comunidades é um dos principais objectivos da aplicação da ARS: descoberta de comunidades na internet (email, chats, fóruns e partilha de informação) (43) (44) (45), descoberta de comunidades em actividades diárias comuns para fins comerciais e de marketing (46) (47). Existe mesmo aplicações para tentar calcular

futuras ligações sociais dentro de uma rede (48). Um estudo da aplicação da ARS na área criminal (49) procura obter informação útil para combater uma rede criminal como por exemplo: quem é o elemento central da rede? Que subgrupos existem? Quais os padrões da interacção entre os seus elementos? Que elemento causaria a ruptura da rede caso fosse removido da rede?

Na área das telecomunicações também já existem aplicações da ARS, mas com outros propósitos que não a detecção de fraude, como por exemplo, para fins comerciais (50) ou para evitar perda de clientes, ou controlo de *churn*[9] (41).

4.5. Processos

O agente de *profiling* recebe dois tipos de *input*:

- O conteúdo dos CDRs que recebe do Fraud:RAID na primeira fase da iteração entre o SMA e o Fraud: RAID (ver *Figura 3 – Interacção SMA - Fraud:RAID, passos 1 e 2*);
- Na terceira iteração recebe a indicação de subscritores dos agentes de detecção e de extracção de conhecimento (ver *Figura 4 – Interacção SMA - Fraud:RAID, passo 3*), para os quais é preciso construir um perfil, seja porque este subscritor foi detectado como fraudulento pelo Fraud:RAID, seja porque são subscritores que o agente de detecção considera como suspeitos e para os quais é necessário construir o perfil.

O agente de profiling é composto por dois processos: sumarização e construção do perfil. A Figura 6 pretende traduzir o funcionamento do agente de *profiling*.

Numa primeira fase, o agente de *profiling* recebe o conteúdo dos CDRs e executa o processo de sumarização, efectuando as transformações necessárias para depois armazenar o resultado, o sumário da actividade dos subscritores, num repositório. Este repositório contém os sumários para todos os subscritores. Numa segunda fase, o

[9] Em telecomunicações, o conceito de *churn* é entendido como a perda de clientes de um operador para outro operador.

agente de *profiling* recebe indicação de subscritores para os quais é necessário construir um perfil, recolhe os sumários para esses subscritores a partir do repositório e constrói um perfil, composto por atributos de identidade e de comportamento, devolvendo este perfil ao agente de detecção e de extracção de conhecimento. Os sumários e os perfis são o *output* deste agente.

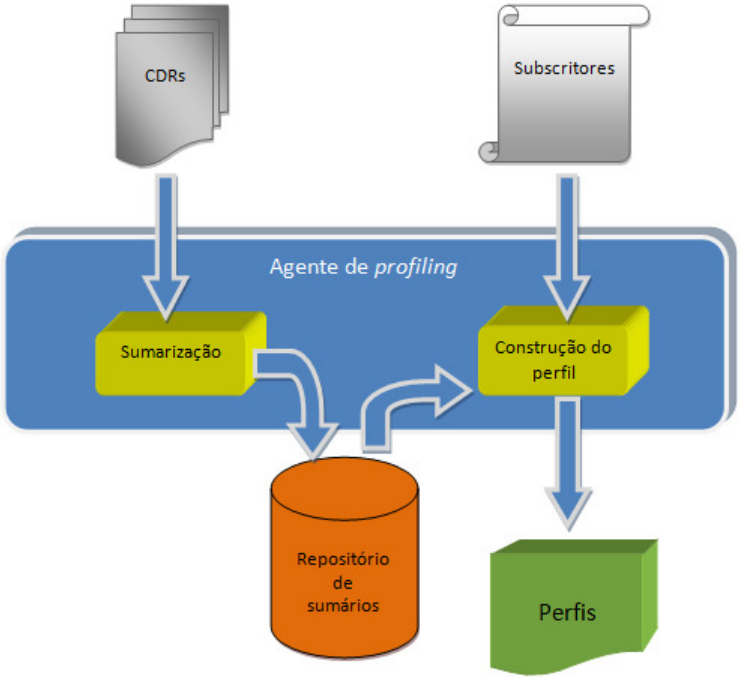


Figura 6 – Processos de *Profiling*

4.4.1. Processo de sumarização

O processo de sumarização recebe como *input* o conteúdo dos CDRs: eventos da actividade dos subscritores na rede do operador. Cada evento é constituído por um número de campos que varia consoante o operador. Conforme foi explicado numa secção anterior (4.2 Dados de *input*), há um conjunto de campos que estão sempre presentes. É com este conjunto de campos que o processo de sumarização irá trabalhar. Relembrando, cada evento é constituído pelos campos representados na Tabela 1.

Campo	Descrição	Obrigatório
A_NUMBER	Originador do evento	Sim
B_NUMBER	Receptor do evento	Sim
EVENT_DATE	Data do início do evento	Sim
EVENT_TYPE	Tipo do evento (SMS, Voz, etc.)	Sim
EVENT_AMOUNT	Medida do evento	Sim
CELL_ID_A	Célula de rede que processou o evento na origem	Não
IMEI_A	Aparelho usado na origem	Não
CELL_ID_B	Célula de rede que processou o evento no destino	Não
IMEI_B	Aparelho usado no destino	Não

Tabela 1 – Estrutura de um evento de um CDR

São estes os campos que a partir de agora consideramos que constitui cada evento do CDR.

O processo de sumarização guarda no repositório os sumários para cada subscritor, por dia. A decisão de guardar os sumários por dia relaciona-se com aspectos de *housekeeping* do repositório, isto é, para que possa ser mais fácil manter o repositório quando se pretender apagar dados antigos. Nesta fase pressupomos que os sumários serão guardados relativamente aos últimos 30 dias. Os dias anteriores vão sendo apagados. A razão pela qual se deve apagar dados antigos é porque as redes sociais e o comportamento evoluem ao longo do tempo, entrando e saindo elementos da rede social, alteração de serviços usados e de períodos de actividade. Como tal devem ser considerados apenas os dados mais recentes, descartando os mais antigos, para que a rede social e o comportamento representados no final do processo sejam o mais actual possível. A Tabela 2 traduz os campos que constituem um sumário.

Conforme já foi referido anteriormente, os CDRs são criados para efeitos de facturação. Assim há aqui um aspecto importante que deve ser mencionado: os CDRs apenas contêm informação acerca dos eventos gerados dentro da rede. Ou seja, temos acesso a toda a actividade gerada por um subscritor, mas quanto à actividade recebida de um subscritor só temos essa informação se o evento tiver sido originado por ou outro subscritor que pertença ao operador. Para melhor se perceber este aspecto e como influencia o nosso processo de sumarização, vejamos a Tabela 3, que traduz

todos os cenários possíveis. A1 e A2 são subscritores do operador, B1 e B2 são subscritores de outro operador.

Campo	Descrição
SUBSCRIBER_ID	Identifica o subscritor a que pertence sumário. É um identificador único do subscritor.
DAY	Identifica o dia a que pertence o sumário. É uma data no formato yyyy-mm-dd (ano-mês-dia).
CALLED	Eventos originados pelo subscritor. É uma **lista** de pares <C,N>, em que C é um contacto de destino e N é o número de eventos para esse contacto.
RECEIVED	Eventos recebidos pelo subscritor. É uma **lista** de pares <C,N>, em que C é um contacto de origem e N é o número de eventos recebidos desse contacto.
CELL_USAGE	Células usadas pelo subscritor. É uma **lista** de pares <C,N>, em que C é o identificador da célula e N é o número de eventos ocorridos nessa célula.
SERVICES_USAGE	Serviços usados pelo subscritor. É uma **lista** de pares <S,N>, em que S é o identificador do serviço e N é o número de eventos que usaram esse serviço.
IMEI_LIST	Aparelhos usados pelo subscritor. É uma **lista** de IMEIs.
ACTIVITY	Actividade do subscritor. É uma **lista** de períodos em o subscritor esteve activo. Consideramos que um dia está dividido em períodos de 10 minutos, sendo que um dia é constituído por 144 períodos. Dividir um dia em períodos de 1 minuto seria demasiado granular. Dividir um dia em períodos de 1 hora seria muito pouco granular. É uma lista de 0 e 1 (tamanho fixo de 144, inicialmente toda preenchida a 0), cada posição indica se o subscritor esteve activo nesse período (1) ou não (0).

Tabela 2 – Estrutura de um sumário

De	Para	Sumarização
A1	A2	Acesso a informação de um registo: A1 → A2. Processo de sumarização deve actualizar sumários dos subscritores A1 e A2.
A1	B1	Acesso a informação de um registo A1 → B1. Processo de sumarização apenas deve actualizar sumário de A1.
B1	A1	Apesar de esta situação envolver um subscritor da rede do operador, não temos acesso a qualquer informação.
B1	B2	Naturalmente, para esta situação não temos acesso a qualquer informação.

Tabela 3 – Cenários de sumarização

Apenas o primeiro cenário, normalmente designado por um evento *OnNet* (um evento de um subscritor do operador para outro subscritor do operador) exige um processamento extra, pois é preciso actualizar o sumário do subscritor de destino.

Os seguintes algoritmos traduzem o processo de sumarização para cada evento que recebe. De notar que o algoritmo contempla a actualização do sumário do subscritor de destino caso seja um evento *OnNet*, e que também contempla o facto de nem todos os campos do evento serem obrigatórios (CELL_ID e IMEI).

```
1:  sumariza(evento){
2:
3:     /* PASSO_1: Ir buscar os campos do evento */
4:     A_NUMBER = evento.getA_NUMBER();
5:     B_NUMBER = evento.getB_NUMBER();
6:     EVENT_DATE = evento.getEVENT_DATE();
7:     EVENT_TYPE = evento.getEVENT_TYPE();
8:     EVENT_AMOUNT = evento.getEVENT_AMOUNT();
9:     CELL_ID_A = evento.getCELL_ID_A();
10:    IMEI_A = evento.getIMEI_A();
11:    CELL_ID_B = evento.getCELL_ID_B();
12:    IMEI_B = evento.getIMEI_B();
13:
14:    /* PASSO_2: Calcular o dia no formato ano-mês-dia */
15:    EVENT_DAY=getDAY(EVENT_DATE);
16:
17:    /* PASSO_3:
18:     * Verificar se o sumário já existe para
19:     * o subscritor A_NUMBER para o dia EVENT_DAY
20:     * no repositório de sumários */
21:    Se ( REPOSITORIO.contem_sumario(A_NUMBER,EVENT_DAY) )
22:    Então
23:       /* PASSO_3a: Ir buscar o sumário para actualizar */
24:       SUMARIO = REPOSITORIO.getSUMARIO(A_NUMBER,EVENT_DAY);
25:    Senão
26:       /* PASSO_3b: Construir novo perfil */
27:       SUMARIO = new SUMARIO();
28:    Fim Se
```

Algoritmo 1 – Sumarização: passos 1, 2 e 3

1º passo: Ir buscar os campos ao evento.

2º passo: Calcular o dia (formato ano-mês-dia) a que pertence o evento a partir do campo EVENT_DATE (formato ano-mês-dia horas:minutos:segundos).

3º passo: Verificar se o sumário já existe para o subscritor e para a data que estamos a tratar. Caso não exista, é necessário criar um novo sumário, onde as listas (CALLED, RECEIVED, CELL_USAGE, SERVICES_USAGE, IMEI_LIST) estão vazias e a lista de ACTIVITY é uma lista de 144 zeros.

```
30:    /* PASSO_4: tratar destino */
31:    CALLED = SUMARIO.getCALLED_LIST();
32:    Se ( CALLED.contem(B_NUMBER) )
33:    Então
34:       NUMERO_EVENTOS = CALLED.get(B_NUMBER)
35:       CALLED.set(B_NUMBER,NUMERO_EVENTOS+1)
36:    Senão
37:       CALLED.adiciona(B_NUMBER,1);
38:    Fim Se
39:    SUMARIO.setCALLED_LIST(CALLED);
40:
41:    /* PASSO_5: tratar célula de rede */
42:    Se ( CELL_ID_A tem valor )
43:    Então
44:       CELL_USAGE = SUMARIO.getCELL_USAGE_LIST();
45:       Se ( CELL_USAGE.contem(CELL_ID_A) )
46:       Então
47:          NUMERO_EVENTOS = CELL_USAGE.get(CELL_ID_A)
48:          CELL_USAGE.set(CELL_ID_A,NUMERO_EVENTOS+1)
49:       Senão
50:          CELL_USAGE.adiciona(CELL_ID_A,1);
51:       Fim Se
52:       SUMARIO.setCELL_USAGE_LIST(CELL_USAGE);
53:    Fim Se
54:
55:    /* PASSO_6: tratar serviço */
56:    SERVICE_USAGE = SUMARIO.getSERVICE_USAGE_LIST();
57:    Se ( SERVICE_USAGE.contem(EVENT_TYPE) )
58:    Então
59:       NUMERO_EVENTOS = SERVICE_USAGE.get(EVENT_TYPE)
60:       SERVICE_USAGE.set(EVENT_TYPE,NUMERO_EVENTOS+1)
61:    Senão
62:       SERVICE_USAGE.adiciona(EVENT_TYPE,1);
63:    Fim Se
64:    SUMARIO.setSERVICE_USAGE_LIST(SERVICE_USAGE);
65:
```

Algoritmo 2 – Sumarização: passos 4, 5 e 6

4º passo: Verificar se o destino do evento já existe na lista de destinos. Caso já exista é

necessário incrementar o número de eventos para esse destino. Caso não exista deve ser inserida uma nova entrada na lista.

5º passo: Verificar se existe um valor para a variável que contém a célula de rede. Se existir, então verificar se a lista de células contém esta célula. Caso contenha é necessário incrementar o número de eventos para essa célula. Caso não contenha deve ser inserida uma nova entrada na lista.

6º passo: Verificar se o tipo de serviço já existe na lista de serviços usados. Caso já exista é necessário incrementar o número de eventos para esse serviço. Caso não exista deve ser inserida uma nova entrada na lista.

```
66:    /* PASSO_7: tratar IMEI */
67:    Se ( IMEI_A tem valor )
68:    Então
69:       IMEI_LIST = SUMARIO.getIMEI_LIST();
70:       Se ( Não ( IMEI_LIST.contem(IMEI_A) ) )
71:       Então
72:          IMEI_LIST.adiciona(IMEI_A);
73:       Fim Se
74:       SUMARIO.setIMEI_LIST(IMEI_LIST);
75:    Fim Se
76:
77:    /* PASSO_8: registar actividade */
78:    ACTIVITY = SUMARIO.getACTIVITY();
79:    INDICE_PERIODO = getINDICE_PERIODO(EVENT_DATE);
80:    ACTIVITY[INDICE_PERIODO]=1;
81:    SUMARIO.setACTIVITY(ACTIVITY);
82:
83:    /* PASSO_9: guardar sumário */
84:    REPOSITORIO.setSUMARIO(A_NUMBER,EVENT_DAY,SUMARIO);
85:
```

Algoritmo 3 – Sumarização: passos 7, 8 e 9

7º passo: Verificar se existe um valor para a variável que contém o IMEI. Se existir, então verificar se a lista de IMEIs contém este IMEI. Caso não contenha deve ser inserida uma nova entrada na lista.

8º passo: Descobrir a que período corresponde a data de ocorrência do evento, usando a função *getINDICE_PERIODO(EVENT_DATE)*. Actualizar a lista de actividade.

9º passo: Actualizar o sumário para este subscritor no repositório.

O passo que se segue é para verificar se o B_NUMBER pertence à rede do operador, tratando-se assim de um evento *OnNet*.

```
86:   /* PASSO_10: verificar se B_NUMER pertence ao operador */
87:   Se ( B_NUMBER pertece ao operador )
88:   Então
89:   /* PASSO_11:
90:      * Verificar se o sumário já existe para
91:      * o subscritor B_NUMBER para o dia EVENT_DAY
92:      * no repositório de sumários */
93:     Se ( REPOSITORIO.contem_sumario(B_NUMBER,EVENT_DAY) )
94:     Então
95:       /* PASSO_11a: Ir buscar o sumário para actualizar */
96:       SUMARIO = REPOSITORIO.getSUMARIO(B_NUMBER,EVENT_DAY);
97:     Senão
98:       /* PASSO_11b: Construir novo perfil */
99:       SUMARIO = new SUMARIO();
100:    Fim Se
101:
```

Algoritmo 4 – Sumarização: passos 10 e 11

Caso o B_NUMBER pertença à rede do operador, é necessário replicar a lógica para actualizar o sumário do subscritor B_NUMBER. Apenas o passo 12 será apresentado, onde em vez de actualizar lista de destinos actualiza a lista de origens. Toda a lógica restante é igual à explicada para o subscritor A_NUMBER. O algoritmo completo encontra-se no *Anexo I – Algoritmo de sumarização*.

```
101:
102:    /* PASSO_12: tratar origem */
103:    RECEIVED = SUMARIO.getRECEIVED_LIST();
104:    Se ( RECEIVED.contem(A_NUMBER) )
105:    Então
106:      NUMERO_EVENTOS = RECEIVED.get(A_NUMBER)
107:      RECEIVED.set(A_NUMBER,NUMERO_EVENTOS+1)
108:    Senão
109:      RECEIVED.adiciona(A_NUMBER,1);
110:    Fim Se
111:    SUMARIO.setRECEIVED_LIST(RECEIVED);
112:
```

Algoritmo 5 – Sumarização: passo 12

Os resultados deste processo de sumarização, os sumários, serão guardados num repositório de sumários para puderem ser usados no segundo processo do agente de *profiling*, a construção do perfil. Este processo é detalhado na seguinte secção.

4.4.2. Processo de construção do perfil

A construção de um perfil é realizada quando o agente de *profiling* recebe a indicação de que um subscritor foi detectado como fraudulento e, consequentemente, banido

da rede do operador, ou de um subscritor que o agente detecção considera suspeito de ser um fraudulento previamente banido e precisa do seu perfil para determinar se efectivamente se trata do mesmo subscritor. Neste momento o agente deve construir o perfil para esse subscritor, calculando os seus atributos de identidade e de comportamento. Este perfil será devolvido aos outros dois agentes, de detecção e de extracção de conhecimento, possam usar os seus atributos para atingir os seus objectivos.

O primeiro passo deste processo de construção de perfil do agente de *profiling* é ir ao repositório de sumários obter os sumários para o subscritor. Normalmente serão 30 sumários, uma vez que tal como foi referido anteriormente, pressupomos que os sumários serão guardados relativamente aos últimos 30 dias. A partir dos dados destes 30 sumários este processo constrói os atributos de e de comportamento (serão usados pelo agente de extracção de conhecimento).

No caso em que se constrói um perfil devido à indicação da detecção de um fraudulento por parte do Fraud:RAID todos os atributos, de identidade e de comportamento, serão calculados e serão usados pelo agente de detecção e pelo agente de extracção de conhecimento, respectivamente.

No caso em que se constrói um perfil por pedido do agente de detecção por se tratar de um suspeito apenas os atributos de identidade serão calculados e serão usados no agente de detecção, pois a detecção é feita com base na identidade.

A Tabela 4 traduz a composição de um perfil, identificando os atributos que o compõem, bem como o seu tipo (identidade ou comportamento).

O cálculo de cada um destes atributos será explicado individualmente ao longo desta secção. Conforme já foi referido, quando se calcula o perfil para um subscritor o primeiro passo é obter todos os sumários para esse subscritor, normalmente serão 30 sumários, mas pode acontecer que sejam menos. Ao longo da explicação do cálculo para cada um dos atributos, refere-se a N como o número de sumários disponíveis para esse mesmo subscritor.

Atributo	Tipo	Descrição
SUBSCRIBER_ID		Identifica o subscritor a que pertence perfil. É um identificador único do subscritor.
SOCIAL_NET_IN	Identidade	Lista de números que mais frequentemente contactam o subscritor.
SOCIAL_NET_OUT	Identidade	Lista de números que o subscritor mais frequentemente contacta.
CELL_LIST	Identidade	Lista de células de rede mais utilizadas pelo subscritor.
IMEI_LIST	Identidade	Lista de aparelhos usados pelo subscritor.
ACTIVITY	Comportamento	Percentagem média de tempo em que o subscritor tem actividade.
CALL_DISPERSION_50	Comportamento	Número mínimo de contactos que são necessários para atingir 50% da actividade de saída do subscritor.
CALL_DISPERSION_70	Comportamento	Número mínimo de contactos que são necessários para atingir 70% da actividade de saída do subscritor.
CALL_DISPERSION_90	Comportamento	Número mínimo de contactos que são necessários para atingir 90% da actividade de saída do subscritor.
IMEI_STUFFING	Comportamento	Número de aparelhos usados pelo subscritor.
CELL_DISPERSION_50	Comportamento	Número mínimo de células de rede que são necessárias para atingir 50% da actividade de saída do subscritor.
CELL_DISPERSION_70	Comportamento	Número mínimo de células de rede que são necessárias para atingir 70% da actividade de saída do subscritor.
CELL_DISPERSION_90	Comportamento	Número mínimo de células de rede que são necessárias para atingir 90% da actividade de saída do subscritor.
IN_OUT_RATIO	Comportamento	Rácio entre a actividade de entrada e de saída do subscritor.

SERVICE_DISPERSION_50	Comportamento	Número mínimo de serviços que são necessários para atingir 50% da actividade de saída do subscritor.
SERVICE_DISPERSION_70	Comportamento	Número mínimo de serviços que são necessários para atingir 70% da actividade de saída do subscritor.
SERVICE_DISPERSION_90	Comportamento	Número mínimo de serviços que são necessários para atingir 90% da actividade de saída do subscritor.

Tabela 4 – Estrutura de um perfil

Os dois primeiros atributos, SOCIAL_NET_IN e SOCIAL_NET_OUT, têm como objectivos representar a rede social do subscritor fraudulento. Nem todos os contactos de um fraudulento estarão presentes na rede social representada por estes dois atributos. É necessário encontrar uma medida para aplicar às ligações sociais de forma a determinar quais as mais relevantes para constituir a rede social.

Nos vários artigos referidos na secção *4.4 Redes Sociais* são apresentadas algumas medidas para determinar a força da ligação entre dois elementos de uma rede social, como por exemplo: número de interacções, densidade das interacções, frequência das interacções.

Na explicação dos atributos SOCIAL_NET_IN e SOCIAL_NET_OUT, é referida a frequência como a chave para determinar a rede social. A razão pela qual é dada prioridade à frequência é porque estes atributos de identidade serão usados para detectar uma possível reentrada de um subscritor fraudulento na rede do operador. Assim, torna-se óbvio que os contactos com quem o subscritor tem mais frequência sejam provavelmente os primeiros a voltar a ter interacção com o subscritor, sendo logicamente os de maior interesse em guardar no perfil do subscritor fraudulento.

SOCIAL_NET_IN

Para calcular este atributo são reunidas as listas RECEIVED dos N sumários. Estas listas contêm os eventos que foram recebidos pelo subscritor, são uma lista de pares <CONTACT, NUMBER_OF_EVENTS>, em que CONTACT é um contacto de origem e

NUMBER_OF_EVENTS é o número de eventos originado por esse contacto. Neste caso o NUMBER_OF_EVENTS é descartado, pois o que interessa é descobrir quem contacta o subscritor mais frequentemente, e para calcular a frequência considera-se o número de presenças por dia. Ou seja, considera-se que o contacto que contacta o subscritor mais frequentemente é o que tiver mais presenças nas N listas RECEIVED. Assim, todos os contactos que estiverem presentes em pelo menos 30% das N listas RECEIVED, ou seja, que contactaram o subscritor em 30% dos dias em análise, são inseridos na lista SOCIAL_NET_IN.

SOCIAL_NET_OUT

Para calcular este atributo são reunidas as listas CALLED dos N sumários. Estas listas contêm os eventos que foram originados pelo subscritor, são uma lista de pares <CONTACT, NUMBER_OF_EVENTS>, em que CONTACT é um contacto de destino e NUMBER_OF_EVENTS é o número de eventos para por esse contacto. Tal como no exemplo anterior, o NUMBER_OF_EVENTS é descartado, pois considera-se que a frequência é determinada pelo número de presenças por dia. Neste caso, considera-se que o contacto que o subscritor mais frequentemente contacta é o que tiver mais presenças nas N listas CALLED. Assim, todos os contactos que estiverem presentes em pelo menos 30% das N listas CALLED, ou seja, que o subscritor contactou em 30% dos dias em análise, são inseridos na lista SOCIAL_NET_OUT.

CELL_LIST

Para calcular a lista de células de rede mais utilizadas são reunidas as listas CELL_USAGE dos N sumários. Relembrando, CELL_USAGE é uma lista de pares <CELL_ID, NUMBER_OF_EVENTS>, em que CELL_ID é o identificador da célula de rede e NUMBER_OF_EVENTS é o número de eventos ocorridos nessa célula. Agrega-se as N listas por CELL_ID numa só, somando o número de eventos para esse mesmo CELL_ID. Esta lista agregada de pares <CELL_ID, NUMBER_OF_EVENTS> é ordenada por ordem decrescente do número de eventos, para que na primeira posição esteja a célula de rede com maior número de eventos e na última posição esteja a célula de rede com menor número de eventos. Calcula-se a soma de toda a lista agregada, de forma a obter o total de eventos, TOTAL_NUMBER_OF_EVENTS, para todas as células de rede.

De seguida percorre-se a lista agregada, que está ordenada por ordem decrescente de número de eventos, somando o número de eventos, até que o total seja superior ou igual a 90% do valor de TOTAL_NUMBER_OF_EVENTS. Quando este valor for atingido as restantes células de rede são apagadas da lista agregada. A lista de células de rede contida na lista agregada é a CELL_LIST.

A razão pela qual se considera apenas 90% do número de eventos total para calcular esta lista é para eliminar valores residuais de utilização de células, sendo o resultado final uma lista de células mais utilizadas pelo subscritor, representando a sua zona de acção. Apesar de não ser um atributo que por si só permita identificar que o fraudulento reentrou na rede, pode ser um indicativo de identidade bastante útil.

IMEI_LIST

O cálculo deste atributo é bastante simples: é a junção das listas IMEI_LIST dos N sumários. Uma IMEI_LIST de um sumário é uma lista de IMEIs. Estas N listas de IMEIs são agregadas numa só, sem IMEIs repetidos, obviamente.

Esta lista representa as “armas do crime” utilizadas pelo fraudulento e constituem um poderoso atributo identificativo.

ACTIVITY

Para o cálculo deste atributo são usadas as listas ACTIVITY dos N sumários. Estas listas são listas de 144 períodos de inactividade ou actividade, 0 ou 1, respectivamente. Para cada uma destas listas é calcula a percentagem de períodos activos: soma dos 144 períodos / 144. O valor final deste atributo é a média destas N percentagens.

Este atributo de comportamento pretende representar a percentagem de tempo que um subscritor está activo.

CALL_DISPERSION_50

Este atributo é calculado através do campo CALLED dos N sumários. O campo CALLED é uma lista de pares <CONTACT, NUMBER_OF_EVENTS>, em que CONTACT é um contacto de destino e NUMBER_OF_EVENTS é o número de eventos para esse contacto. Agrega-se as N listas por CONTACT numa só, somando o número de eventos

para esse mesmo CONTACT. Esta lista agregada de pares <CONTACT, NUMBER_OF_EVENTS> é ordenada por ordem decrescente do número de eventos, para que na primeira posição esteja o contacto de destino com maior número de eventos e na última posição esteja o contacto de destino com menor número de eventos. Calcula-se a soma de toda a lista agregada, de forma a obter o total de eventos, TOTAL_NUMBER_OF_EVENTS, para todos os contactos de destino. De seguida percorre-se a lista agregada, que está ordenada por ordem decrescente de número de eventos, somando o número de eventos, até que o total seja superior ou igual a 50% do valor de TOTAL_NUMBER_OF_EVENTS. Quando este valor for atingido os restantes contactos de destino são apagados da lista agregada. O valor de CALL_DISPERSION_50 é o tamanho desta lista agregada.

Este atributo de comportamento pretende representar o número de contactos que dominam significativamente a actividade do subscritor.

CALL_DISPERSION_70

A forma de calcular este atributo é exactamente igual ao anterior, só quem com 70% do número total de eventos em vez dos 50% do método anterior.

Este atributo de comportamento pretende representar o número de contactos que constituem parte significativa da actividade do subscritor.

CALL_DISPERSION_90

A forma de calcular este atributo é exactamente igual ao CALL_DISPERSION_50, só quem com 90% do número total de eventos em vez dos 50% desse mesmo método.

Este atributo de comportamento pretende representar o número de contactos que fazem parte da normal actividade do subscritor. Apenas 10% dos eventos não são considerados de forma a remover contactos efectuados com menor significância.

IMEI_STUFFING

Este atributo representa o número de aparelhos utilizados pelo subscritor. O seu valor é o tamanho da lista contida no atributo de identidade IMEI_LIST.

CELL_DISPERSION_50

Para calcular este atributo são reunidas as listas CELL_USAGE dos N sumários. Relembrando, CELL_USAGE é uma lista de pares <CELL_ID, NUMBER_OF_EVENTS>, em que CELL_ID é o identificador da célula de rede e NUMBER_OF_EVENTS é o número de eventos ocorridos nessa célula. Agrega-se as N listas por CELL_ID numa só, somando o número de eventos para esse mesmo CELL_ID. Esta lista agregada de pares <CELL_ID, NUMBER_OF_EVENTS> é ordenada por ordem decrescente do número de eventos, para que na primeira posição esteja a célula de rede com maior número de eventos e na última posição esteja a célula de rede com menor número de eventos. Calcula-se a soma de toda a lista agregada, de forma a obter o total de eventos, TOTAL_NUMBER_OF_EVENTS, para todas as células de rede. De seguida percorre-se a lista agregada, que está ordenada por ordem decrescente de número de eventos, somando o número de eventos, até que o total seja superior ou igual a 50% do valor de TOTAL_NUMBER_OF_EVENTS. Quando este valor for atingido as restantes células de rede são apagadas da lista agregada. O tamanho da lista agregada é o CELL_DISPERSION_50.

Este atributo de comportamento pretende representar o número de células usadas significativamente na actividade do subscritor.

CELL_DISPERSION_70

A forma de calcular este atributo é exactamente igual ao anterior, só quem com 70% do número total de eventos em vez dos 50% do método anterior.

Este atributo de comportamento pretende representar o número de células que constituem parte significativa da actividade do subscritor.

CELL_DISPERSION_90

A forma de calcular este atributo é exactamente igual ao CELL_DISPERSION_50, só quem com 90% do número total de eventos em vez dos 50% desse mesmo método.

Este atributo de comportamento pretende representar o número de células que fazem parte da normal actividade do subscritor. Apenas 10% dos eventos não são considerados de forma a remover células usadas com menor significância.

IN_OUT_RATIO

O cálculo deste atributo é baseado nas listas CALLED e RECEIVED do N sumários. É somado o total de número de eventos em todas as listas RECEIVED para uma variável TOTAL_IN. É somado o total de número de eventos em todas as listas CALLED para uma variável TOTAL_OUT. O valor de IN_OUT_RATIO é (TOTAL_OUT / TOTAL_IN) * 100.

Este atributo de comportamento representa o rácio entre a actividade de entrada e de saída de um subscritor. Este valor é afectado pelo facto de a lista de RECEIVED não conter eventos com origem noutros operadores (ver *Tabela 3 – Cenários de sumarização*).

SERVICE_DISPERSION_50

Para calcular este atributo são reunidas as listas SERVICES_USAGE dos N sumários. Relembrando, SERVICES_USAGE é uma lista de pares <SERVICE_ID, NUMBER_OF_EVENTS>, em que SERVICE_ID é o identificador serviço e NUMBER_OF_EVENTS é o número de eventos que usaram esse serviço. Agrega-se as N listas por SERVICE_ID numa só, somando o número de eventos para esse mesmo SERVICE_ID. Esta lista agregada de pares <SERVICE_ID, NUMBER_OF_EVENTS> é ordenada por ordem decrescente do número de eventos, para que na primeira posição esteja o serviço com maior número de eventos e na última posição esteja o serviço com menor número de eventos. Calcula-se a soma do número de eventos de toda a lista agregada, de forma a obter o total de eventos, TOTAL_NUMBER_OF_EVENTS, para todos os serviços. De seguida percorre-se a lista agregada, que está ordenada por ordem decrescente de número de eventos, somando o número de eventos, até que o total seja superior ou igual a 50% do valor de TOTAL_NUMBER_OF_EVENTS. Quando este valor for atingido os restantes serviço são apagados da lista agregada. O tamanho da lista agregada é o SERVICE_DISPERSION_50.

Este atributo de comportamento pretende representar o número de serviços usados significativamente na actividade do subscritor.

SERVICE_DISPERSION_70

A forma de calcular este atributo é exactamente igual ao anterior, só quem com 70% do número total de eventos em vez dos 50% do método anterior.

Este atributo de comportamento pretende representar o número de serviços que constituem parte significativa da actividade do subscritor.

SERVICE_DISPERSION_90

A forma de calcular este atributo é exactamente igual ao SERVICE_DISPERSION_50, só quem com 90% do número total de eventos em vez dos 50% desse mesmo método.

Este atributo de comportamento pretende representar o número de serviços que fazem parte da normal actividade do subscritor. Apenas 10% dos eventos não são considerados de forma a remover serviços usados com menor significância.

No final do cálculo de todos estes atributos está construído o perfil do subscritor. Nesta fase é necessário armazenar o perfil num repositório de perfis.

Resumindo, o processo de construção de um perfil é feito em 3 passos:

1. Obter os sumários disponíveis;
2. Cálculo do perfil (cálculo dos atributos de identidade e de comportamento);
3. Devolução do perfil ao agente de detecção e/ou de extracção de conhecimento.

5. Agente de detecção

O agente de detecção é responsável por detectar subscritores que reentram na rede do operador, subscritores esses que foram previamente detectados como fraudulentos e consequentemente banidos da rede do operador. Para atingir o seu objectivo usa os atributos de identidade calculados pelo agente de *profiling* para esses mesmos subscritores. Como resultado, este agente devolve ao Fraud:RAID a indicação desses mesmos suspeitos.

Neste capítulo é detalhado o agente de detecção, explicando o seu funcionamento e processos, quais as técnicas de detecção envolvidas e qual o resultado devolvido à aplicação de FMS da WeDo.

5.1. Introdução

O agente de detecção recebe como *input* do Fraud:RAID a indicação de subscritores detectados como fraudulentos e banidos da rede do operador (ver *Figura 3 – Interacção SMA - Fraud:RAID, passos 1 e 2*). Tem como objectivo a detecção de reentrada desses subscritores na rede do operador. Fornece como *output* para o Fraud:RAID suspeitos de serem subscritores previamente banidos a tentarem reentrar na rede.

Dentro do SMA existe interacção entre o agente de detecção e o agente de *profiling* (ver Figura 4 – Interacção SMA - Fraud:RAID, passo 3), não só para obter informação acerca do subscritor fraudulento, mas também para obter informação acerca de um possível subscritor suspeito de ser um fraudulento a reentrar na rede do operador.

5.2. Processos

O agente de detecção é composto por dois processos: criação de um caso e detecção. O primeiro processo, criação de um caso, executa quando o agente recebe indicação de que um subscritor foi detectado como fraudulento e desencadeia o processo de construção de perfil do agente de *profiling*. O segundo processo, de detecção, ao contrário do primeiro processo deste agente e dos dois processos do agente de *profiling* (sumarização e construção de perfil), não é reactivo, ou seja, não precisa de nenhum *input* para executar. Este processo executa periodicamente sobre os casos criados pelo primeiro processo deste agente, usa os sumários criados pelo processo de sumarização do agente de *profiling* para detectar fraudulentos a reentrar na rede do operador e pode desencadear o processo de construção de perfil do agente de *profiling* caso precise de obter informações acerca de um subscritor.

5.2.1. Processo de criação de um caso

O processo de criação de um caso consiste, como o próprio nome indica, em criar um caso. Um caso é uma referência para um subscritor que foi detectado como fraudulento pelo Fraud:RAID e banido da rede, que o agente de detecção deve detectar caso este tente reentrar na rede do operador.

A Figura 7 representa a estrutura do agente de detecção, os seus processos e a interacção com o agente de *profiling* durante o processo de criação de um caso.

Este processo executa quando recebe indicação do Fraud:RAID de um subscritor detectado como fraudulento (passo 1 na imagem). O passo seguinte é desencadear o processo de construção de perfil do agente de *profiling* para esse subscritor (passo 2 na imagem). Este processo retorna um perfil, cujos atributos de identidade (SOCIAL_NET_IN, SOCIAL_NET_OUT, CELL_LIST, IMEI_LIST) são utilizados para construir um caso. Este caso é armazenado num repositório de casos (passo 3 na imagem).

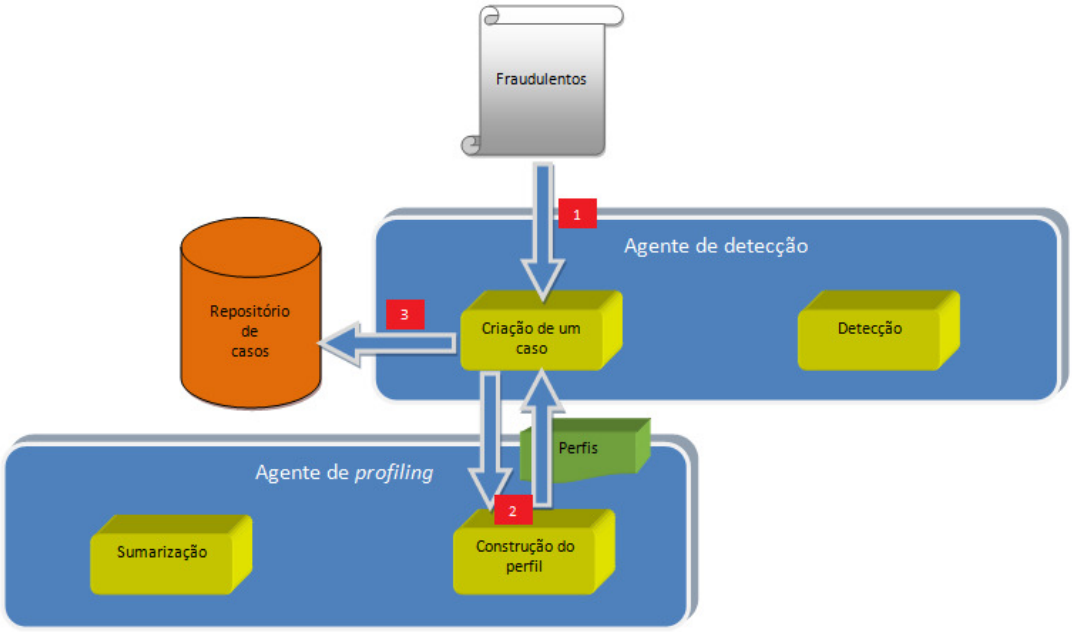


Figura 7 – Processo de criação de um caso

A estrutura de um caso é um subconjunto de atributos do perfil do subscritor fraudulento, mais concretamente, é constituído pelos atributos de identidade. A estrutura é apresentada na Tabela 5.

Atributo	Descrição
SUBSCRIBER_ID	Identifica o subscritor a que pertence caso. É um identificador único do subscritor.
CREATED_DATE	Data de criação do caso.
SOCIAL_NET_IN	Lista de números que mais frequentemente contactam o subscritor.
SOCIAL_NET_OUT	Lista de números que o subscritor mais frequentemente contacta.
CELL_LIST	Lista de células de rede mais utilizadas pelo subscritor.
IMEI_LIST	Lista de aparelhos usados pelo subscritor.

Tabela 5 – Estrutura de um caso

Os casos que existirem criados no repositório de casos serão os casos que o processo de detecção irá analisar quando executar periodicamente. Estes casos representam subscritores fraudulentos banidos da rede do operador que é necessário detectar que reentraram na rede.

5.2.2. Processo de detecção

Dando uma visão geral acerca deste processo, a Figura 8 representa o processo de detecção, a sua interacção com o agente de *profiling* e o seu *output*.

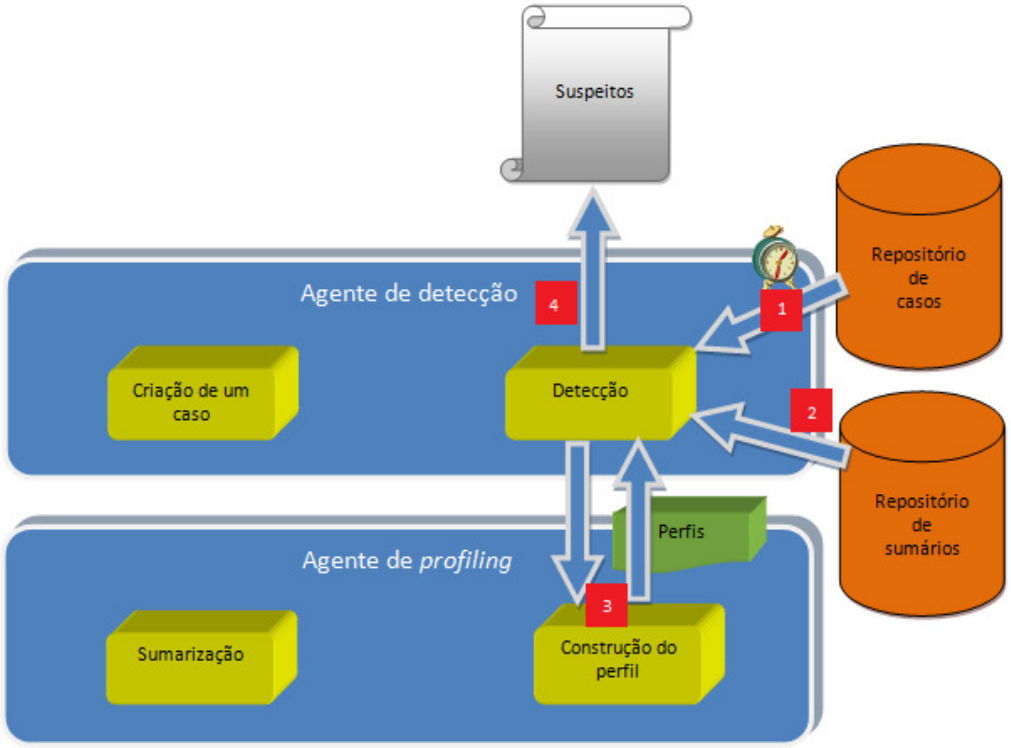


Figura 8 – Processo de detecção

O processo de detecção executa periodicamente, passo 1 na imagem, para os casos que estão no repositório de casos. Para cada um destes casos o processo de detecção obtém informação do repositório de sumários, passo 2 na imagem, para tentar detectar se o subscritor fraudulento tentou reentrar na rede do operador. Caso seja necessária mais informação, pode pedir ao processo de construção de perfil para construir um perfil para um dado suspeito, passo 3 na imagem. Caso exista a suspeita da reentrada de um subscritor fraudulento previamente banido o Fraud:RAID deve ser notificado acerca do subscritor em questão.

Este agente de detecção detecta subscritores a reentrar na rede do operador através de dois métodos: por associação de actividade social ou por associação de IMEI. Cada um dos métodos será explicado separadamente. Contudo, o primeiro passo é comum

para os dois métodos de detecção: quando o processo de detecção executa o primeiro passo é obter uma lista de casos a partir do repositório de casos. Para cada um dos casos o processo pega nos seus atributos (SOCIAL_NET_IN, SOCIAL_NET_OUT, CELL_LIST, IMEI_LIST) e executa cada um dos métodos de detecção.

5.2.2.1. Detecção por associação por actividade social

O objectivo deste método é descobrir se o subscritor fraudulento reentra na rede monitorizando os contactos com quem ele tinha mais frequentemente actividade antes de ser banido. Para isso, este método usa os atributos do caso CREATED_DATE, SOCIAL_NET_IN e SOCIAL_NET_OUT. Relembrando, CREATED_DATE é a data de criação do caso (idealmente é a data em que o subscritor foi detectado como fraudulento e banido da rede), SOCIAL_NET_IN é a lista de números que mais frequentemente contactam o subscritor e SOCIAL_NET_OUT é a lista de números que o subscritor mais frequentemente contacta.

A Figura 9 representa o significado destas listas, SOCIAL_NET_IN e SOCIAL_NET_OUT.

Do lado esquerdo, a vermelho, estão o N contactos presentes na lista **S**OCIAL_**N**ET_**I**N, de SNI 1 a SNI N, que mais frequentemente contactaram o subscritor fraudulento. Do lado direito, a verde, estão os N contactos presentes na lista **S**OCIAL_**N**ET_**O**UT, de SNO 1 a SNO N, que foram mais frequentemente contactados pelo subscritor fraudulento.

O primeiro passo é obter os sumários com data superior à data CREATED_DATE para todos contactos presentes nas listas SOCIAL_NET_IN e SOCIAL_NET_OUT. Cada contacto presente em qualquer das listas terá no repositório N sumários.

Para os sumários de um contacto presente na lista **SOCIAL_NET_IN** interessa o campo **CALLED** (eventos gerados pelo subscritor, é uma lista de pares <CONTACT, NUMBER_OF_EVENTS>) dos N sumários. Estas N listas CALLED são agregadas numa só lista de subscritores, onde são filtrados os subscritores que não pertencem à rede do

operador. Deste modo, esta lista contém todos os subscritores que foram contactados pelo subscritor.

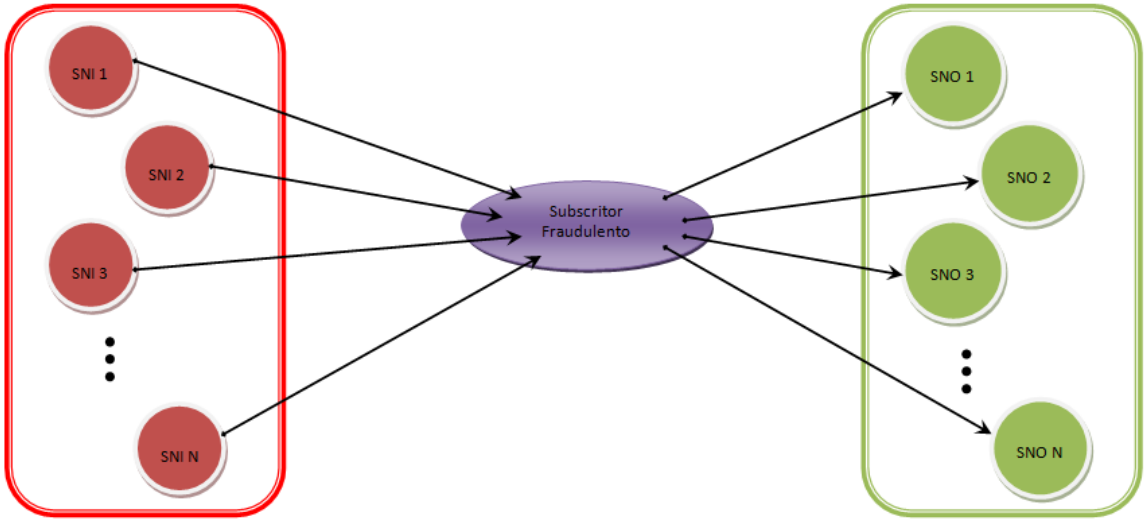


Figura 9 – Representação dos atributos SOCIAL_NET_IN e SOCIAL_NET_OUT

Para os sumários de um contacto presente na lista **SOCIAL_NET_OUT** interessa o campo **RECEIVED** (eventos recebidos pelo subscritor, é uma lista de pares <CONTACT, NUMBER_OF_EVENTS>) dos N sumários. Estas N listas RECEIVED são agregadas numa só lista de subscritores. Deste modo, esta lista contém todos os subscritores que este subscritor contactou. A Figura 10 apresenta o estado actual do método de detecção.

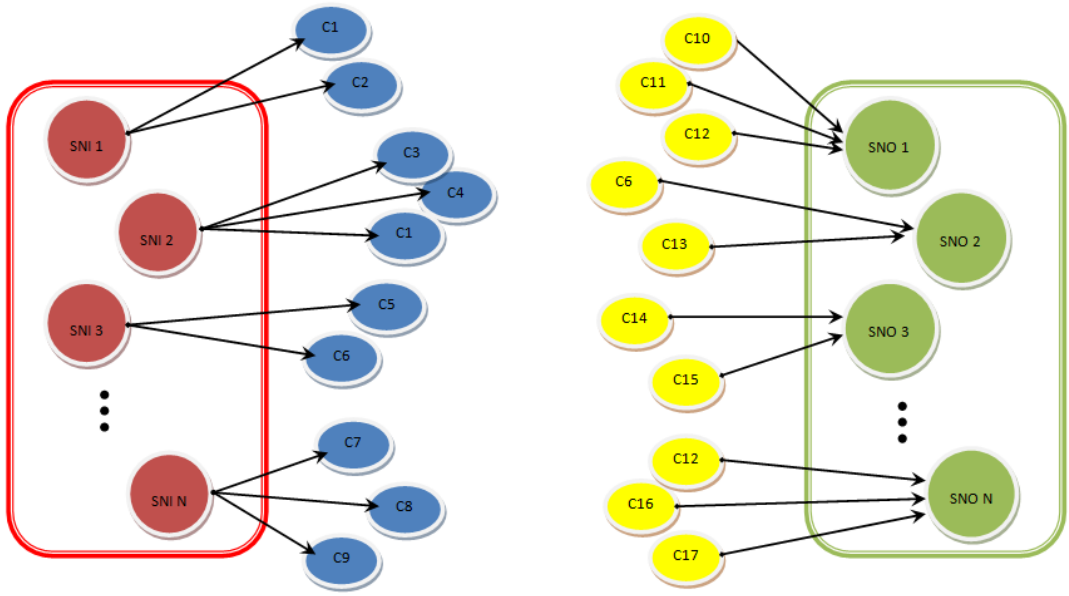


Figura 10 – Método de detecção por associação por actividade social

Nesta fase o método tem conhecimento de:

- Todos os subscritores contactados para cada contacto na lista SOCIAL_NET_IN, representados a azul na figura;
- Todos os subscritores que contactaram cada contacto na lista SOCIAL_NET_OUT, representados a amarelo na figura.

O próximo passo é determinar quais destes subscritores serão investigados. Escolhem-se para ser investigados todos os subscritores que:

1. Estejam nas listas CALLED de mais que um dos subscritores da lista SOCIAL_NET_IN;
2. Estejam nas listas RECEIVED de mais que um dos subscritores da lista SOCIAL_NET_OUT;
3. Estejam nas listas CALLED de um subscritor da lista SOCIAL_NET_IN e nas listas RECEIVED de um subscritor da lista SOCIAL_NET_OUT.

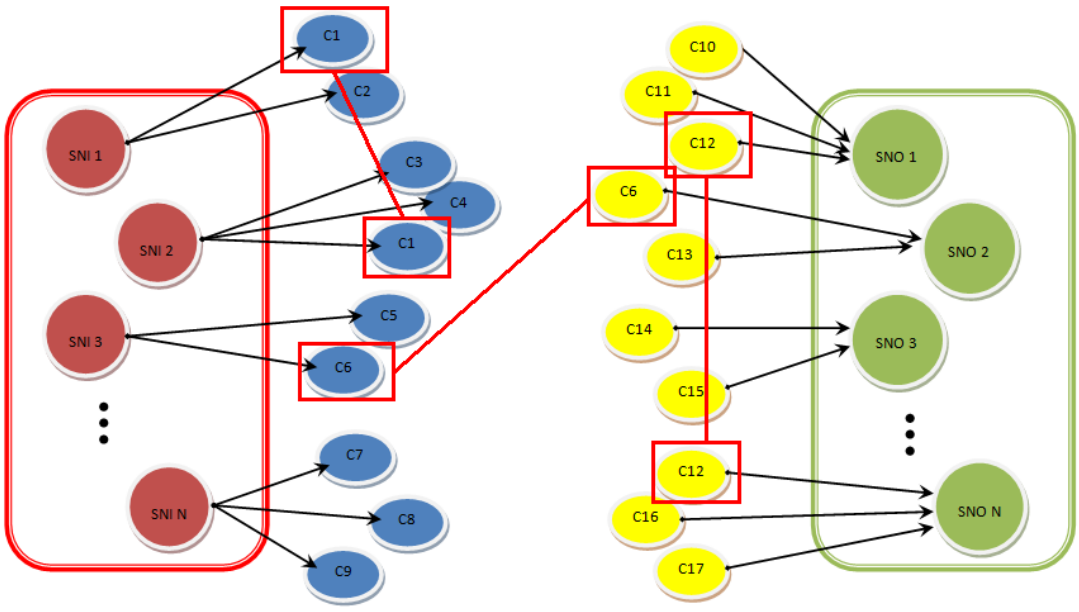


Figura 11 – Candidatos a investigação

Analisando a Figura 11, existem 3 candidatos a serem investigados:

- C1, porque está nas listas CALLED de SNI 1 e nas listas CALLED de SNI 2;
- C12, porque está nas listas RECEIVED de SNO 1 e nas listas RECEIVED de SNO N;
- C6, porque está nas listas CALLED de SNI 3 e nas listas RECEIVED de SNO 2.

Após identificados os candidatos a serem investigados o processo irá para cada um dos candidatos:

1. Obter os sumários para esse candidato;
2. Pedir o perfil deste candidato ao processo de construção do agente de *profiling*.

Com os sumários, o processo usa as listas CALLED e RECEIVED e calcula três valores:

- **TOTAL_SNI_PERCENTAGE** – percentagem dos contactos presentes na lista SOCIAL_NET_IN que estão presentes nas listas RECEIVED;
- **TOTAL_SNO_PERCENTAGE** – percentagem dos contactos presentes na lista SOCIAL_NET_OUT que estão presentes na lista CALLED;
- **TOTAL_SN_PERCENTAGE** – percentagem dos contactos presentes na lista SOCIAL_NET_IN e SOCIAL_NET_OUT que estão presentes nas listas RECEIVED e RECEIVED respectivamente.

Com o perfil do candidato o processo compara as listas SOCIAL_NET_IN, SOCIAL_NET_OUT e CELL_LIST do perfil do candidato com as listas SOCIAL_NET_IN, SOCIAL_NET_OUT e CELL_LIST do perfil do subscritor fraudulento. Destas comparações o processo calcula quatro novos valores:

- **SNI_PERCENTAGE** – percentagem de contactos da lista SOCIAL_NET_IN do perfil do subscritor fraudulento que estão presentes na lista SOCIAL_NET_IN do perfil do candidato;
- **SNO_PERCENTAGE** – percentagem de contactos da lista SOCIAL_NET_OUT do perfil do subscritor fraudulento que estão presentes na lista SOCIAL_NET_OUT do perfil do candidato;
- **SN_PERCENTAGE** – percentagem de contactos das listas SOCIAL_NET_IN e SOCIAL_NET_OUT do perfil do subscritor fraudulento que estão respectivamente presentes nas listas SOCIAL_NET_IN e SOCIAL_NET_OUT do perfil do candidato;
- **CELL_PERCENTAGE** – percentagem de células de rede da lista CELL_LIST do subscritor fraudulento que estão presentes na lista CELL_LIST do candidato.

O valor de CELL_PERCENTAGE apesar de não ser um atributo que permita identificar se realmente se trata do mesmo subscritor é um indicador que pode ajudar bastante nessa decisão, pois retrata a área de acção do subscritor.

Os valores calculados usando os sumários (TOTAL_SNI_PERCENTAGE, TOTAL_SNO_PERCENTAGE e TOTAL_SN_PERCENTAGE) representam a quantidade de contactos em comum entre o candidato suspeito e o subscritor fraudulento considerando toda a actividade disponível.

Os valores calculados usando os perfis (SNI_PERCENTAGE, SNO_PERCENTAGE, SN_PERCENTAGE) representam a quantidade de contactos em comum entre o candidato suspeito e o subscritor fraudulento considerando apenas os contactos mais significativos do candidato suspeito.

5.2.2.2. Detecção por associação por IMEI

O método anterior detecta se um subscritor fraudulento reentra da rede do operador através da monitorização da actividade da sua rede social após o momento em que o subscritor foi detectado e banido. Este método usa todos os dados disponíveis (incluindo sumários de datas anteriores à data de detecção do subscritor fraudulento) e baseia-se apenas nas "armas do crime": os IMEIs. Relembrando, IMEI significa *International Mobile Equipment Identity* (Identificação Internacional de Equipamento Móvel).

O primeiro passo do método de detecção por IMEI é a partir do caso do subscritor fraudulento obter a lista de IMEIs, IMEI_LIST, e nas suas listas SOCIAL_NET_IN e SOCIAL_NET_OUT. De seguida, obtém do repositório de sumários os sumários para cada um dos elementos das listas SOCIAL_NET_IN e SOCIAL_NET_OUT. Desses sumários obtém as listas IMEI_LIST, CALLED e RECEIVED. Para cada um dos elementos presentes nas listas CALLED e RECEIVED, vai novamente ao repositório obter os sumários para esse elemento, de onde obtém as listas IMEI_LIST. A Figura 12 pretende traduzir o estado actual do método de detecção por IMEI.

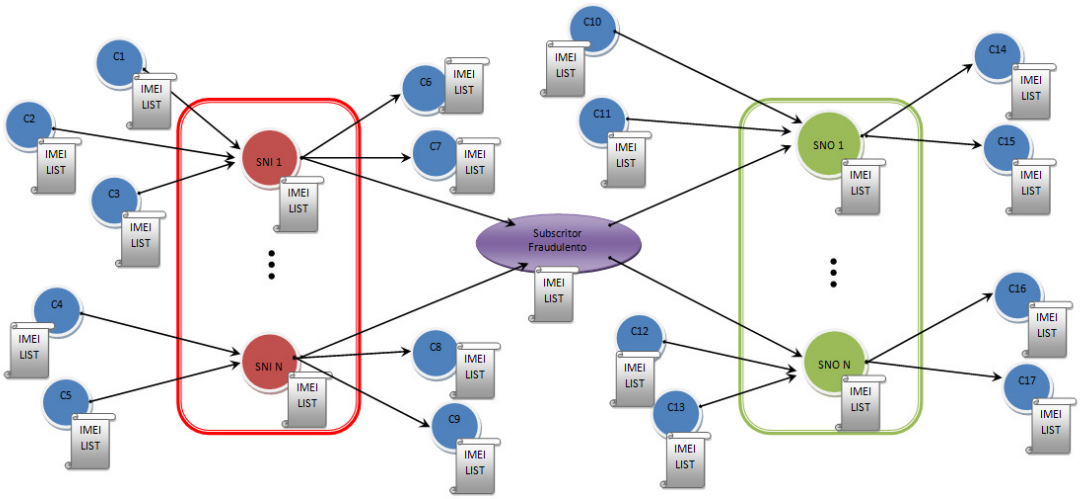


Figura 12 – Estado do método de detecção por IMEI

O passo seguinte é percorrer cada um destes elementos:

- Os elementos da lista SOCIAL_NET_IN, de SNI 1 a SNI N, representados a vermelho na figura;
- Os elementos da lista SOCIAL_NET_OUT, de SNO 1 a SNO N, representados a verde na figura;
- Os elementos que tiveram actividade com todos os elementos da rede social do subscritor fraudulento, de C1 a C17, representados a azul na figura;

e para cada destes elementos verificar se tem na sua IMEI_LIST algum dos IMEIs presentes na IMEI_LIST do subscritor fraudulento. Caso exista um ou mais IMEIs em comum, o subscritor é considerado um suspeito, ao qual se associam os IMEIs em questão.

A maior parte das soluções de fraude analisadas anteriormente já oferece métodos para detecção por IMEI para impedir futuros acessos desse IMEI, mas a diferença é que esses métodos usam o IMEI que foi usado na altura da detecção do fraudulento. A vantagem deste método é que usa todos os IMEIs que foram utilizados pelo subscritor fraudulento anteriormente ao momento da detecção, bem como IMEIs utilizados anteriormente por outros subscritores, permitindo assim uma maior eficiência deste tipo de detecção por IMEI.

Este método pode tornar-se ainda mais eficiente quando o subscritor detectado como suspeito for considerado como fraudulento pelo FMS Fraud:RAID, tornando este processe iterativo, pois usará os IMEIs do suspeito para detectar mais casos suspeitos.

5.2.3. Tratamento dos suspeitos

O tratamento dos suspeitos detectados pelo processo de detecção fica sob a responsabilidade do FMS Fraud:RAID. O método de detecção passa os suspeitos para o Fraud:RAID, indicando para cada suspeito a causa que levou à detecção, associação por actividade social ou associação por IMEI, com os devidos valores que causaram a detecção, no caso da associação por actividade social deve devolver os valores de TOTAL_SNI_PERCENTAGE, TOTAL_SNO_PERCENTAGE, TOTAL_SN_PERCENTAGE, SNI_PERCENTAGE, SNO_PERCENTAGE, SN_PERCENTAGE e CELL_PERCENTAGE; no caso da associação por IMEI deve devolver os valores dos IMEIs que possibilitaram a associação.

Assim o FMS Fraud:RAID tem a flexibilidade para:

- Gerar automaticamente um caso de fraude ou fazer os valores passar um conjunto de regras para determinar se os valores são suficientes para ser considerado um fraudulento a reentrar na rede (por exemplo: apenas considerar suspeitos cujo valor de TOTAL_SNO_PERCENTAGE seja maior que 50% e cujo valor de CELL_PERCENTAGE seja superior a 30%);
- Tomar as acções mais apropriadas (bloquear o suspeito, adicionar o suspeito a uma lista negra, mover o suspeito para um grupo de risco).

6. Agente de extracção de conhecimento

O objectivo do agente de extracção de conhecimento é extrair conhecimento a partir do comportamento dos utilizadores fraudulentos. Para atingir o seu objectivo, este agente usa os atributos de comportamento calculados pelo agente de *profiling* para esses mesmos subscritores fraudulentos. O resultado deste agente é conhecimento sob a forma de um relatório de regras de associação, para que os analistas de fraude possam melhorar o seu *know-how* acerca do comportamento dos fraudulentos na rede, potenciando os recursos (informáticos, humanos, definição de novas abordagens) para detecção e prevenção de fraude.

Ao longo deste capítulo é detalhado o agente de extracção de conhecimento, começando com uma pequena introdução à tecnologia que o suporta – a extracção de conhecimento, seguindo-se a motivação que levaram à inclusão deste agente no SMA, quais os seus objectivos, finalizando com uma secção dedicada a explicar o seu funcionamento.

6.1. Descoberta de Conhecimento em Base de Dados

O termo DCBD (Descoberta de Conhecimento em Base de Dados) (51), ou em inglês KDD (Knowledge Discovery in Database), foi formalizado em 1989 como uma referência ao conceito mais amplo de procura de conhecimento em dados e é um processo que envolve a identificação e o reconhecimento de padrões numa Base de Dados de uma forma automática.

Descobrir conhecimento significa extrair, de grandes conjuntos de dados, sem nenhuma formulação prévia de hipóteses, informações genéricas, relevantes e previamente desconhecidas, que podem ser utilizadas para a tomada de decisões. A principal característica é a extracção não trivial de informações a partir de um conjunto

de dados de grande porte. Essas informações são necessariamente implícitas, previamente desconhecidas, válidas e potencialmente úteis (51).

Entre as principais motivações que levaram ao desenvolvimento e crescimento desta área encontram-se as seguintes:

- Tecnologias de Bases de Dados solidificadas;
- Ferramentas automáticas de procura e arquivo de informação;
- O armazenamento digital de informação promove um aumento muito significativo na quantidade de dados disponíveis;
- O custo do arquivo de informação diminui drasticamente.

O processo de DCBD inicia-se com a compreensão do estudo do domínio da aplicação e percepção dos objectivos finais a serem atingidos. De seguida o processo de DCBD é constituído por um conjunto de etapas, representadas na Figura 13.

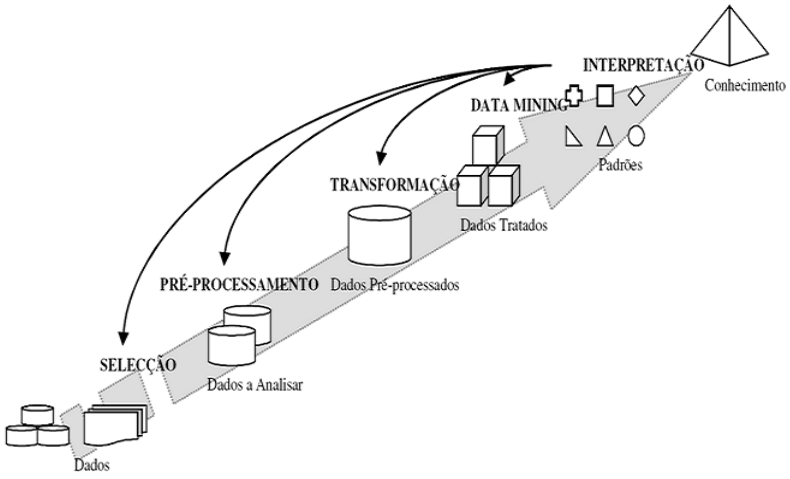


Figura 13 – Processo de DCBD, extraído de "Data Mining - Descoberta de conhecimento em bases de dados" (51)

Uma vez definido o domínio sobre o qual se pretende executar o processo de DCBD, é feita a **selecção** e recolha de um conjunto de dados ou variáveis necessárias. Pretende-se seleccionar ou segmentar os dados de acordo com alguns critérios, determinando os subconjuntos.

A etapa de **pré-processamento** comporta a limpeza dos dados, onde são tratados dados inconsistentes, estabelecidas estratégias de para resolver omissão de dados, eliminação de ruídos e erros, determinação de informação relevante.

Os dados pré-processados devem ainda passar por uma fase de **transformação**, que os armazena no formato adequado, facilitando o uso das técnicas de Data Mining da fase seguinte. Os dados são classificados por assunto, cronologicamente, agregando-se ou generalizando-se e constituem um repositório de dados organizado e de grande dimensão (Data Warehouses).

Prosseguindo no processo, chega-se à fase de DM (**Data Mining**) propriamente dita, que começa com a escolha dos métodos e das técnicas a serem aplicados. Essa escolha depende fundamentalmente do objectivo do processo. Geralmente, o processo de pesquisa é interactivo, permitindo que os analistas revejam o resultado, formem um novo conjunto de questões para refinar a selecção e re-alimentem o sistema com novos parâmetros.

No final do processo, o sistema de DM gera um relatório de descobertas que passam, então, por uma etapa de **interpretação**, permitindo obter o conhecimento, o qual pode ser usado para o suporte à tomada de decisão. A informação encontrada deve estar numa forma perceptível ao utilizador do sistema e deve-se fazer a verificação da qualidade dessa mesma informação.

Esta representação das fases do processo de DCBD pode sugerir que se trata de um percurso linear, no entanto, isso geralmente não se verifica, uma vez que em cada etapa pode ser identificada a necessidade de voltar para qualquer uma das fases anteriores.

Conforme se referiu na explicação do processo de DCBD, na etapa de DM deve-se escolher os métodos e as técnicas a serem aplicados consoante o objectivo do processo. A Tabela 6 apresenta os objectivos para os modelos de DM.

Após a escolha do modelo de DM a ser implementado, é necessário escolher a técnica que será aplicada. Estas técnicas podem ser usadas isoladamente ou podem ser combinadas. A maioria das técnicas de DM é baseada em conceitos de AA (Aprendizagem Automática), reconhecimento de padrões, estatística, IA (Inteligência Artificial) e modelos gráficos.

Modelo de DM	Objectivo
Classificação	Encontrar uma função que associa um caso a uma classe dentro de um conjunto de diversas classes discretas; é o objectivo mais comum;
Previsão	Determinação de uma função que realize a previsão de valores futuros ou desconhecidos de outras variáveis de interesse com base em algumas variáveis e na descoberta de padrões a partir de exemplos;
Regressão linear	Determinação de uma função que realize a previsão de uma variável, representando, de uma forma aproximada, um comportamento variável;
Segmentação	Identificação de um conjunto finito de segmentos (categorias/clusters) como forma de descrição dos dados; a similaridade intra-segmento é alta, a similaridade inter-segmentos é baixa;
Associação	Procura de um modelo que descreva dependências significativas entre variáveis; identifica grupos de factos relacionados, que possam estar directa ou indirectamente associados;
Sumarização	Encontrar uma descrição compacta para um subconjunto de dados;
Visualização	Tratar da apresentação dos resultados, permitindo uma análise gráfica dos dados;
Detecção de desvios	Descoberta de alterações significativas nos dados, a partir de valores previamente medidos ou valores normativos.

Tabela 6 – Modelos de Data Mining e objectivos

A técnica a ser aplicada depende do modelo de DM escolhido anteriormente. A Tabela 7 representa as técnicas associadas para alguns dos modelos.

Modelo	Técnicas
Segmentação	Árvores de Decisão
	Redes Neuronais
	Algoritmos Genéticos
	Indução de Regras
	Redes Bayes
Classificação	Árvores de Decisão
	Redes Neuronais
	Indução de Regras
	Conjuntos Difusos
	Conjuntos Aproximados
	Sistemas de Classificação
	Redes de Bayes
	Algoritmos Genéticos
Previsão	Árvores de Decisão
	Redes Neuronais
	Algoritmos Genéticos
	Indução de Regras
Associação	Redes Neuronais
	Indução de Regras
	Redes de Bayes
Sumarização	Redes de Bayes
Visualização	Árvores de Decisão
	Redes de Bayes

Tabela 7 – Modelos e Técnicas de Data Mining

6.2. Motivação e objectivos

A definição de uma nova abordagem, desenhada na secção *2.4*, surgiu pela necessidade da detecção de subscritores fraudulentos por identidade em vez de comportamento. Então porquê incluir uma análise ao comportamento? A motivação para inserir um agente de extracção de conhecimento na solução surgiu quando se

definiu o agente de *profiling*: uma vez que vamos ter um agente a sumarizar informação para os subscritores acerca da sua identidade, porque não aproveitar e guardar informação acerca do seu comportamento de forma a tentar obter conhecimento útil? Além da vantagem óbvia, a extracção de conhecimento, a inclusão deste agente traz uma outra vantagem à solução desenvolvida: enquanto a utilização da detecção por identidade será apenas aplicável em alguns tipos de fraude em que faz sentido tentar detectar reentradas de fraudulentos na rede do operador, a componente da extracção de conhecimento pode ser usada em todos os tipos de fraude, maximizando assim a utilidade do agente de *profiling*.

O objectivo deste agente é a extracção de conhecimento. Esta extracção de conhecimento é efectuada a partir dos atributos de comportamento dos perfis dos subscritores fraudulentos. Analisando os objectivos dos vários modelos de DM na Tabela 6, o modelo de Associação é o que mais se adequa a este objectivo. Especificando mais, o objectivo é utilizar o modelo de DM de Associação, de forma a perceber que variáveis (atributos de comportamento) estão directamente associadas a um grupo (subscritores fraudulentos).

6.3. Processos

O agente de EC é composto por dois processos:

- Armazenamento de conhecimento – este processo obtém o perfil do subscritor através do processo de construção de perfil do agente de *profiling* e armazena os atributos de comportamento num repositório de conhecimento;
- Extracção de conhecimento – este processo com base no conhecimento que obtém do repositório de conhecimento aplica as técnicas necessárias para gerar um relatório com o conhecimento que foi extraído.

Enquanto o primeiro processo executa quando recebe indicação de que um subscritor foi detectado como fraudulento, o segundo processo pode executar logo de seguida ou então pode ser executado a pedido de um analista de fraude.

6.3.1. Processo de armazenamento de conhecimento

O processo de armazenamento de conhecimento está representado na Figura 14. O processo é desencadeado quando recebe a indicação que um conjunto de subscritores foi detectado como fraudulento, representado no passo 1 na figura. De forma que este agente consiga melhores resultados, a cada subscritor identificado como fraudulento deve estar associado o tipo de fraude para o qual foi identificado como fraudulento.

Nesta altura o processo obtém do processo de construção de perfil do agente de *profiling* os perfis para cada um dos subscritores fraudulentos presentes no conjunto inicial que desencadeou o processo, passo 2 na figura.

Por último, no passo 3, o processo retira do perfil os atributos de comportamento (ACTIVITY, CALL_DISPERSION_50 70 e 90, IMEI_STUFFING, CELL_DISPERSION_50 70 e 90, IN_OUT_RATIO, SERVICE_DISPERSION_50 70 e 90) cria um caso de conhecimento de fraude e armazena este caso num repositório de conhecimento. A Tabela 8 representa a estrutura de um caso de conhecimento de fraude.

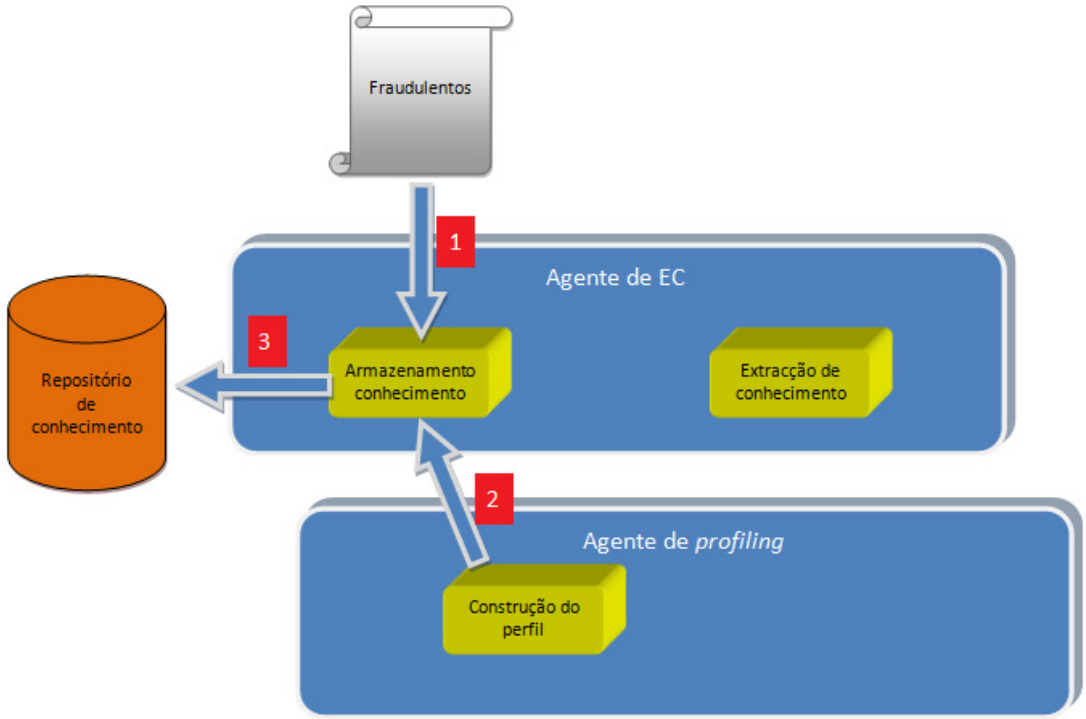


Figura 14 – Processo de armazenamento do agente de EC

Atributo	Descrição
FRAUD_TYPE	Tipo de fraude a que pertence o caso de conhecimento de fraude.
ACTIVITY	Percentagem média de tempo em que o subscritor tem actividade.
CALL_DISPERSION_50	Número mínimo de contactos que são necessários para atingir 50% da actividade de saída do subscritor.
CALL_DISPERSION_70	Número mínimo de contactos que são necessários para atingir 70% da actividade de saída do subscritor.
CALL_DISPERSION_90	Número mínimo de contactos que são necessários para atingir 90% da actividade de saída do subscritor.
IMEI_STUFFING	Número de aparelhos usados pelo subscritor.
CELL_DISPERSION_50	Número mínimo de células de rede que são necessárias para atingir 50% da actividade de saída do subscritor.
CELL_DISPERSION_70	Número mínimo de células de rede que são necessárias para atingir 70% da actividade de saída do subscritor.
CELL_DISPERSION_90	Número mínimo de células de rede que são necessárias para atingir 90% da actividade de saída do subscritor.
IN_OUT_RATIO	Rácio entre a actividade de entrada e de saída do subscritor.
SERVICE_DISPERSION_50	Número mínimo de serviços que são necessários para atingir 50% da actividade de saída do subscritor.
SERVICE_DISPERSION_70	Número mínimo de serviços que são necessários para atingir 70% da actividade de saída do subscritor.
SERVICE_DISPERSION_90	Número mínimo de serviços que são necessários para atingir 90% da actividade de saída do subscritor.

Tabela 8 – Estrutura de um caso de conhecimento de fraude

6.3.2. Processo de extracção de conhecimento

Ao longo do tempo, um analista de fraude desenvolve o seu *know-how* e percebe que existem padrões no comportamento de um subscritor fraudulento que o distinguem.

Alguns dos padrões de comportamento de um subscritor comummente identificados são:

- Dispersão de chamadas – um subscritor realiza chamadas para diversos destinos. Por exemplo, em cada 100 chamadas existe uma diversidade de 80% destinos.
- Rácio de serviços – um subscritor utiliza os serviços de uma forma desproporcional. Por exemplo, o serviço de voz é muito mais utilizado que o serviço de SMS.
- Concentração de eventos numa célula – um subscritor origina muitos eventos numa única célula. Pode indicar que se trata de um equipamento utilizado para cometer fraude.
- Stuffing – um subscritor que efectua trocas frequentes de cartão/equipamento.

Estes padrões devem ser usados como indicadores: o facto de um utilizador evidenciar um destes padrões não é sinónimo de fraude. De forma a detectar situações de fraude estes padrões devem ser correlacionados entre si: um utilizador que evidencia um destes padrões não é sinónimo de fraude, mas se evidenciar dois destes padrões de comportamento então provavelmente trata-se de uma situação de fraude. Esta correlação de padrões permite a identificação de comportamentos fraudulentos. Por exemplo, padrões de elevada dispersão de destinos, elevada concentração de chamadas numa só célula, uso desproporcional de serviços de chamadas internacionais em comparação com as chamadas nacionais e presença de *stuffing* indicam tratar-se de uma fraude por *Bypass*. Padrões de elevada dispersão de destinos, elevada concentração de chamadas numa só célula, uso desproporcional de serviços de chamadas nacionais em comparação com as chamadas internacionais e presença de *stuffing* indicam tratar-se de uma fraude por *Call Sell*.

O objectivo do processo de extracção de conhecimento é, com base nos atributos de comportamento dos perfis dos subscritores fraudulentos, descobrir e quantificar estes indicadores para cada tipo de fraude, de forma a enriquecer o *know-how* dos analistas de fraude. Importa reforçar o facto do processo de EC de ser executado

separadamente por cada tipo de fraude, uma vez que os indicadores variam consoante o tipo de fraude.

O primeiro passo do processo de EC é a **selecção** de dados. Neste passo, o processo selecciona do repositório de conhecimento os casos do tipo de fraude que está a analisar. Além destes dados, o processo precisa de dados de contra-exemplo, ou seja, dados de comportamento de subscritores não fraudulentos. Nesta fase, o processo deve obter do agente de *profiling* perfis de subscritores não fraudulentos. Os subscritores são escolhidos aleatoriamente e em número igual ao número de casos de fraude do tipo que o processo está a analisar.

Uma vez que os dados provêm do agente de *profiling*, o segundo passo do processo de EC, o **pré-processamento**, não necessita de efectuar a limpeza e tratamento dos dados, uma vez que o processo de construção de perfis já efectuou este trabalho.

O terceiro passo do processo de EC é a **transformação** de dados, que implica o armazenamento dos dados, de fraude e de contra-exemplo, no formato adequado, com vista à sua utilização na fase seguinte de DM.

O quarto passo do processo de EC é a **aplicação de métodos de DM**. Nesta fase, aplica-se o método de Associação. Este método tem como objectivo encontrar padrões frequentes, associações, correlações ou estruturas ocasionais em conjuntos de dados e a definição de regras (implicação ou correlação) de relacionamento entre elementos que ocorrem em comum.

A estrutura típica de uma regra de associação é:

Antecedente → Consequente [suporte, confiança]

em que o suporte e a confiança são medidas de interesse definidas no processo.

O suporte é uma medida de utilidade, mede a frequência com que uma regra de associação surge no conjunto de dados em análise. A confiança é uma medida de certeza, corresponde à percentagem de ocorrências do antecedente juntamente com o consequente. Aplicando este modelo ao processo de EC: o antecedente é uma medida de um atributo de comportamento, por exemplo, ACTIVITY>45%; o

consequente é sempre o caso de fraude; o suporte é a percentagens de casos de fraude que estão incluídos na quantificação do antecedente; a confiança é a percentagem de casos que são realmente de fraude que estão incluídos na quantificação do antecedente.

Expondo um exemplo, para melhor se perceber, a Tabela 9 apresenta um conjunto de dados de entrada, em que existem quatro casos de fraude por *call sell*. No primeiro passo do processo de EC foram seleccionados esses quatro casos e os atributos de comportamento dos perfis de quatro subscritores escolhidos aleatoriamente. Apenas os atributos de comportamento ACTIVITY e são IN_OUT_RATIO apresentados, para simplificar a informação apresentada na Tabela 9.

Tipo do caso	ACTIVITY	IN_OUT_RATIO
Fraude por *call sell*	53%	4.5
Fraude por *call sell*	48%	2.4
Fraude por *call sell*	65%	3.7
Fraude por *call sell*	71%	4.1
Não fraude	17%	1.9
Não fraude	37%	2.3
Não fraude	47%	1.5
Não fraude	29%	3.1

Tabela 9 – Exemplo de dados de entrada para o processo de EC

Analisando os dados da tabela vemos que existem 5 casos com ACTIVITY superior a 45%, e que desses 5 casos, 4 são de fraude. Então a regra ACTIVITY>45% → Fraude por *call sell* tem como suporte 62.5% (5/8=0.625, 5 é o número total de casos em que ACTIVITY é superior a 45% e 8 é o número total de casos) e tem uma confiança de 80% (4/5=0.80):

ACTIVITY>45% → Fraude por *call sell* [62.5% , 80%]

Aplicando o mesmo raciocínio à regra IN_OUT_RATIO>2.0, existem 6 casos com um rácio superior a 2.0 (suporte = 6/8 = 75%) e 4 desses 6 casos são casos de fraude (confiança = 4/6 = 66.7%):

IN_OUT_RATIO>2.0 → Fraude por *call sell* [75% , 66.7%]

Neste contexto, a regra perfeita é a regra que tiver uma confiança de 100%, significando que todos os casos que a regra abrange são efectivamente casos de fraude, e um suporte de 50%, significando que todos os casos de fraude que estão dos dados de entrada foram abrangidos pela regra. Neste exemplo, essa regra perfeita existe:

ACTIVITY>45%, IN_OUT_RATIO>2.0 → Fraude por *call sell* [50% , 100%]

A última etapa do processo de EC é a **interpretação**. Nesta fase, o processo gera um relatório, contendo todas as regras de associação, com os respectivos antecedentes, suportes e confianças, de forma que os analistas de fraude possam tirar partido desta informação.

A Figura 15 representa o processo de extracção de conhecimento. O primeiro passo consiste em obter do repositório de conhecimento os atributos de comportamento para um determinado tipo de fraude. No segundo passo é pedido ao agente de *profiling* perfis de utilizadores "*comuns*", não fraudulentos, de forma a obter os seus atributos de comportamento para utilizar como contra-exemplo. No terceiro passo, após a aplicação do processo de extracção de conhecimento é gerado um relatório de extracção de conhecimento, contendo as regras de associação calculadas.

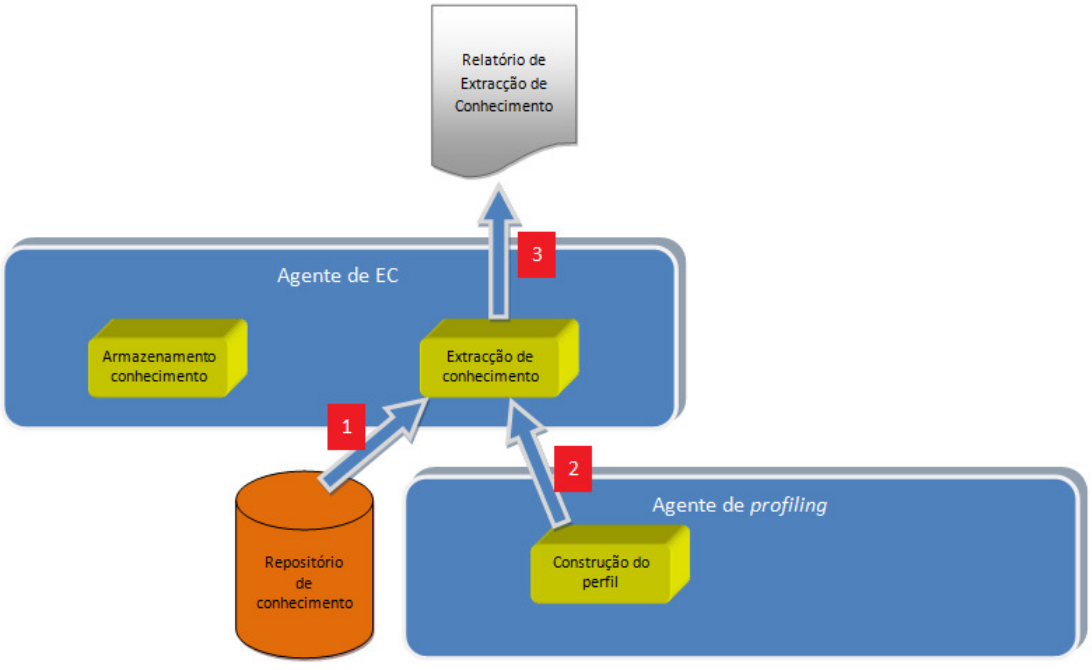


Figura 15 – Processo de extracção de conhecimento do agente de EC

7. Implementação de um protótipo

Este capítulo é dedicado ao trabalho desenvolvido na implementação de um protótipo que implementa a solução especificada ao longo dos capítulos anteriores. O objectivo da implementação de um protótipo é obter indicações acerca da *performance* (qualitativa e quantitativa) da nova solução.

A primeira secção explica como foi desenvolvido o protótipo, na segunda secção são apresentados os resultados de alguns testes efectuados e a terceira secção apresenta uma análise aos resultados obtidos.

7.1. Implementação

O protótipo foi desenvolvido utilizando as tecnologias Java, XML e Oracle. A escolha destas tecnologias foi influenciada pelo facto de estas serem as tecnologias utilizadas no desenvolvimento do FMS da WeDo Techonologies, o Fraud:RAID. A base de dados Oracle foi utilizada para suporte de informação e XML foi utilizado para estruturar a informação. O desenvolvimento dos agentes e dos seus processos foi efectuado em Java, implementando a lógica descrita ao longo dos últimos três capítulos, que não será repetida neste capítulo.

Foi criado um objecto chamado Profile, que representa o perfil que o agente de *profiling* fornece aos agentes de detecção e de EC. Este objecto contém como atributos todos os atributos que compõem o perfil, ver Tabela 4. Foram criadas 5 tabelas na base de dados Oracle:

- FRAUD_T_SUMMARY – representa o repositório de sumários (Figura 6);
- FRAUD_T_FRAUD_CASE – representa o repositório de casos de fraude (Figura 7);

- FRAUD_T_KNOWLEDGE_BASE – representa o repositório de conhecimento (Figura 14);
- FRAUD_T_SUSPECT – tabela onde são guardados os resultados (suspeitos) do processo de detecção do agente de detecção (Figura 8);
- FRAUD_T_ASSOCIATION_RULE – tabela onde são guardados os resultados (regras de associação) do processo de detecção do agente de EC (Figura 15).

A tabela FRAUD_T_SUMMARY tem a estrutura apresentada na Tabela 10. A Figura 16 representa um exemplo dos dados armazenados na tabela FRAUD_T_SUMMARY. O sumário apresentado é para o dia 6 de Julho de 2009, coluna SUMMARY_DATE, o SUBSCRIBER_ID é o meu número de telemóvel e na coluna SUMMARY é possível ver todos os campos que constituem um sumário (ACTIVITY, IMEI_LIST, SERVICES_USAGE, CALLED, RECEIVED, CELL_USAGE) estruturados em XML.

Coluna	Tipo	Descrição
SUMMARY_DATE	DATE	Data do sumário
SUBSCRIBER_ID	VARCHAR2(20 BYTE)	Identificação do subscritor
SUMMARY	CLOB	Dados do sumário

Tabela 10 – Estrutura da tabela FRAUD_T_SUMMARY

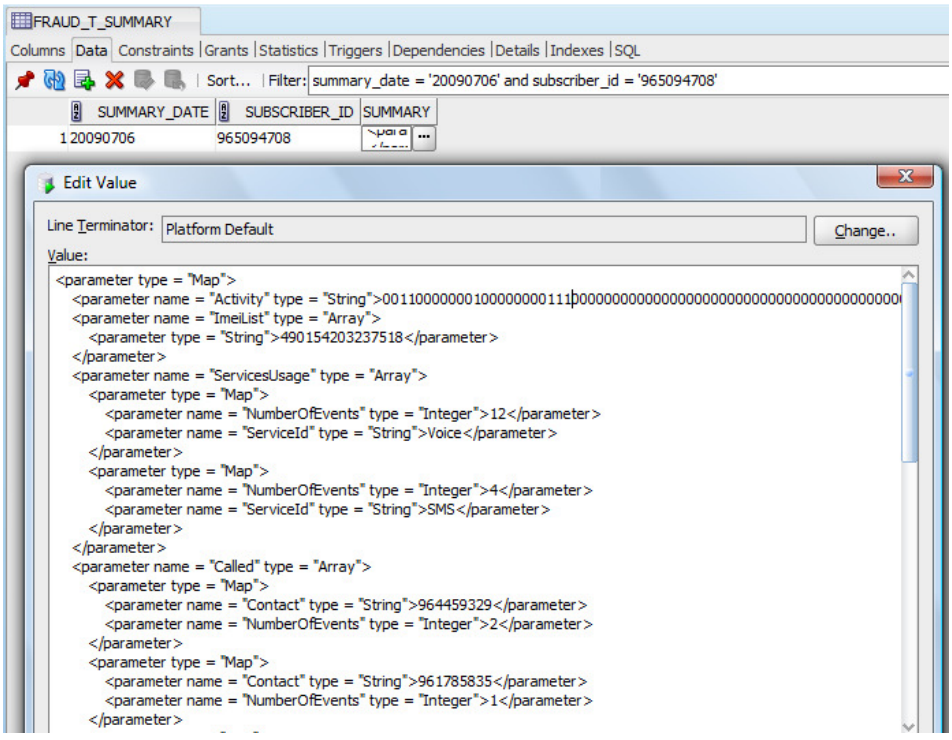


Figura 16 – Exemplo de dados da tabela FRAUD_T_SUMMARY

A tabela FRAUD_T_FRAUD_CASE tem a estrutura apresentada na Tabela 11.

Coluna	Tipo	Descrição
SUBSCRIBER_ID	VARCHAR2(20 BYTE)	Subscritor fraudulento a que pertence o caso de fraude
DETECTION_DATE	DATE	Data em que o subscritor foi detectado como fraudulento
IDENTITY_ATTRIBUTES	CLOB	Atributos de identidade do subscritor fraudulento

Tabela 11 – Estrutura da tabela FRAUD_T_FRUAD_CASE

A Figura 17 representa um exemplo dos dados armazenados na tabela FRAUD_T_FRAUD_CASE. O caso de fraude apresentado foi criado no dia 15 de Julho de 2009, coluna DETECTION_DATE, o SUBSCRIBER_ID é o meu número de telemóvel e na coluna IDENTITY_ATTRIBUTES é possível ver todos os campos do que constituem um caso de fraude (SOCIAL_NET_IN, SOCIAL_NET_OUT, IMEI_LIST, CELL_LIST) estruturados em XML.

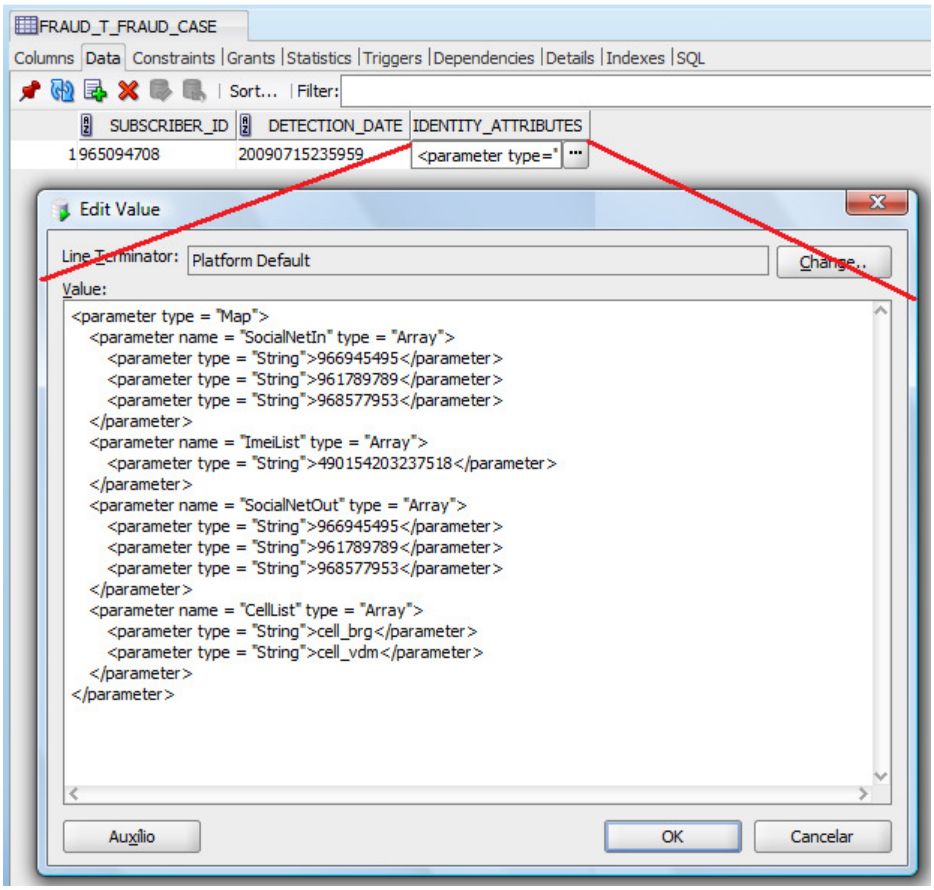


Figura 17 – Exemplo de dados da tabela FRAUD_T_FRAUD_CASE

A tabela FRAUD_T_KNOWLEDGE_BASE tem a estrutura apresentada na Tabela 12.

Coluna	Tipo	Descrição
SUBSCRIBER_ID	VARCHAR2(20 BYTE)	Subscritor (fraudulento neste caso) a que pertence o caso de conhecimento
FRAUD_TYPE	VARCHAR2(20 BYTE)	Tipo de fraude pela qual o subscritor fraudulento foi detectado
BEHAVIOR_ATTRIBUTES	CLOB	Atributos de comportamento do subscritor fraudulento

Tabela 12 – Estrutura da tabela FRAUD_T_KNOWLEDGE_BASE

A Figura 18 representa um exemplo dos dados armazenados na tabela FRAUD_T_KNOWLDGE_BASE. O caso de conhecimento apresentado pertence ao tipo de fraude "Cloning Fraud", coluna FRAUD_TYPE, o SUBSCRIBER_ID é o meu número de telemóvel[10] e na coluna BEHAVIOR_ATTRIBUTES é possível ver todos os campos do que constituem um caso de conhecimento (ACTIVITY, CALL_DISPERSION_50 70 e 90, SERVICE_DISPERSION_50 70 e 90, IMEI_STUFFING, CELL_DISPERSION_50 70 e 90, IN_OUT_RATIO) estruturados em XML.

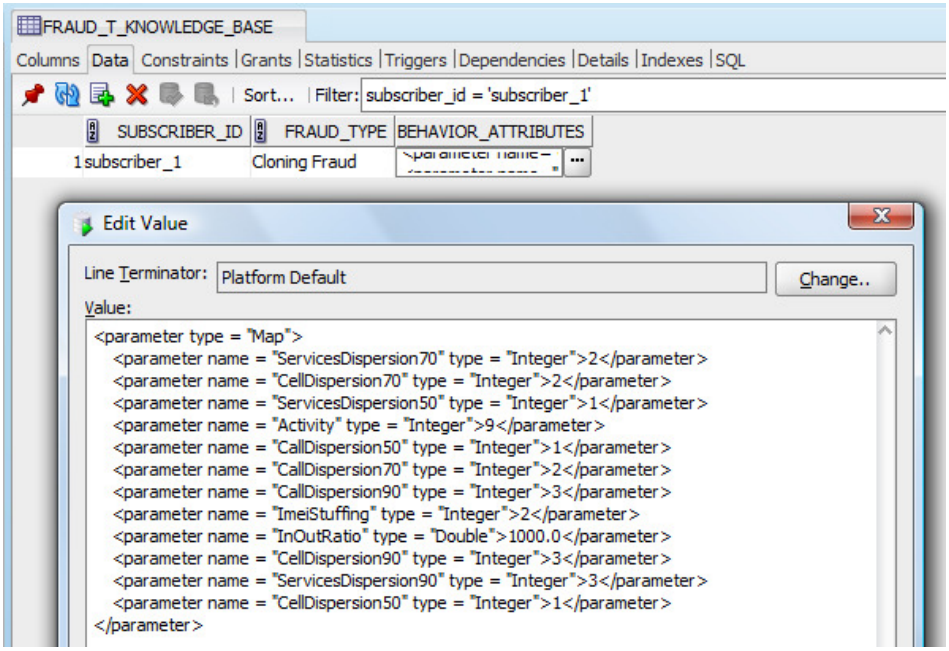


Figura 18 – Exemplo de dados da tabela FRAUD_T_KNOWLEDGE_BASE

[10] Devido à falta de dados reais uso os números de telemóvel como números de subscritores

A tabela FRAUD_T_SUSPECT tem a estrutura apresentada na Tabela 13.

Coluna	Tipo	Descrição
RUN_ID	VARCHAR2(20 BYTE)	Identificador da execução do processo de detecção. Cada execução do processo tem um RUN_ID diferente. É uma sequência
DETECTION_DATE	DATE	Data em que o processo de detecção executou e inseriu estes resultados
FRAUDSTER_ID	VARCHAR2(20 BYTE)	Identificador do fraudulento a que fica associado o novo suspeito
SUSPECT_ID	VARCHAR2(20 BYTE)	Identificação do suspeito
TOTAL_SNI_PERCENTAGE	NUMBER(10,0)	Resultado do processo de detecção, método associação por actividade
TOTAL_SNO_PERCENTAGE	NUMBER(10,0)	Resultado do processo de detecção, método associação por actividade
TOTAL_SN_PERCENTAGE	NUMBER(10,0)	Resultado do processo de detecção, método associação por actividade
SNI_PERCENTAGE	NUMBER(10,0)	Resultado do processo de detecção, método associação por actividade
SNO_PERCENTAGE	NUMBER(10,0)	Resultado do processo de detecção, método associação por actividade
SN_PERCENTAGE	NUMBER(10,0)	Resultado do processo de detecção, método associação

		por actividade
CELL_PERCENTAGE	NUMBER(10,0)	Resultado do processo de detecção, método associação por actividade
IMEI_INTERSECTION	VARCHAR2(2000 BYTE)	Resultado do processo de detecção, método associação por IMEI

Tabela 13 – Estrutura da tabela FRAUD_T_SUSPECT

A Figura 19 representa um exemplo dos dados armazenados na tabela FRAUD_T_SUSPECT. A figura apresenta as colunas divididas em duas partes pois a tabela é muito larga. O processo de detecção executou no dia 16 de Julho de 2009, coluna DETECTION_DATE. É possível ver que o suspeito 965094709 foi identificado por ter em comum com o fraudulento 965094708 vários elementos em comum na sua rede social, 67% dos contactos, (colunas TOTAL_SNI_PERCENTAGE, TOTAL_SNO_PERCENTAGE, TOTAL_SN_PERCENTAGE, SNI_PERCENTAGE, SNO_PERCENTAGE, SN_PERCENTAGE), utiliza metade das células de rede que o fraudulento usava (coluna CELL_PERCENTAGE = 50) e existe um IMEI em comum entre os dois (coluna IMEI_INTERSECTION).

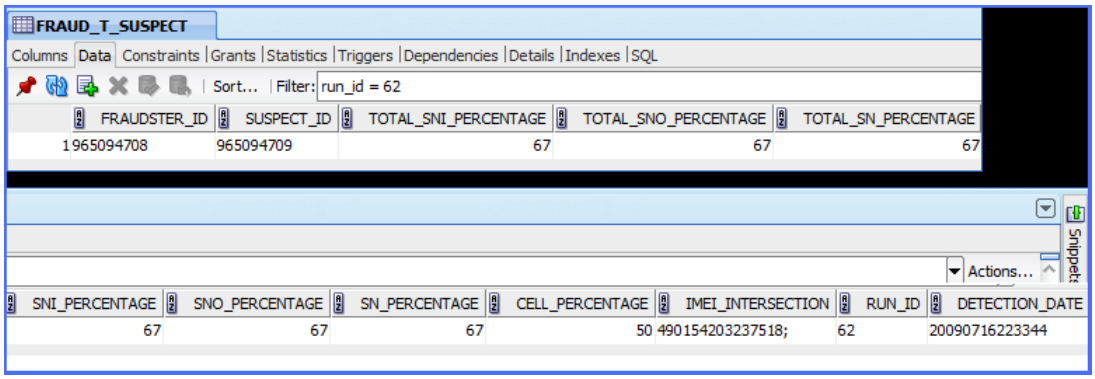


Figura 19 – Exemplo de dados da tabela FRAUD_T_SUSPECT

A tabela FRAUD_T_ASSOCIATION_RULE tem a estrutura apresentada na Tabela 14.

Coluna	Tipo	Descrição
FRAUD_TYPE	VARCHAR2(20 BYTE)	Tipo de fraude para o qual foi executado o processo de extracção de conhecimento
RULE	VARCHAR2(50 BYTE)	Definição da regra de associação
SUPPORT	NUMBER(10,0)	Suporte da regra de associação
TRUST	NUMBER(10,0)	Confiança da regra de associação.

Tabela 14 – Estrutura da tabela FRAUD_T_ASSOCIATION_RULE

A Figura 20 representa um exemplo dos dados armazenados na tabela FRAUD_T_ASSOCIATION_RULE. É possível ver quatro regras calculadas para o tipo de fraude "Cloning Fraud", coluna FRAUD_TYPE. A cada regra, coluna RULE, estão associados valores de suporte e confiança, colunas SUPPORT e TRUST, respectivamente.

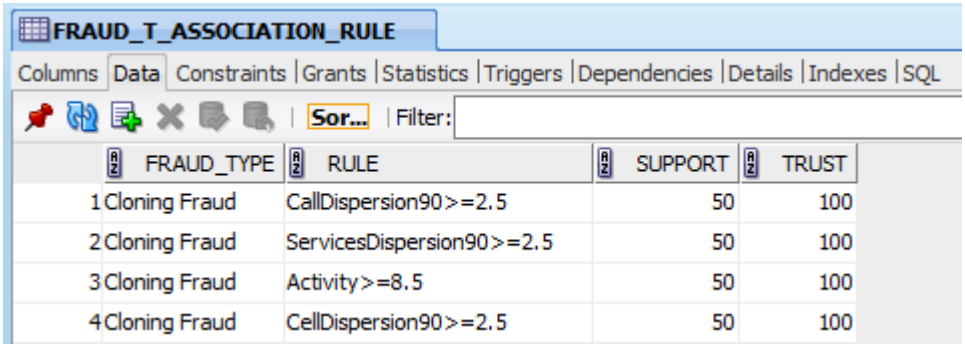


Figura 20 – Exemplo de dados da tabela FRAUD_T_ASSOCIATION_RULE

As tabelas apresentadas nesta secção são o suporte para o armazenamento e troca de informação entre os agentes. Além das tabelas que representam os três repositórios (sumários, casos de fraude e casos de conhecimento) foram definidas mais duas tabelas onde os agentes guardam os seus resultados (suspeitos e regras de associação). De realçar ainda a utilização de XML para estruturação de informação, que permite uma melhor organização dos dados.

7.2. Resultados

Foram efectuados 3 testes. O primeiro teste visa a avaliação da *performance* quantitativa do processo de sumarização. O segundo teste tem como objectivo obter indicações acerca da *performance* qualitativa dos agentes de *profiling* e de detecção. O terceiro teste pretende verificar que o processo de extracção de conhecimento tem o comportamento desejado.

7.2.1. Teste 1 – Sumarização

O primeiro teste efectuado foi um teste de carga, de forma a medir a *performance* quantitativa do processo de sumarização, pois este processo lida com milhões de registos por hora. Estes testes de carga foram efectuados numa máquina com as propriedades apresentadas na Tabela 15.

Propriedade	Valor
Server Model	ia64 hp server BL860c
Operating System	HP-UX B.11.23 U (64 bits)
RAM Memory	12.267 MB
Total SWAP Memory	24.576 MB
Number of CPUs	4
Clock Speed	1.595 MHz
Processor family	32 Intel(R) Itanium 2 9000 series

Tabela 15 – Propriedades da máquina de testes

Foram efectuados 3 testes:

- 1 milhão de subscritores a gerar 5 eventos cada subscritor num total de 5 milhões de eventos;
- 1 milhão de subscritores a gerar 10 eventos cada subscritor num total de 10 milhões de eventos:

- 2 milhões de subscritores a gerar 5 eventos cada subscritor num total de 10 milhões de eventos.

Os resultados obtidos são apresentados na Tabela 16.

Nº subscritores	Nº eventos por subscritor	Nº total de eventos	Tempo (segundos)	Tempo (minutos)
1.000.000	5	5.000.000	1267	21min 07seg
1.000.000	10	10.000.000	1411	23min 17seg
2.000.000	5	10.000.000	1712	28min 32seg

Tabela 16 – Resultados do teste de *performance*

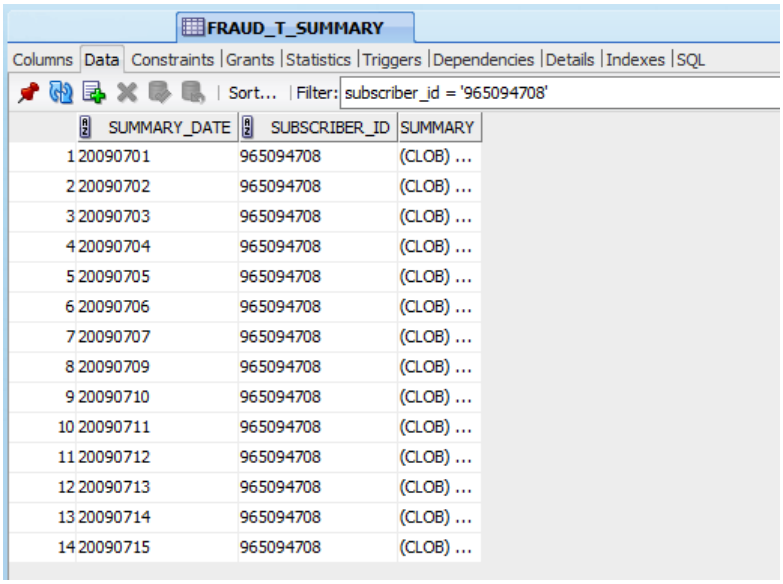
Estes resultados são animadores, pois dão indicações de uma *performance* quantitativa muito boas do protótipo: fazendo a média de eventos por hora é possível ver que o desempenho do protótipo varia entre os 15 e os 25 milhões de eventos processados por hora, o que são realmente números muito bons.

7.2.2. Teste 2 – Detecção

O segundo teste pretende retirar indicações acerca da *performance* qualitativa do processo de detecção. Uma vez que não possuo dados reais, para efectuar este teste utilizei os dados da minha actividade na rede da meu operador do mês de Julho de 2009. O teste consiste em:

1. Processar metade dos dados, até dia 15 inclusive;
2. Marcar o meu número como fraudulento;
3. Processar os restantes dias da minha actividade até ao fim do mês, com um número diferente (mudei de 965094708 para 965094709), sendo que ao fim do processamento de cada dia executava o processo de detecção.

Após executar o primeiro passo, a tabela FRAUD_T_SUMMARY apresenta 15 sumários para a minha actividade na rede, conforme é possível ver na Figura 21.


FRAUD_T_SUMMARY

Columns | Data | Constraints | Grants | Statistics | Triggers | Dependencies | Details | Indexes | SQL

Sort... | Filter: subscriber_id = '965094708'

	SUMMARY_DATE	SUBSCRIBER_ID	SUMMARY
1	20090701	965094708	(CLOB) ...
2	20090702	965094708	(CLOB) ...
3	20090703	965094708	(CLOB) ...
4	20090704	965094708	(CLOB) ...
5	20090705	965094708	(CLOB) ...
6	20090706	965094708	(CLOB) ...
7	20090707	965094708	(CLOB) ...
8	20090709	965094708	(CLOB) ...
9	20090710	965094708	(CLOB) ...
10	20090711	965094708	(CLOB) ...
11	20090712	965094708	(CLOB) ...
12	20090713	965094708	(CLOB) ...
13	20090714	965094708	(CLOB) ...
14	20090715	965094708	(CLOB) ...

Figura 21 – Sumários

Após executar o segundo passo, um registo foi criado na tabela FRAUD_T_FRAUD_CASE, conforme é possível ver na Figura 22. De realçar que os números que estão presentes na minha rede social correspondente a metade do mês são os números de familiares e amigos próximos, que são realmente as pessoas com quem mais frequentemente efectuo e recebo chamadas de voz e mensagens de texto, o que dá boas indicações acerca da *performance* qualitativa do agente de *profiling* em criar a minha rede social.

Após a execução do terceiro passo, o processo de detecção inseriu 13 novos registos na tabela FRAUD_T_SUSPECT, isto porque relativamente ao resto do mês eu só tinha actividade em 13 dias, logo o processo de detecção executou 13 vezes, em vez de executar periodicamente (uma vez por dia por exemplo).

A Figura 23 apresenta os resultados do processo de detecção visualizados numa ferramenta *Web* do Fraud:RAID.

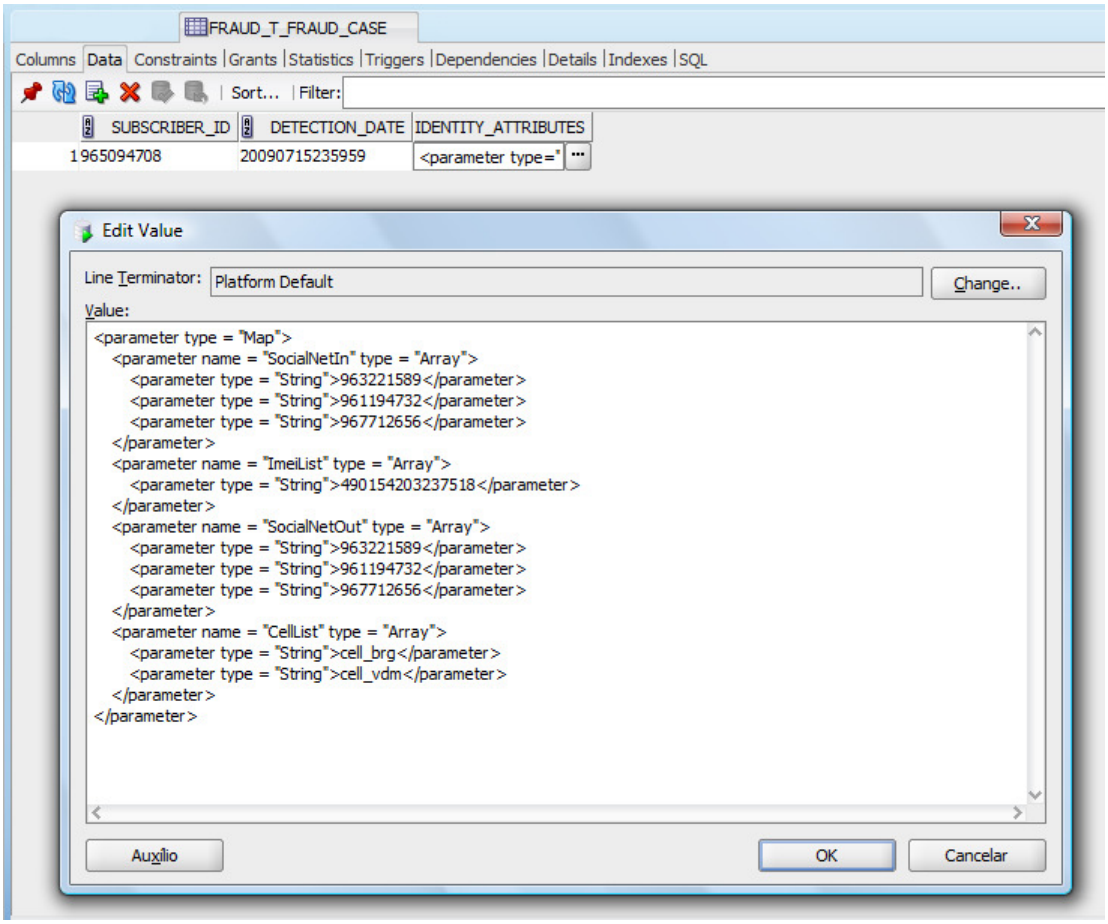


Figura 22 – Caso de fraude

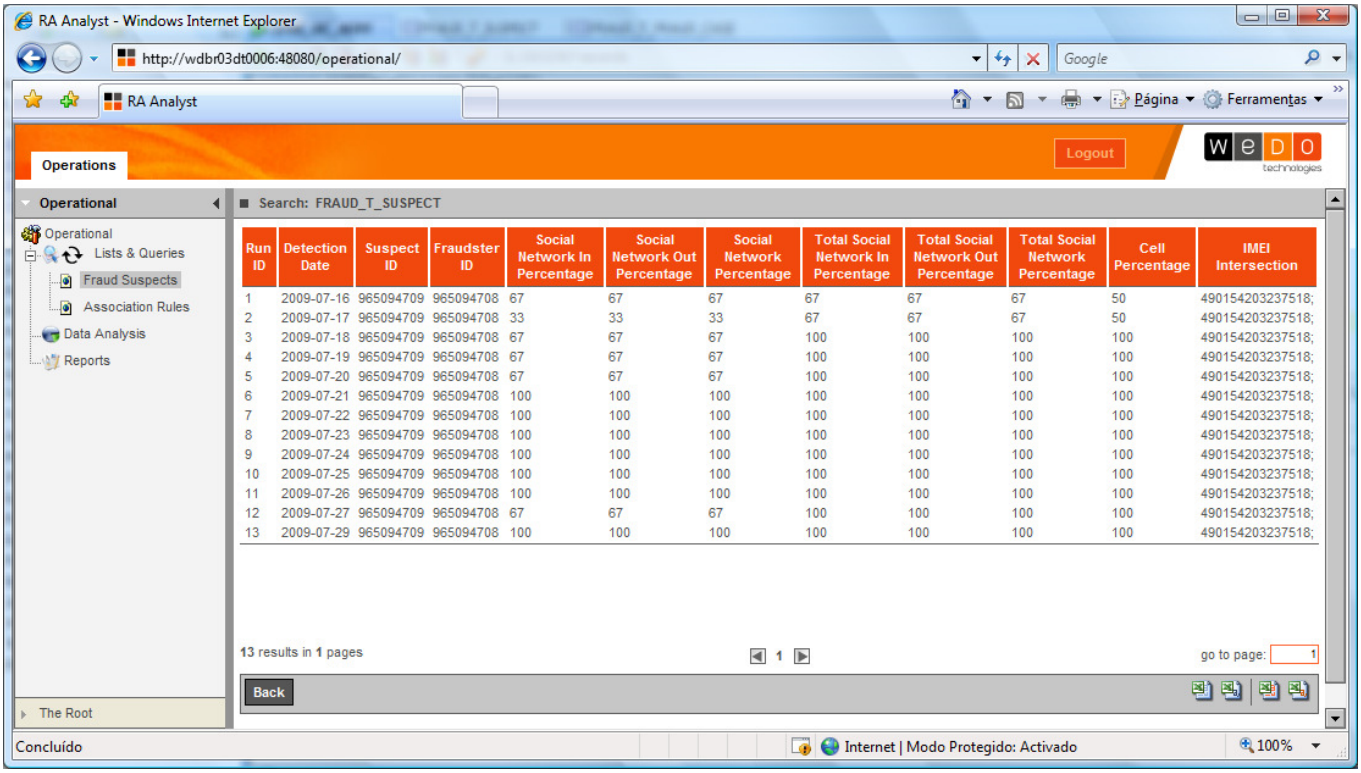


Figura 23 – Suspeitos

7.2.3. Teste 3 – Extracção de conhecimento

O terceiro teste tem como objectivo testar o agente de EC e visualizar os seus resultados. Foram criados manualmente dados para 4 subscritores. Os dois primeiros têm propositadamente mais actividade, mais dispersão de chamadas, serviços e células de rede. O teste consiste em:

1. Processar os dados para os 4 subscritores;
2. Marcar os dois primeiros como fraudulentos por fraude de subscrição, os outros dois servirão como contra-exemplo;
3. Executar o processo de extracção de conhecimento.

Após a execução do primeiro passo, os sumários para os 4 subscritores foram inseridos na tabela FRAUD_T_SUMMARY. Após a execução do segundo passo, foram inseridos dois casos de conhecimento na tabela FRAUD_T_KNOWLEDGE_CASE, conforme se pode ver na Figura 24. Cada um dos casos tem os seus atributos de comportamento definidos na coluna BEHAVIOR_ATTRIBUTES, estruturados em XML.

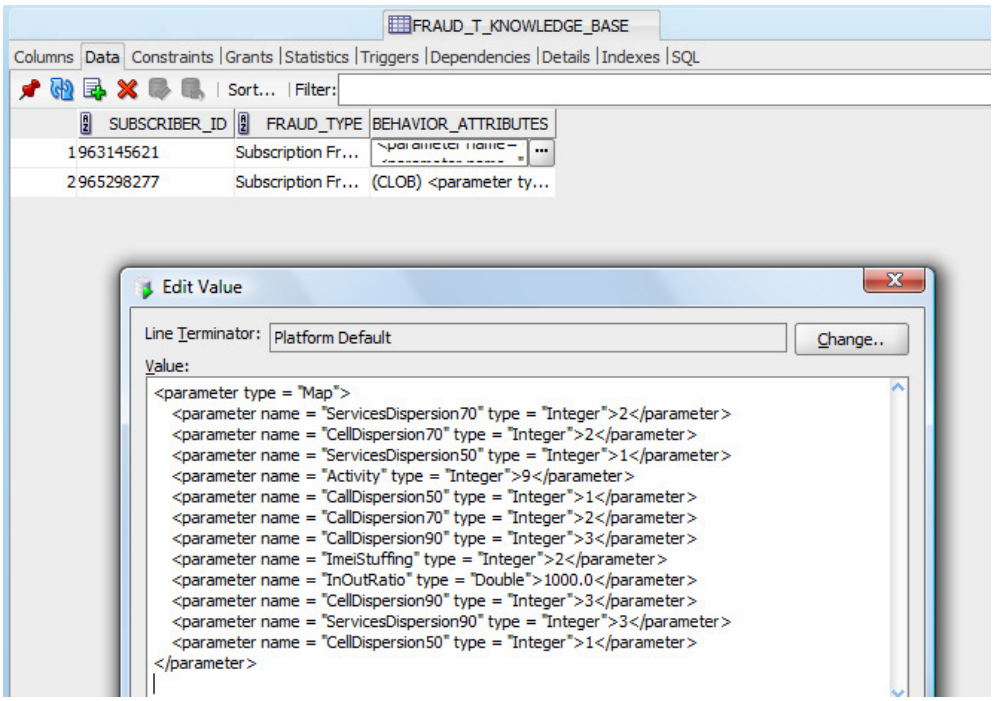


Figura 24 – Casos de conhecimento

Após a execução do terceiro passo foram inseridos vários registos na tabela FRAUD_T_ASSOCIATION_RULE. A Figura 25 apresenta os resultados do processo de extracção de conhecimento visualizados numa ferramenta *Web* do Fraud:RAID.

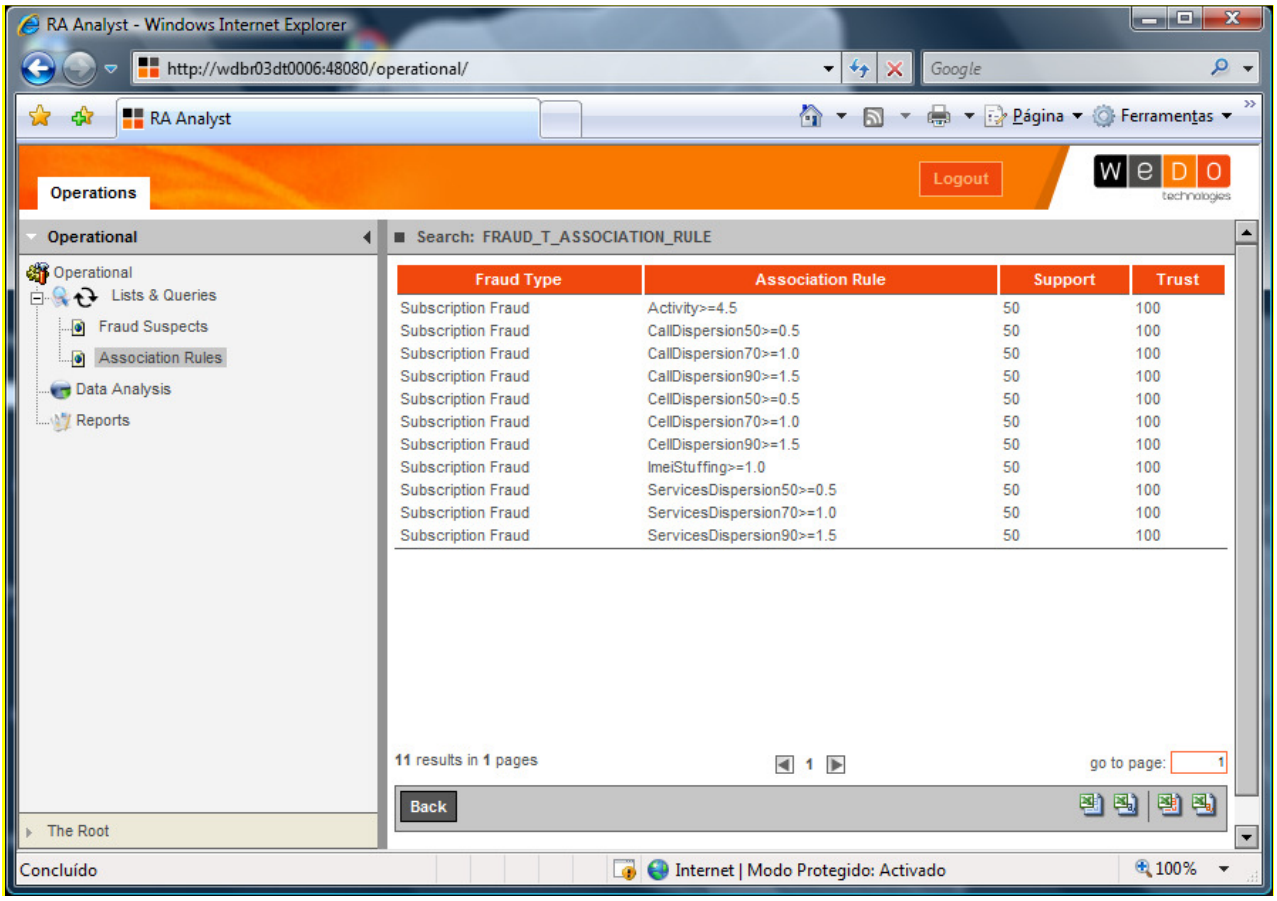


Figura 25 – Regras de Associação

7.3. Análise dos resultados

Analisando a Tabela 16, é possível extrapolar o número de registos que o processo de sumarização consegue processar por hora.

Nº subscritores	Nº total de eventos	Tempo (segundos)	Registos por hora
1.000.000	5.000.000	1267	14.206.788
1.000.000	10.000.000	1411	25.513.820
2.000.000	10.000.000	1712	21.028.037

Tabela 17 – Análise aos resultados do processo de sumarização

A primeira conclusão é que em todos os testes se obtém um **valor estimado bastante bom da capacidade de processamento do processo de sumarização**. O mínimo que se obteve foi perto dos 15 milhões de eventos por hora e o máximo ultrapassou os 25 milhões de eventos por hora. Mas há uma conclusão importante que se deve retirar desta análise: quando se aumentaram o número de eventos e manteve o número de subscritores a *performance* subiu de 14 milhões para mais de 25 milhões de eventos por hora. Mas quando se aumentou o número de subscritores mas manteve o número total de eventos a *performance* desceu de 25 milhões para 21 milhões de eventos por hora. Assim pode-se concluir que **a *performance* depende mais do número de subscritores que do número de eventos total**.

Analisando os resultados do teste de detecção na Figura 23 é possível retirar algumas conclusões que se encontram enquadradas na Figura 26:

1. O processo de detecção executou 13 vezes, uma vez ao fim do processamento de cada dia de dados;
2. A estrutura da informação permite ver que o suspeito identificado como "965094709" foi detectado como sendo o fraudulento "965094708", subscritor previamente banido, a tentar reentrar na rede do operador;
3. O método de detecção por IMEI encontra sempre um IMEI em comum entre o suspeito e o fraudulento;
4. Nas duas primeiras execuções a percentagem de células mais usadas pelo suspeito em comum com as mais usadas pelo fraudulento era de 50%, após a 3ª execução o valor atinge os 100%;
5. Os valores de TOTAL_SNI_PERCENTAGE, TOTAL_SNO_PERCENTAGE e TOTAL_SN_PERCENTAGE começam com 67% dos contactos em comum e depois cresce para 100% e estabiliza;
6. Os valores de TOTAL começam com 67% dos contactos em comum e depois cresce para 100% mas mais tarde descem para 67% e terminam em novamente em 100%;

Run ID	Detection Date	Suspect ID	Fraudster ID	Social Network In Percentage	Social Network Out Percentage	Social Network Percentage	Total Social Network In Percentage	Total Social Network Out Percentage	Total Social Network Percentage	Cell Percentage	IMEI Intersection
1	2009-07-16	965094709	965094708	67	67	67	67	67	67	50	490154203237518
2	2009-07-17	965094709	965094708	33	33	33	67	67	67	50	490154203237518
3	2009-07-18	965094709	965094708	67	67	67	100	100	100	100	490154203237518
4	2009-07-19	965094709	965094708	67	67	67	100	100	100	100	490154203237518
5	2009-07-20	965094709	965094708	67	67	67	100	100	100	100	490154203237518
6	2009-07-21	965094709	965094708	100	100	100	100	100	100	100	490154203237518
7	2009-07-22	965094709	965094708	100	100	100	100	100	100	100	490154203237518
8	2009-07-23	965094709	965094708	100	100	100	100	100	100	100	490154203237518
9	2009-07-24	965094709	965094708	100	100	100	100	100	100	100	490154203237518
10	2009-07-25	965094709	965094708	100	100	100	100	100	100	100	490154203237518
11	2009-07-26	965094709	965094708	100	100	100	100	100	100	100	490154203237518
12	2009-07-27	965094709	965094708	67	67	67	100	100	100	100	490154203237518
13	2009-07-29	965094709	965094708	100	100	100	100	100	100	100	490154203237518
	1.	2.		6.			5.			4.	3.

Figura 26 – Análise aos resultados da detecção

A razão pela qual os valores de TOTAL_SNI_PERCENTAGE, TOTAL_SNO_PERCENTAGE e TOTAL_SN_PERCENTAGE crescem até 100% e depois estabilizam nesse valor é porque estes valores representam a percentagem de contactos da rede social do fraudulento com que o suspeito teve alguma actividade, mesmo que seja só um evento. Por isso é normal que este valor nunca diminua.

Por outro lado, os valores de SNI_PERCENTAGE, SNO_PERCENTAGE e SN_PERCENTAGE crescem e diminuem é porque representam a percentagem de contactos da rede social do fraudulento que também estão na rede social do suspeito, e como a rede social do suspeito vai evoluindo ao longo do tempo, pode acontecer que contactos da rede social do suspeito que eram comuns com a rede social do fraudulento sejam removidos por falta de actividade. Assim é lógico que os valores destes atributos que comparam as redes sociais do suspeito e fraudulento diminuam de percentagem.

Analisando a Figura 25, que contém os resultados do teste de extracção de conhecimento, pode-se concluir que:

1. O processo criou regras de associação para tipo de fraude “Subscription Fraud”;
2. As regras de associação estão quantificadas, a cada atributo de comportamento está associada uma medida, por exemplo, “Activity>=4.5”;
3. Todas as regras são “perfeitas”, pois têm 50% de suporte e 100% de confiança. Isto significa que a regra identifica todos os casos de fraude presentes no conjunto de entrada e todos os casos que identifica são efectivamente casos de fraude. O valor real destas regras é insignificativo, pois os valores de entrada

foram construídos manualmente, sendo os dados dos subscritores que seriam marcados como fraudulentos alterados propositadamente para terem maior actividade, dispersão de chamadas, serviços e células. O objectivo deste teste era meramente verificar que o processo funciona.

Fraud Type	Association Rule	Support	Trust
Subscription Fraud	Activity>=4.5	50	100
Subscription Fraud	CallDispersion50>=0.5	50	100
Subscription Fraud	CallDispersion70>=1.0	50	100
Subscription Fraud	CallDispersion90>=1.5	50	100
Subscription Fraud	ImeiStuffing>=1.0	50	100
Subscription Fraud	ServicesDispersion90>=1.5	50	100
Subscription Fraud	CellDispersion70>=1.0	50	100
Subscription Fraud	CellDispersion90>=1.5	50	100
Subscription Fraud	ServicesDispersion50>=0.5	50	100
Subscription Fraud	ServicesDispersion70>=1.0	50	100
Subscription Fraud	CellDispersion50>=0.5	50	100
1.	2.	3.	

Figura 27 – Análise aos resultados da extracção de conhecimento

8. Conclusões

O presente capítulo serve dois propósitos: na primeira secção é efectuada uma revisão dos objectivos definidos inicialmente, seguindo-se uma análise individual a cada um dos objectivos e por fim é apresentada uma reflexão crítica sobre o trabalho desenvolvido ao longo da presente tese; na segunda secção é descrito o trabalho futuro onde é apresentada uma avaliação crítica do que será a evolução dos resultados da presente tese.

8.1. Reflexão crítica

Relembrando os objectivos definidos inicialmente:

- Estudar, analisar e detectar oportunidades de evolução da actual solução de detecção de fraude;
- Concretizar as oportunidades de evolução da solução, definindo um novo modelo de detecção de fraude;
- Implementar um protótipo de forma a ser possível obter indicações acerca da performance (qualitativa e quantitativa) do novo modelo.

A primeira fase da realização da presente tese consistiu no estudo do problema em questão, a fraude nas telecomunicações, e na análise das soluções existentes para a gestão de fraude, os FMS (Fraud Management System). Foram estudados os principais tipos de fraude que afectam os operadores de telecomunicações para melhor se compreender o problema e foram analisados vários FMS bem como vários estudos publicados no âmbito da detecção de fraude. O trabalho efectuado na primeira fase permitiu retirar as seguintes conclusões acerca das soluções para gestão de fraude:

- São modelos reactivos, foram concebidos para reagir a comportamentos fraudulentos por parte dos subscritores. A detecção de um fraudulento só é

possível após este ter efectivamente cometido a fraude e consequentemente ter causado danos ao operador;

- Baseiam-se em métodos orientados à detecção por comportamento e não suportam métodos orientados à detecção por identidade dos fraudulentos. Quando um subscritor é detectado como fraudulento é banido da rede do operador. O problema é que não existem métodos para prevenir que este subscritor reentre na rede do operador. Deste modo, o fraudulento só será novamente detectado quando tiver novamente um comportamento fraudulento e consequentemente ter causado novamente danos no operador;
- Não existem processos orientados à extracção do conhecimento. Quando um subscritor é detectado como fraudulento é banido da rede do operador, contudo os seus dados de comportamento não são aproveitados para tentar retirar conhecimento acerca do comportamento do fraudulento, permitindo assim aos analistas de fraude melhorarem o seu *know-how* nesta área.

A partir destas conclusões foram definidas duas oportunidades de evolução das actuais soluções para gestão de fraude:

1. Desenvolvimento de métodos de detecção por identidade;
2. Desenvolvimento de métodos de extracção de conhecimento.

Na segunda fase definiu-se uma nova abordagem ao problema com dois principais objectivos: a detecção por identidade de fraudulentos previamente banidos a tentarem reentrar na rede do operador e a extracção de conhecimento a partir do comportamento dos subscritores fraudulentos.

Esta nova abordagem foi desenvolvida usando como suporte a metodologia dos Sistemas Multi-Agente. A utilização desta metodologia revelou-se fundamental para o sucesso do desenvolvimento da nova solução. Os Sistemas Multi-Agente são adequados para o desenvolvimento de soluções complexas, como é o caso. As suas características como a modularidade, extensibilidade, flexibilidade e eficiência, permitiram desenvolver cada agente separadamente, que executam as suas tarefas autonomamente e ir adicionando agentes à medida que se ia desenvolvendo a

solução. O Sistema Multi-Agente é composto por três agentes, cada um com objectivos, funcionalidades e processos bem definidos.

O primeiro agente, o agente de *profiling*, utiliza a técnica de *profiling*, uma técnica que é usada em várias áreas com bons resultados. A aplicação desta técnica dividindo o perfil em duas partes, identidade e comportamento, permite servir dois propósitos: os atributos de identidade permitirão ao agente de detecção identificar fraudulentos previamente banidos a tentarem reentrar na rede e os atributos de comportamento permitirão ao agente de extracção de conhecimento identificar padrões no comportamento dos fraudulentos.

O agente de detecção contém dois métodos, ambos com o objectivo de detectar fraudulentos previamente banidos a reentrar na rede do operador: associação por actividade social e associação por IMEI. No primeiro método a aplicação da técnica de Análise de Redes Sociais (ARS) permite que não seja necessário monitorizar todos os subscritores da rede de forma a detectar a reentrada de um fraudulento, apenas monitorizando os subscritores com quem o fraudulento tinha uma interacção mais frequente. Quanto ao segundo método, associação por IMEI, já existem operadores que fazem detecção baseada exclusivamente em IMEI, adicionando o IMEI do equipamento que o fraudulento usava na altura em que foi detectado. Este método é bem mais eficiente, por duas razões:

1. Não usa só o IMEI do fraudulento na altura da detecção. Usa os IMEIs que o fraudulento tem no seu histórico.
2. Não compara apenas os IMEIs dos suspeitos que eles estão a utilizar após o momento da detecção do fraudulento com que estão a ser comparados. Compara também os IMEIs do seu histórico.

O último agente, de extracção de conhecimento, permite cobrir uma lacuna dos FMS: a desvalorização e falta de métodos orientados ao conhecimento. Este agente, baseado na área de Descoberta de Conhecimento em Base de Dados (DCBD), identifica padrões no comportamento dos fraudulentos através da análise dos atributos de comportamento provenientes do agente de *profiling*. Apesar de não ter um impacto directo na detecção de fraude, os seus resultados permitirão aos analistas de fraude

melhorar o seu *know-how* no que diz respeito ao comportamento dos fraudulentos e assim optimizar os seus recursos e métodos para detecção.

Gostava ainda de destacar o facto de ter sido submetido um artigo científico no Encontro Português de Inteligência Artificial (EPIA) (52) com o título "Telecommunications Fraud: Problem analysis - an agent-based KDD perspective" (53) cujo conteúdo é a definição desta nova abordagem definida sobre um Sistema Multi-Agente. Este trabalho foi aceite, publicado e apresentado no EPIA'2009, em Aveiro, na 14ª conferência portuguesa de Inteligência Artificial, tendo recebido boas críticas no final do processo.

A última fase, a implementação de um protótipo, foi a de maior dificuldade na realização desta tese, devido ao facto de não ser possível (por motivos legais) obter um volume aceitável de dados reais para usar no protótipo. Assim, os objectivos para a realização desta fase foram afectados. Não foi possível obter conclusões sólidas acerca da *performance* qualitativa da solução desenvolvida, pelo que objectivo principal foi desenvolver as ferramentas necessárias para que possam ser usadas numa fase posterior, quando for possível obter dados reais. Para obter algumas indicações acerca da *performance* do protótipo foi utilizado um pequeno conjunto de dados, constituído pelos meus registos, bem como os de alguns familiares e amigos que gentilmente me cederam os seus registos.

Após o desenvolvimento do protótipo foi efectuado um conjunto de testes sobre o mesmo, que permitiu retirar algumas conclusões:

- Em termos quantitativos o protótipo teve resultados muito bons, apresentando uma capacidade de processamento na ordem dos 15 aos 25 milhões de eventos por hora. Outra conclusão retirada destes testes é que o número de subscritores envolvidos também tem um impacto na *performance* que não deve ser ignorado.
- Em termos qualitativos o protótipo deixou boas indicações, tanto na construção dos atributos de identidade, mais concretamente na construção da rede social, como no método de detecção por associação por actividade social.

- Em termos de extracção de conhecimento, a falta de dados reais não permitiu mais do que testar a funcionalidade e a apresentação dos resultados.

No geral, o protótipo apresentou resultados muito bons, com uma *performance* quantitativa elevada e com muito boas indicações quanto à *performance* qualitativa.

8.2. Trabalho futuro

Conforme já foi referido, uma das maiores dificuldades na realização desta tese foi a impossibilidade por motivos legais de obter um volume aceitável de dados reais para usar no protótipo. A obtenção de dados reais é crítica para a evolução dos resultados da presente tese.

Como trabalho futuro é proposto o seguinte plano:

- Definir e realizar um conjunto de testes, usando dados reais, sobre a solução desenvolvida. Este conjunto de testes deve ser definido com o objectivo de que os seus resultados possam reflectir o desempenho da nova abordagem, tanto a nível qualitativo como a nível quantitativo.
- Após a análise dos resultados obtidos na fase anterior deve ser avaliada a solução do ponto de vista da definição do Sistema Multi-Agente (SMA). O objectivo é determinar se é possível melhorar a solução modificando as características do SMA: avaliar a necessidade de mudar a arquitectura, coordenação e organização do SMA de forma a obter melhores resultados; avaliar a necessidade de mais agentes (replicar os agentes para executar as mesmas tarefas de forma a melhorar o desempenho ou definir novos agentes para executar novas tarefas); avaliar as características dos agentes e detectar que características podem ser adicionadas para contribuir para uma solução melhor (introduzir proactividade em alguns agentes e processos, por exemplo).
- Ainda no âmbito da fase anterior, deve ser analisado individualmente cada processo de cada agente. Os algoritmos (sumarização, construção de perfis, detecção por associação e detecção por IMEI) e a definição dos atributos

(sumários, perfis, casos de fraude, casos de comportamentos) devem ser questionados e avaliados sob a perspectiva de tornar a solução mais eficiente.

- Por fim, deve ser efectuada uma comparação dos resultados da aplicação da nova abordagem com a solução actual de gestão de fraude, de forma a ser possível obter uma indicação da vantagem da aplicação desta nova abordagem.

Os resultados da realização da presente tese serão apresentados na empresa WeDo Technologies. Numa primeira fase será necessária a avaliação da nova abordagem e consequentemente possíveis alterações na definição dos agentes, processos e atributos de forma a adaptar a solução ao produto. Numa segunda fase será necessário implementar na solução de gestão de fraude da WeDo, o produto Fraud:RAID, as novas funcionalidades desenvolvidas nesta tese.

Bibliografia

1. **CFCA.** Communications Fraud Control Association. *Communications Fraud Control Association.* [Online] http://www.cfca.org/.

2. **TUFF.** Telecommunications UK Fraud Forum. *Telecommunications UK Fraud Forum.* [Online] http://www.tuff.co.uk/.

3. **GSM.** GSM Association. *GSM Association.* [Online] http://www.gsmworld.com/.

4. **O'Brien, John T.** Telecommunications Fraud. *The FBI Law Enforcement Bulletin.* 1998.

5. **Detica.** Detica. *Detica.* [Online] www.detica.com.

6. **Azure.** Azure Solutions. *Azure Solutions.* [Online] www.nvpllc.com/Portfolio/Details/Azure_Solutions.

7. **Agilis.** Agilis International. *Agilis International.* [Online] www.agilisinternational.com.

8. **Telbit.** Telbit. *Telbit.* [Online] www.telbit.pt/product/centaur.

9. **Estévez, P.A.; Held, C.M.; Perez, C.A.** Subscription Fraud Prevention in Telecommunications Using Fuzzy Rules and Neural Networks. *Expert Systems with Applications.* 2006, Vol. 31, 2. http://www.cec.uchile.cl/~pestevez/RI0.pdf.

10. **Fawcett, Tom; Provost, Foster.** Combining Data Mining and Machine Learning for Effective User Profiling. 1996. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.35.1774.

11. **Fawcett, Tom; Provost, Foster.** Adaptive Fraud Detection. 1997. Vol. 1, pp. 291-316. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.35.2902.

12. **Taniguchi, Michiaki; Haft, Michael; Hollmén, Jaakko; Tresp; Volker.** Fraud Detection In Communications Networks Using Neural And Probabilistic Methods.

Proceedings of the 1998 IEEE Int. 1998. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.21.5355.

13. **Hollmén, Jaakko.** Novelty filter for fraud detection in mobile communications networks. *Technical Report A48.* 1997.

14. **Burge, Peter; Shawe-Taylor, John.** *Detecting Cellular Fraud Using Adaptive Prototypes.* Communications Technologies Project. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.49.1156.

15. **Burge, Pete; Shawe-Taylor, John.** An unsupervised neural network approach to profiling the behavior of mobile phone users for use in fraud detection. *J. Parallel Distrib. Comput.* 2001, Vol. 61, 7, pp. 915-925. http://portal.acm.org/citation.cfm?id=569546.569551&dl=GUIDE&dl=ACM&CFID=25752378&CFTOKEN=65343834#.

16. **Cortes, Corinna; Pregibon, Daryl.** Signature-Based Methods for Data Streams. *Data Min. Knowl. Discov.* 2001, Vol. 5, 3, pp. 167-182. http://portal.acm.org/citation.cfm?id=593512.

17. **Cortes, Corinna; Pregibon, Daryl; Volinsky, Chris.** Communities of Interest. *IDA '01: Proceedings of the 4th International Conference on Advances in Intelligent Data Analysis.* London, UK : Springer-Verlag, 2001, pp. 105-114. http://portal.acm.org/citation.cfm?id=741620.

18. **Cortes, Corinna; Pregibon, Daryl; Volinsky, Chris.** Computational Methods for Dynamic Graphs. 2003. http://pubs.amstat.org/doi/abs/10.1198/1061860032742.

19. **Ferreira, Pedro; Alves, Ronnie; Belo, Orlando; Cortesão, Luís.** Establishing Fraud Detection Patterns Based on Signatures. *Signatures, Industrial Conference on Data Mining, LNAI.* s.l. : SpringerVerlag, 2006. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.62.9950.

20. **Alves, Ronnie; Ferreira, Pedro; Belo, Orlando; Lopes, Joao.** Discovering telecom fraud situations through mining anomalous behavior patterns. 2006. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.65.3293.

21. **Weiss, Gary.** Data Mining in Telecommunications. *Data Mining and Knowledge Discovery Handbook: A Complete Guide for Practitioners and Researchers, Kluwer Academic, 2005.* s.l. : kluwer, 2004, pp. 1189-1201. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.60.955.

22. **Bond, Alan; Grasser, Les.** Readings in Distributed Artificial Intelligence. San Mateo, USA : Morgan Kaufmann Publishers, 1988.

23. **Weiss, Gerhard.** Multiagent Systems, A Modern Approach to Distributed Artificial Intelligence. *MIT Press.* USA : s.n., 1999.

24. **Durfee, Edmund; Rosenschein, Jeffrey.** Distributed Problem Solving and Multi Agent Systems: Comparisons and Examples. Seattle : s.n., 1994.

25. **Wooldridge, Michael J.** Intelligent Agents. [ed.] Weiss. Cambridge, USA : MIT Press, 1999. pp. 27-77.

26. **Wooldridge, Michael J.; Jennings, Nicholas R.** Intelligent Agents: Theory and Practice. [ed.] Knowledge Engineering Review. 1995. pp. 115-152.

27. **Olson, G.M.; Malone, T.W.; Smith, J.B.** Coordination Theory and Collaboration Technology. s.l. : Prentice Hall International Inc., 2001.

28. **Russell, Stuart; Norvig, Peter.** Artificial Intelligence, A Modern Approach. USA : Prentice Hall International Inc., 1995.

29. **Wooldrige, M.** An Introduction to Multiagent Systems. USA : John Wiley and Sons, 2002. 0-47149691X.

30. **CORREIA, Elisabete; LUCAS, Susana; LAMIA, Alicia.** Profiling: Uma técnica auxiliar de investigação criminal. *Análise Psicológica.* 2007.

31. **McHugh, John; McLeod, Ron; Nagaonkar, Vagishwari.** Passive network forensics: behavioural classification of network hosts based on connection patterns. [ed.] SIGOPS Oper. Syst. Rev. New York, USA : ACM, 2008. Vol. 42, 3, pp. 99-111. 0163-5980.

32. **McCrary, G.** Le profilage criminel à l'interieur et à l'extérieur du tribunal. s.l. : PUF, 2001.

33. **Wrightsman, L. S.** Forensic psychology. s.l. : Wadsworth, 2001.

34. **Dasgupta, Koustuv; Singh, Rahul; Viswanathan, Balaji; Chakraborty, Dipanjan; Mukherjea, Sougata; Nanavati, Amit A.; Joshi, Anupam.** Social ties and their relevance to churn in mobile telecom networks. *EDBT '08: Proceedings of the 11th international conference on Extending database technology.* Nantes : ACM, 2008, pp. 668-677.

35. **Du, Nan; Wu, Bin; Pei, Xin; Wang, Bai; Xu, Liutong.** Community detection in large-scale social networks. *WebKDD/SNA-KDD '07: Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis.* San Jose : ACM, 2007, pp. 16-25.

36. **Stolfo, Salvatore J.; Hershkop, Shlomo; Hu, Chia-Wei; Li, Wei-Jen; Nimeskern, Olivier; Wang, Ke.** Behavior-based modeling and its application to Email analysis. [ed.] ACM. 2006, Vol. 6, 2, pp. 187-221.

37. **McHugh, John; McLeod, Ron; Nagaonkar, Vagishwari.** Passive network forensics: behavioural classification of network hosts based on connection patterns. *SIGOPS Oper. Syst. Rev.* 2008, Vol. 42, 3, pp. 99-111.

38. **Xu, Kuai; Zhang, Zhi-Li; Bhattacharyya, S.** Internet Traffic Behavior Profiling for Network Security Monitoring. *IEEE/ACM Transactions on Networking.* 2008, Vol. 16, 6, pp. 1241-1252.

39. **Cadez, Igor V.; Smyth, Padhraic; Mannila, Heikki.** Probabilistic modeling of transaction data with applications to profiling, visualization, and prediction. *KDD '01: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining.* San Francisco : ACM, 2001, pp. 37-46.

40. **Teng, Wei-Guang; Chou, Ming-Chia.** Mining communities of acquainted mobile users on call detail records. *SAC '07: Proceedings of the 2007 ACM symposium on Applied computing.* Seoul : ACM, 2007, pp. 957-958.

41. **Dasgupta, Koustuv; Singh, Rahul; Viswanathan, Balaji; Chakraborty, Dipanjan; Mukherjea, Sougata; Nanavati, Amit A.; Joshi, Anupam.** Social ties and their relevance to churn in mobile telecom networks. *EDBT '08: Proceedings of the 11th international*

conference on Extending database technology. s.l. : ACM, 2008, pp. 668-677. http://doi.acm.org/10.1145/1353343.1353424.

42. **Valentim, Marta.** Métodos de Pesquisa: Análise de Redes Sociais. s.l., Marília, Brasil : Universidade Estadual Paulista, 2008.

43. **Haythornthwaite, Caroline.** Apoiando interações à distância: um estudo sobre interacções e uso dos media ao longo do tempo. Illinois , USA : University of Illinois , June de 2009 .

44. **Stolfo, Salvatore J.; Hershkop, Shlomo; Hu, Chia-Wei; Li, Wei-Jen; Nimeskern, Olivier; Wang, Ke.** Behavior-based modeling and its application to Email analysis. New York, NY, USA : ACM, 2006. Vol. 6, 2, pp. 187-221. http://doi.acm.org/10.1145/1149121.1149125. 1533-5399.

45. **Kumar, Ravi; Raghavan, Prabhakar; Rajagopalan, Sridhar; Tomkins, Andrew.** The Web and Social Networks. Los Alamitos, CA, USA : IEEE Computer Society Press, 2002. Vol. 35, 11, pp. 32-36. http://dx.doi.org/10.1109/MC.2002.1046971. 0018-9162.

46. **Chawathe, Sudarshan S.** Tracking Hidden Groups Using Communications. [autor do livro] Computer Science Department e Sudarshan S. Chawathe. *ISI.* 2003, pp. 195-208. http://link.springer.de/link/service/series/0558/bibs/2665/26650195.htm.

47. **Du, Nan; Wu, Bin; Pei, Xin; Wang, Bai; Xu, Liutong.** Community detection in large-scale social networks. *WebKDD/SNA-KDD '07: Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis.* s.l. : ACM, 2007. http://doi.acm.org/10.1145/1348549.1348552.

48. **Liben-Nowell, David; Kleinberg, Jon.** The link prediction problem for social networks. *CIKM '03: Proceedings of the twelfth international conference on Information and knowledge management.* s.l. : ACM, 2003, pp. 556-559. http://doi.acm.org/10.1145/956863.956972.

49. **Xu, Jennifer J.; Chen, Hsinchun.** CrimeNet explorer: a framework for criminal network knowledge discovery. s.l. : ACM, 2005. Vol. 23, 2, pp. 201-226. http://doi.acm.org/10.1145/1059981.1059984. 1046-8188.

50. **Teng, Wei-Guang; Chou, Ming-Chia.** Mining communities of acquainted mobile users on call detail records. *SAC '07: Proceedings of the 2007 ACM symposium on Applied computing.* s.l. : ACM, 2007, pp. 957-958. http://doi.acm.org/10.1145/1244002.1244212.

51. **Santos, M. Filipe; Azevedo, Carla.** *Data Mining - Descoberta de conhecimento em bases de dados.* s.l. : FCA, 2005.

52. **EPIA'2009.** EPIA 2009. *EPIA 2009.* [Online] http://epia2009.web.ua.pt/.

53. **Rosas, Eugénio; Analide, Cesar.** Telecommunications Fraud: Problem analysis - an agent-based KDD perspective. Aveiro : EPIA 2009, 2009.

Anexos

Anexo I – Algoritmo de sumarização

```
1:  sumariza(evento){
2:
3:     /* PASSO_1: Ir buscar os campos do evento */
4:     A_NUMBER = evento.getA_NUMBER();
5:     B_NUMBER = evento.getB_NUMBER();
6:     EVENT_DATE = evento.getEVENT_DATE();
7:     EVENT_TYPE = evento.getEVENT_TYPE();
8:     EVENT_AMOUNT = evento.getEVENT_AMOUNT();
9:     CELL_ID_A = evento.getCELL_ID_A();
10:    IMEI_A = evento.getIMEI_A();
11:    CELL_ID_B = evento.getCELL_ID_B();
12:    IMEI_B = evento.getIMEI_B();
13:
14:    /* PASSO_2: Calcular o dia no formato ano-mês-dia */
15:    EVENT_DAY=getDAY(EVENT_DATE);
16:
17:    /* PASSO_3:
18:     * Verificar se o sumário já existe para
19:     * o subscritor A_NUMBER para o dia EVENT_DAY
20:     * no repositório de sumários */
21:    Se ( REPOSITORIO.contem_sumario(A_NUMBER,EVENT_DAY) )
22:    Então
23:       /* PASSO_3a: Ir buscar o sumário para actualizar */
24:       SUMARIO = REPOSITORIO.getSUMARIO(A_NUMBER,EVENT_DAY);
25:    Senão
26:       /* PASSO_3b: Construir novo perfil */
27:       SUMARIO = new SUMARIO();
28:    Fim Se
29:
30:    /* PASSO_4: tratar destino */
31:    CALLED = SUMARIO.getCALLED_LIST();
32:    Se ( CALLED.contem(B_NUMBER) )
33:    Então
34:       NUMERO_EVENTOS = CALLED.get(B_NUMBER)
35:       CALLED.set(B_NUMBER,NUMERO_EVENTOS+1)
36:    Senão
37:       CALLED.adiciona(B_NUMBER,1);
38:    Fim Se
39:    SUMARIO.setCALLED_LIST(CALLED);
40:
41:    /* PASSO_5: tratar célula de rede */
42:    Se ( CELL_ID_A tem valor )
43:    Então
44:       CELL_USAGE = SUMARIO.getCELL_USAGE_LIST();
45:       Se ( CELL_USAGE.contem(CELL_ID_A) )
46:       Então
47:          NUMERO_EVENTOS = CELL_USAGE.get(CELL_ID_A)
48:          CELL_USAGE.set(CELL_ID_A,NUMERO_EVENTOS+1)
49:       Senão
50:          CELL_USAGE.adiciona(CELL_ID_A,1);
51:       Fim Se
52:       SUMARIO.setCELL_USAGE_LIST(CELL_USAGE);
53:    Fim Se
54:
55:    /* PASSO_6: tratar serviço */
56:    SERVICE_USAGE = SUMARIO.getSERVICE_USAGE_LIST();
57:    Se ( SERVICE_USAGE.contem(EVENT_TYPE) )
```

```
58:   Então
59:     NUMERO_EVENTOS = SERVICE_USAGE.get(EVENT_TYPE)
60:     SERVICE_USAGE.set(EVENT_TYPE,NUMERO_EVENTOS+1)
61:   Senão
62:     SERVICE_USAGE.adiciona(EVENT_TYPE,1);
63:   Fim Se
64:   SUMARIO.setSERVICE_USAGE_LIST(SERVICE_USAGE);
65:
66:   /* PASSO_7: tratar IMEI */
67:   Se ( IMEI_A tem valor )
68:   Então
69:     IMEI_LIST = SUMARIO.getIMEI_LIST();
70:     Se ( Não ( IMEI_LIST.contem(IMEI_A) ) )
71:     Então
72:       IMEI_LIST.adiciona(IMEI_A);
73:     Fim Se
74:     SUMARIO.setIMEI_LIST(IMEI_LIST);
75:   Fim Se
76:
77:   /* PASSO_8: registar actividade */
78:   ACTIVITY = SUMARIO.getACTIVITY();
79:   INDICE_PERIODO = getINDICE_PERIODO(EVENT_DATE);
80:   ACTIVITY[INDICE_PERIODO]=1;
81:   SUMARIO.setACTIVITY(ACTIVITY);
82:
83:   /* PASSO_9: guardar sumário */
84:   REPOSITORIO.setSUMARIO(A_NUMBER,EVENT_DAY,SUMARIO);
85:
86:   /* PASSO_10: verificar se B_NUMER pertence ao operador */
87:   Se ( B_NUMBER pertece ao operador )
88:   Então
89:   /* PASSO_11:
90:      * Verificar se o sumário já existe para
91:      * o subscritor B_NUMBER para o dia EVENT_DAY
92:      * no repositório de sumários */
93:     Se ( REPOSITORIO.contem_sumario(B_NUMBER,EVENT_DAY) )
94:     Então
95:       /* PASSO_11a: Ir buscar o sumário para actualizar */
96:       SUMARIO = REPOSITORIO.getSUMARIO(B_NUMBER,EVENT_DAY);
97:     Senão
98:       /* PASSO_11b: Construir novo perfil */
99:       SUMARIO = new SUMARIO();
100:    Fim Se
101:
102:    /* PASSO_12: tratar origem */
103:    RECEIVED = SUMARIO.getRECEIVED_LIST();
104:    Se ( RECEIVED.contem(A_NUMBER) )
105:    Então
106:      NUMERO_EVENTOS = RECEIVED.get(A_NUMBER)
107:      RECEIVED.set(A_NUMBER,NUMERO_EVENTOS+1)
108:    Senão
109:      RECEIVED.adiciona(A_NUMBER,1);
110:    Fim Se
111:    SUMARIO.setRECEIVED_LIST(RECEIVED);
112:
113:    /* PASSO_13: tratar célula de rede */
114:    Se ( CELL_ID_B tem valor )
115:    Então
116:      CELL_USAGE = SUMARIO.getCELL_USAGE_LIST();
117:      Se ( CELL_USAGE.contem(CELL_ID_B) )
118:      Então
119:        NUMERO_EVENTOS = CELL_USAGE.get(CELL_ID_B)
120:        CELL_USAGE.set(CELL_ID_B,NUMERO_EVENTOS+1)
```

```
121:          Senão
122:             CELL_USAGE.adiciona(CELL_ID_B,1);
123:          Fim Se
124:          SUMARIO.setCELL_USAGE_LIST(CELL_USAGE);
125:       Fim Se
126:
127:       /* PASSO_14: tratar serviço */
128:       SERVICE_USAGE = SUMARIO.getSERVICE_USAGE_LIST();
129:       Se ( SERVICE_USAGE.contem(EVENT_TYPE) )
130:       Então
131:          NUMERO_EVENTOS = SERVICE_USAGE.get(EVENT_TYPE)
132:          SERVICE_USAGE.set(EVENT_TYPE,NUMERO_EVENTOS+1)
133:       Senão
134:          SERVICE_USAGE.adiciona(EVENT_TYPE,1);
135:       Fim Se
136:       SUMARIO.setSERVICE_USAGE_LIST(SERVICE_USAGE);
137:
138:       /* PASSO_15: tratar IMEI */
139:       Se ( IMEI_B tem valor )
140:       Então
141:          IMEI_LIST = SUMARIO.getIMEI_LIST();
142:          Se ( Não ( IMEI_LIST.contem(IMEI_B) ) )
143:          Então
144:             IMEI_LIST.adiciona(IMEI_B);
145:          Fim Se
146:          SUMARIO.setIMEI_LIST(IMEI_LIST);
147:       Fim Se
148:
149:       /* PASSO_16: registar actividade */
150:       ACTIVITY = SUMARIO.getACTIVITY();
151:       INDICE_PERIODO = getINDICE_PERIODO(EVENT_DATE);
152:       ACTIVITY[INDICE_PERIODO]=1;
153:       SUMARIO.setACTIVITY(ACTIVITY);
154:
155:       /* PASSO_17: guardar sumário */
156:       REPOSITORIO.setSUMARIO(B_NUMBER,EVENT_DAY,SUMARIO);
157:    Fim Se
158: }
```