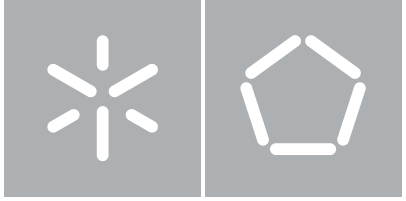
Universidade do Minho
Escola de Engenharia

Eduardo Luís da Silva Lopes

Deteção de Anomalias em Modelos de Publicidade Pay-Per-Click

Outubro de 2012



Universidade do Minho
Escola de Engenharia

Eduardo Luís da Silva Lopes

Deteção de Anomalias em Modelos de Publicidade Pay-Per-Click

Dissertação de Mestrado
Mestrado em Engenharia Informática

Trabalho realizado sob orientação de

Professor Doutor Paulo Jorge de Sousa Azevedo

Outubro de 2012

"Information is not knowledge."

Albert Einstein

Agradecimentos

Em primeiro lugar gostaria de agradecer ao Professor Doutor Paulo Azevedo pela orientação prestada ao longo de todo o período de investigação, bem como pelo apoio e disponibilidade demonstrada. Só com a sua intervenção foi possível alcançar os objetivos delineados no início deste projeto.

Gostaria ainda de agradecer a toda a minha família pelo carinho e incentivo, com especial destaque para os meus pais Serafim Lopes e Luísa Silva. O seu contributo foi fundamental, pois sem o seu esforço diário nunca teria ao meu dispor todas as condições necessárias à realização deste sonho. Não esquecendo um especial obrigado ao meu irmão Pedro Lopes e à Ana Filipa Duarte, pelos conselhos e sugestões apresentadas.

Por fim, gostaria ainda de demonstrar a minha gratidão para com a Adclip, pois for através desta que surgiu este projeto.

Abstract

Nowadays, online advertisement is one of the most effective and profitable marketing strategies. An example of strong growth in online advertisement is the Pay-Per-Click Advertising Model where all parties are benefited.

Due to the number of stakeholders and the amount of money involved, it is inevitable to find efficient methods to analyse the validity of clicks on online advertising, specifically in Pay-Per-Click.

The confidence of the advertiser is a crucial point to the success of this model. So it is necessary the distinction between the valid and the invalid clicks, made with the intention of generating charges, benefiting directly or indirectly with that action.

Therefore, a state of the art about fraud detection techniques in Pay-Per-Click will be presented, as well as the main techniques used to deceive this advertising model.

Other related matters were subject of study, such as the relevant data to collect for an accurate analysis of data flow at the servers.

It was performed a comparative analysis of different approaches of anomaly detection in order to identify the most suitable for the problem at hand. Using this subarea of Data Mining, very satisfactory results have been achieved, thus concluding that anomaly detection can give a major contribution to the resolution of Pay-Per-Click fraud.

Resumo

Os anúncios online são atualmente uma das estratégias de marketing mais rentáveis e eficientes. Um exemplo de forte crescimento nesta área é o modelo de publicidade *Pay-Per-Click*, onde todos os intervenientes são beneficiados.

Devido ao número de intervenientes e à quantidade de dinheiro envolvido, torna-se inevitável encontrar métodos eficientes para analisar a validade dos cliques efetuados em publicidade *online*, mais concretamente em sistemas *Pay-Per-Click*.

A confiança do anunciante é um fator crucial para o sucesso deste modelo. Assim, é necessário distinguir os cliques válidos dos inválidos, feitos com a intenção de gerar um débito, beneficiando direta ou indiretamente com essa ação.

Deste modo, será apresentado um estado da arte sobre técnicas de deteção de fraude em *Pay-Per-Click*, assim como as principais técnicas utilizadas para defraudar esse tipo de modelo.

Outros assuntos relacionados foram também objeto de estudo, tal como os dados necessários para uma análise precisa do fluxo de dados nos servidores.

Foi efetuado uma análise comparativa de diferentes abordagens de deteção de anomalias a fim de identificar quais as mais adequadas para o problema em questão. Com recurso a esta subárea de *Data Mining* foram alcançados resultados bastantes satisfatórios, concluindo-se assim que a deteção de anomalias pode dar um contributo fundamental para a resolução de fraude em *Pay-Per-Click*.

Conteúdo

Lista de Tabelas

Lista de Figuras

Capítulo 1

Introdução

A publicidade é uma forma de difundir publicamente uma ideia, produto, serviço ou o nome de uma empresa, a fim de atingir um público-alvo específico e com finalidade comercial, incentivando à execução de uma determinada ação, por exemplo, comprar um carro novo. Normalmente, o público é confrontado com os anúncios de três formas distintas: visual, geralmente usando imagens ou vídeos, auditiva ou na forma de texto, tipicamente encontrados em televisão, rádio e jornais, respetivamente. Outro meio de divulgação de publicidade é a Internet.

Hoje em dia, a publicidade *online* é uma das estratégias de *marketing* mais eficazes e rentáveis, tendendo a aumentar anualmente (ver Figura 1.1) [IAB and PwC, 2011].

Um exemplo de um forte crescimento em publicidade *online* é o método *Pay-per-Click*, onde todos os intervenientes são beneficiados. O anunciante divulga o seu negócio, a empresa promotora gera lucro a partir dos cliques feitos nos anúncios, e o utilizador-alvo é capaz de encontrar/comprar produtos talhados às suas necessidades. Alguns modelos de negócio assentes em *Pay-Per-Click* introduzem um novo interveniente, denominado angariador, como será explicado posteriormente.

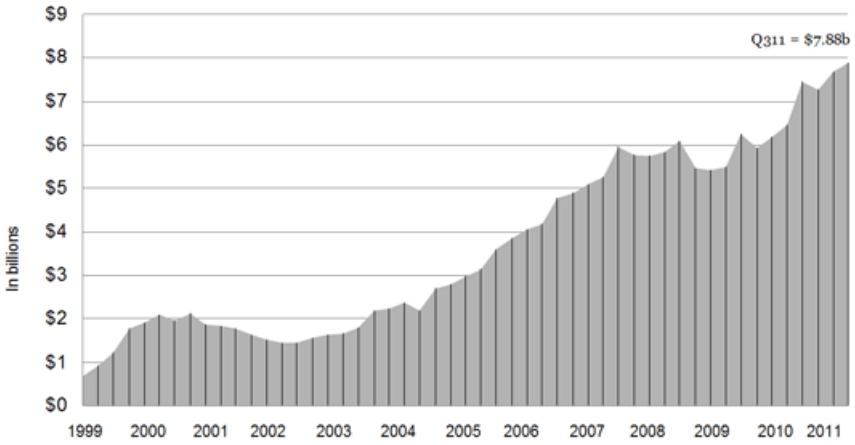


Figura 1.1: Receitas de Publicidade Online nos USA [IAB and PwC, 2011]

A confiança do anunciante é um fator crucial para o sucesso deste modelo. Devido ao volume de negócio envolvido e ao número de intervenientes, as empresas que usam este modelo vêm-se forçadas a procurar e implementar métodos eficientes para avaliar a autenticidade dos cliques efetuados, a fim de salvaguardar a credibilidade do negócio.

Uma das possíveis abordagens para detetar cliques fraudulentos neste cenário é a deteção de anomalias no fluxo de cliques. A deteção de anomalias é uma vasta subárea de *Data Mining* que poderá dar um contributo fundamental para a resolução deste problema [Chandola et al., 2009].

1.1 Motivação e Objetivos

Uma vez que este modelo de negócio está em forte crescimento, a concorrência nesta área tem aumentado consideravelmente. Por isso é essencial que as empresas e os grupos detentores deste tipo de modelo tentem diferenciar-se dos demais, ganhando assim mais utilizadores. Essa diferenciação pode ser alcançada mediante a instauração do sentimento de confiança no anunciante para com o serviço. Isto é,

onde ele sente que, investindo capital na divulgação do seu negócio, irá conseguir o retorno desejado.

Assim, pretende-se efetuar um estudo comparativo de diversas técnicas de Deteção de Anomalias, a fim de verificar a sua viabilidade no processo de deteção e invalidação cliques fraudulentos em modelos de publicidade *Pay-Per-Click*, evitando desta forma a cobrança indevida ao anunciante. De forma a alcançar uma solução que vá de encontro ao pretendido será necessário atingir uma série de objetivos intermédios. Numa primeira fase será necessária a compreensão do *modus operandi* do modelo de publicidade *Pay-Per-Click* , a fim de identificar quais os seus intervenientes e qual o seu interesse em defraudar o sistema.

Será também necessário efetuar um estudo sobre o problema da fraude nos sistemas *Pay-Per-Click*, bem como as principais técnicas usadas para defraudar este tipo de modelo.

Posteriormente será fundamental realizar um levantamento sobre qual o estado de arte relativamente à problemática de fraude em *Pay-Per-Click*, com ênfase em técnicas de deteção de anomalias.

Será ainda necessário um estudo sobre quais os dados necessários para uma análise precisa do fluxo de dados, bem como um tratamento da informação disponível a fim de obter um conjunto de informações coerentes a partir do qual se poderá extrair o conhecimento desejado. Este conhecimento poderá ser obtido recorrendo a um conjunto adequado de técnicas de *Data Mining* e a uma correta interpretação dos seus resultados.

Após a obtenção de um conjunto de dados devidamente tratado e de uma seleção de métodos de deteção de anomalias, será necessária uma comparação desta, a fim de identificar qual a melhor técnica a aplicar ao problema em questão.

1.2 Estrutura do Documento

A presente dissertação encontra-se subdividida em 6 capítulos, sendo que o primeiro se destina à introdução do problema e onde são apresentadas as suas motivações e objetivos.

No capítulo seguinte, Capítulo 2, encontram-se explanados os diversos conceitos associados à publicidade *online*, onde se apresenta o seu contexto histórico, os seus intervenientes e os métodos de cobrança aplicados neste modelo de negócio. Neste capítulo é ainda abordado o tema da fraude praticada em *Pay-Per-Click* bem como o caso de estudo.

Posteriormente é apresentado o estado da arte com foco na deteção de fraude nos sistemas assentes em *Pay-Per-Click*.

No Capítulo 4 é apresentado o tema da Deteção de Anomalias, iniciando-se com a definição e identificação dos diversos tipos de anomalias, bem como as principais abordagens existentes nesta área.

O Capítulo 5 expõe o processo efetuado para a obtenção de um conjunto de dados propício à extração de conhecimento, bem como a seleção dos métodos de deteção de anomalias a aplicar ao referido conjunto.

O resultado da aplicação dos métodos selecionados sobre o conjunto de dados criado, é apresentado no Capítulo 6.

Por fim, são apresentadas as conclusões obtidas através da análise ao trabalho desenvolvido. Ainda neste capítulo, são antecipadas futuras etapas deste projeto.

Capítulo 2

Publicidade *Online*

Este capítulo apresenta os principais marcos históricos da publicidade *online*, com foco na evolução do *Pay-Per-Click*. São também introduzidos os conceitos básicos utilizados neste tipo de negócio, bem como os principais métodos de cobrança aplicados, os seus intervenientes e os diversos tipos de fraude praticados. É apresentado o caso de estudo que deu origem a este projeto de dissertação e que introduz algumas particularidades relativamente ao modelo *Pay-Per-Click* base.

2.1 Evolução da Publicidade *Online*

A publicidade online deu o seu primeiro passo quando a HotWired introduziu um *banner* publicitário no seu site (ver Figura 2.1) , em outubro de 1994, com a finalidade de gerar receitas. A Yahoo! seguiu esta estratégia e fez um acordo com a Procter & Gamble, em 1996, cobrando apenas quando os utilizadores clicavam no *banner*, introduzindo assim o conceito de *Pay-Per-Click* [Hollis, 2005].

Em 1998, uma empresa chamada *GoTo*[1], mais tarde *Overture Services*, foi a primeira a introduzir o conceito de Pesquisa Patrocinada (PP), acrescentando

[1]Naquela época hospedado em http://goto.com

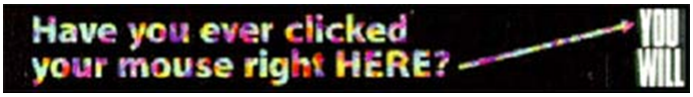


Figura 2.1: O primeiro *banner* publicitário [Li et al., 2011]

anúncios como parte dos resultados das pesquisas [Fain and Pedersen, 2006]. Assim, os anunciantes definiam quanto estariam dispostos a pagar para aparecerem na primeira página de resultados de uma dada pesquisa, conhecido como *Pay-For-Placement* (PFP). O maior cliente da *Overture*, a *Yahoo!*, adquiriu a empresa em 2003, começando assim o *Yahoo! Search Marketing*[2] em 2005.

No final de 1999, a empresa Google iniciou seu sistema de PP e introduziu o Google AdWords, um sistema PFP, em Outubro de 2000 com o anúncio na página principal, transmitindo a mensagem "Have a credit card and 5 minutes? Get your ad on Google today". O sistema AdWords usava um modelo de cobrança denominado Cost-per-Mille, cobrando aos anunciantes por cada mil impressões dos seus anúncios [Levy, 2011].

Em 2002, a Google mudou o modelo de negócios do Adwords, introduzindo o sistema de cobrança PPC, tal como a GoTo. Contudo, a Google percebeu que o modelo implementado pela GoTo era ineficiente. A GoTo ordenava os anúncios pela quantia de dinheiro cobrada por clique, o que significa que os anúncios com quantias mais elevadas apareciam sempre no topo, independentemente da sua popularidade, tornando os anúncios pouco atraentes para os utilizadores. Como consequência, a quantidade de cliques nos anúncios era reduzida, levando a um baixo fluxo monetário. Assim, a Google implementou um sistema de ordenação de anúncios baseado numa taxa de cliques por anúncio, medindo desta forma a relevância de cada anúncio para os utilizadores. Com essa implementação, se um anúncio com um menor pagamento por clique for clicado mais vezes, aparecerá com maior destaque na classificação final. Esta pequena mudança fez uma grande

[2] Ver http://searchmarketing.yahoo.com

diferença na receita do Google, revelando que os anúncios com pagamentos mais baixos obtiveram mais cliques do que os restantes.

2.2 Métodos de Cobrança

Desde os primórdios da publicidade *online* que a principal preocupação dos anunciantes era saber como medir a eficácia dos seus investimentos, e a dos editores era encontrar a melhor forma de cobrar. Neste modelo de negócio existem três principais métodos de cobrança que podem ser aplicados a um anunciante. Na literatura [Jakobsson and Ramzan, 2008; Tuzhilin, 2006] são referidos como:

- Custo por Clique (CPC) - onde o anunciante paga um montante específico sempre que um clique é registado num dos seus anúncios. Este é o método mais popular.
- Custo por Ação (CPA)- é aplicado sempre que uma ação é efetuada pelo utilizador (p.e. um pedido de contacto ou compra).
- Custo por Mille (CPM) - onde um dado montante era cobrado por cada milhar de impressões de um determinado anúncio.

2.3 Intervenientes

Normalmente, este modelo é composto por quatro tipos de utilizadores.

A **empresa promotora** é a responsável por angariar, armazenar e disponibilizar todas as informações relativas aos anúncios. A esta compete também a tarefa de gestão de todo o sistema, como por exemplo, a definição de qual o método de cobrança a aplicar (ver 2.2) ou a gestão do saldo dos anunciantes.

O **editor** é responsável por publicar anúncios no seu site. Este terá o interesse de destacar os anúncios, a fim de serem mais facilmente visíveis ao visitante,

aumentado assim a probabilidade deste clicar no anúncio, gerando deste modo maiores receitas.

Ao **anunciante** cabe a tarefa de criar o anúncio e atribuir um orçamento a ser gasto com ele. Os anúncios deverão ser visualmente atrativos, a fim de cativar a atenção do visitante.

Por último, o **visitante**, alvo dos anúncios.

2.4 Fraude em PPC

Click Fraud é uma atividade ilegal que ocorre em publicidade *online*, nomeadamente em sistemas *Pay-Per-Click*. Pode ser realizada por uma pessoa, grupo de pessoas ou um programa de computador com a intenção de gerar receitas ou prejudicar a concorrência [Feily et al., 2009].

Existem dois métodos principais de fraude neste sistema, vulgarmente denominados por *Hit Shaving* e *Hit Inflation*. *Hit Shaving* ocorre quando a empresa promotora decide ocultar uma determinada quantidade de cliques recebidos através de um dado editor, recolhendo assim todas as receitas geradas a partir da cobrança ao anunciante dos referidos cliques [Anupam et al., 1999]. Este cenário pode ocorrer se o editor não se precaver deste tipo de fraude. A figura 2.2 representa o funcionamento básico do protocolo *Hypertext Transfer Protocol* (HTTP), permitindo a ocorrência de *Hit Shaving*. Neste cenário, o utilizador U após receber a *Pagina1.htm* do editor E, efetua um clique, solicitando assim a *Pagina2.htm* à empresa promotora P. Neste processo, E não tem conhecimento da ocorrência do clique, ficando vulnerável a este tipo de fraude. Em [Reiter et al., 1998] é proposta uma solução para este problema, através da notificação do editor aquando do clique, recorrendo à tecnologia *JavaScript*. Desta forma, aquando da solicitação da

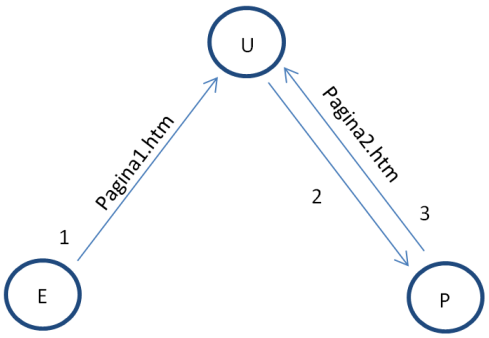


Figura 2.2: Processo vulnerável a *Hit Shaving*

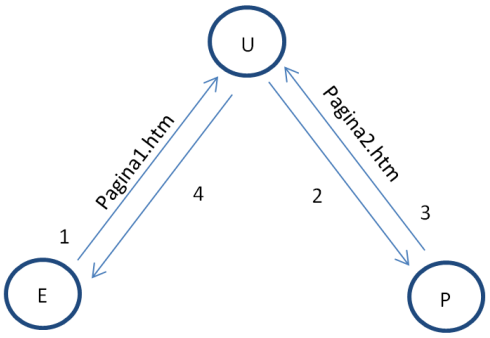


Figura 2.3: Processo imune a *Hit Shaving* [Reiter et al., 1998]

Pagina2.htm por U, é também enviada uma notificação a E (Figura 2.3).

O método de *Hit Inflation* consiste na prática de cliques fraudulentos [Metwally et al., 2007a]. Um clique fraudulento surge sempre que o ator efetue um clique num anúncio publicitário *online* sem ter qualquer interesse no anúncio ou no respetivo anunciante.

Os principais atores interessados na ocorrência de fraude são:

- A **empresa promotora**, pois é a principal beneficiada com a inflação de cliques;
- O **editor**, uma vez que recebe uma dada percentagem do valor de cada clique que tenha ocorrido no seu *site*, beneficiando assim da inflação de cliques;
- O **anunciante**, pois beneficia indiretamente com os cliques fraudulentos efetuados nos anúncios dos seus concorrentes, devido às cobranças feitas. Esses cliques diminuem o saldo do concorrente, removendo-o da lista de patrocinadores de determinada pesquisa;

A figura 2.4 ilustra os pontos mencionado acima.

Diferentes técnicas podem ser utilizadas pelo defraudador numa tentativa de ocultar ou disfarçar as suas ações. Uma vez que técnicas mais complexas reque-

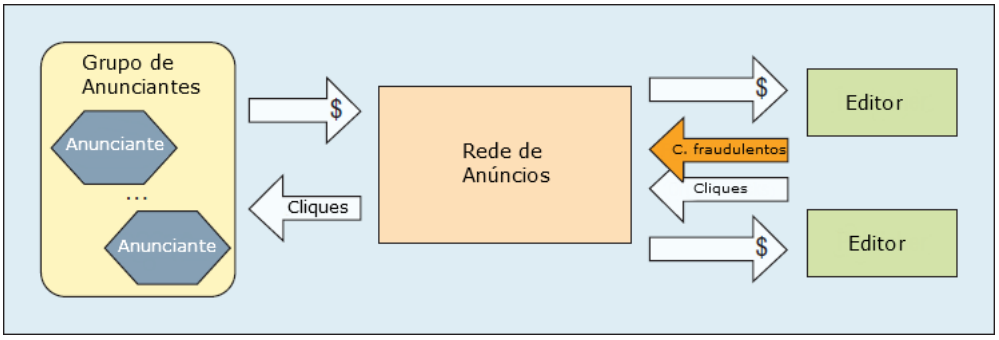


Figura 2.4: Funcionamento PPC [Li et al., 2011]

rem um maior investimento monetário e/ou temporal bem como um maior conhecimento tecnológico, a maior parte dos utilizadores recorre a estratégias mais simplistas [Metwally et al., 2007a]. São exemplo técnicas como a mudança de endereço de IP, limpeza de *cookies*, utilização de *proxies* ou diferentes navegadores.

2.5 Caso de Estudo

AdClip[3] é uma rede *online* de anúncios classificados que teve o seu início no final de 2008. O seu modelo de negócio é assente em *Pay-Per-Click*, mas com algumas particularidades. O sistema é baseado no conceito de *clip*. Um *clip* é uma interface *Web* desenhada para permitir ao visitante navegar através dos anúncios providenciados pela empresa, subdivididos por várias categorias, como por exemplo, bens imóveis ou viaturas.

Neste modelo são contemplados cinco tipos de utilizadores: a **empresa promotora**, neste caso a Adclip; o **editor**, como por exemplo o jornal Público[4] e Correio do Minho[5]; o **anunciante**; e o **visitante**, alvo dos anúncios. O quinto interveniente que o modelo considera é o **angariador** do conteúdo, ou seja, o editor através do qual o anúncio foi inserido na rede.

[3] Ver http://www.adclip.com
[4] Ver http://static.publico.pt/Classificados/
[5] Ver http://www.correiodominho.com/classificados.php

Assim, aquando de um clique efetuado pelo visitante num dado anúncio, através de um dado editor, o montante pago pelo anunciante será dividido pelos intervenientes, isto é, pela empresa promotora, o editor e ainda pelo angariador. Desta forma, deve-se também considerar o angariador como um elemento interessado na fraude deste sistema, adicionalmente aos já mencionados na secção 2.4.

2.6 Síntese

O estudo desenvolvido nesta fase revelou um grande crescimento e investimento na área de publicidade *online* e a consequente importância da deteção de fraude nestes sistemas dada a necessidade de manter estes modelos atrativos a novos investidores.

Foi também importante para uma maior sensibilização para os principais métodos de fraude praticados bem como quais os interessados na sua prática.

Durante o processo de investigação foram identificadas algumas particularidades de diferenciação entre o caso de estudo e os modelos base assentes em *Pay-Per-Click*. Esta fase permitiu uma familiarização com diversos conceitos que serão certamente úteis na idealização e conceção de uma solução, nas subsequentes fases deste projeto.

Capítulo 3

Estado da Arte

A Internet tornou-se numa das formas mais poderosas de comunicação. Está disponível em variados dispositivos, como por exemplo, computadores portáteis, telemóveis ou *tablets*. Atualmente, os seus utilizadores podem ter acesso a notícias, redes sociais, cultura e educação. A Internet pode ainda proporcionar oportunidades de negócios, anúncios, leilões online, etc. No entanto, esta poderosa ferramenta também tem as suas desvantagens. Problemas como *phishing*, *malware* ou roubos de identidade são ameaças constantes para empresas ou utilizadores descuidados [Jakobsson and Ramzan, 2008]. Em [Jakobsson and Ramzan, 2008] são apresentadas algumas ameaças à segurança dos utilizadores, como *rootkits*, *bot networks*, *spyware* e *adware*.

Uma das maiores ameaças à segurança prende-se com as intrusões em dispositivos remotos. Kemmerer e Vigna ([Kemmerer and Vigna, 2002]) apresentam um breve resumo sobre o problema da deteção de intrusões. Os autores mencionam duas técnicas de deteção, nomeadamente deteção de anomalias e de uso indevido. Neste contexto, deteção de uso indevido é o processo que visa a tentativa de identificação de ataques pela rede através da comparação da atividade corrente com a expectável de um intruso. A maioria das abordagens para a de-

teção de uso indevido envolvem o uso de sistemas baseados em regras a fim de identificar indícios de ataques conhecidos [Cannady, 1998]. Contudo, estas técnicas baseadas em deteção de uso indevido tendem a falhar aquando de ataques que difiram dos padrões previamente identificados. Uma técnica capaz de colmatar as falhas anteriormente referidas e dotar os sistemas para a componente variável dos ataques é a técnica de deteção de anomalias. Esta foi proposta para os sistemas de deteção de intrusões por Dorothy Denning em 1987 [Denning, 1987]. Citando [He et al., 2010], "A deteção de anomalias, como um importante complemento às técnicas de deteção de uso indevido, tem a capacidade de descobrir e prevenir os conhecidos ataques "dia-zero"". Deteção de anomalias é uma grande sub-área de *Data Mining* e pode ser aplicada a muitas áreas, tais como sistemas de deteção de intrusão, de fraude em telecomunicações [Ferreira et al., 2006], fraude em cartões de crédito [Ghosh and Reilly, 1994; Dorronsoro et al., 1997], ou processamento de imagem, como por exemplo vídeos de vigilância [Diehl and Hampshire, 2002]. Em [Chandola et al., 2009] é apresentado um estudo alargado sobre diversas técnicas de deteção de anomalias bem como as suas áreas de aplicação.

Em [Jakobsson and Ramzan, 2008], os autores expõem também o problema de fraude em modelos PPC, conhecido como *badvertisement*, onde o criminoso, neste caso o editor, força o visitante a efetuar cliques automaticamente nos anúncios presentes no seu site, tipicamente aquando do carregamento inicial do site do editor, gerando desta forma receitas para o *badvertiser* a custo dos anunciantes. Dada a invisibilidade deste tipo de ataques, este método pode perdurar durante bastante tempo.

Em [Jansen, 2007], o autor apresenta um análise sobre o problema da fraude em PPC, propondo quatro possíveis soluções para solucionar este problema. O autor defende a prática de uma monitorização mais agressiva, aumentando assim o esforço necessário para defraudar o modelo, levando a uma redução do número

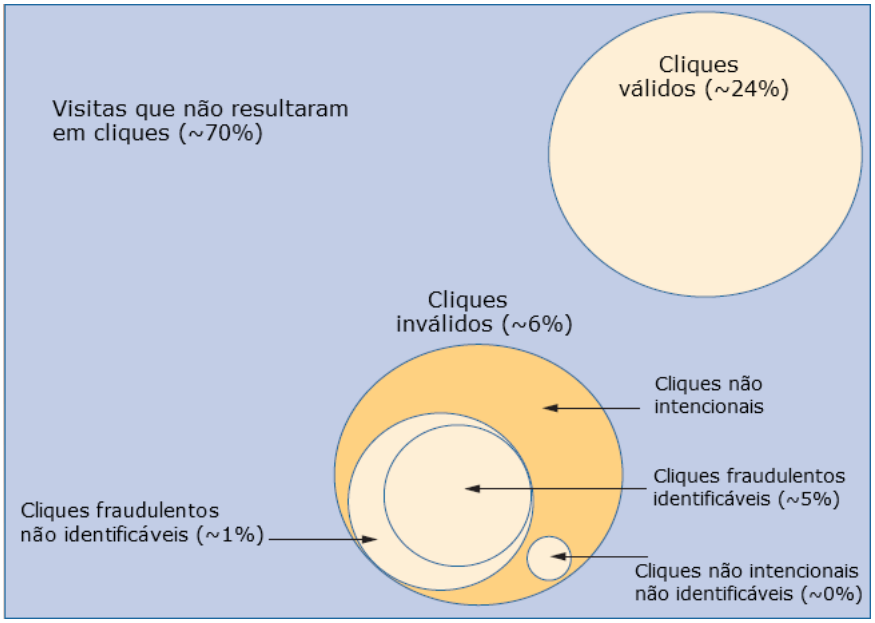


Figura 3.1: Resultados dos cliques efetuados em pesquisas patrocinadas [Jansen, 2007]

de perpetradores. Propõe também a introdução de filtros automatizados mais eficientes, recorrendo a técnicas de *Data Mining*. Uma outra proposta do autor prende-se com a migração dos sistemas baseados em PPC, para sistemas *Pay-Per-Action*. Por último, é proposta uma relação entre editores e empresas promotoras baseadas no sentimento de confiança, através da celebração de contratos. Segundo o autor, cerca de 30 por cento de todas as visitas aos motores de busca resultam em um ou mais cliques nos resultados patrocinados, sendo na sua maioria considerados cliques válidos (Figura 3.1). Cerca de 5 por cento dos cliques efetuados são identificados como sendo fraudulentos.

Em [Anupam et al., 1999], foi apresentado um ataque de *hit inflation* em modelos de PPC, virtualmente impossível de ser detetado de forma conclusiva. Os autores defendem que se amplamente praticado, este tipo de ataques poderia causar a transição de modelos assentes em *Pay-Per-Click* para sistemas onde apenas seria cobrado ao anunciante nos casos em que o visitante efetuasse uma compra (*Pay-Per-Sale*) ou aderisse a determinadas atividades no site anunciado (*Pay-Per-*

Lead).

O problema de *hit inflation* por coligações entre defraudadores foi estudada em [Metwally et al., 2007a,b]. Os autores propõem um algoritmo de procura de comportamentos similares, denominado *Similarity-Seeker*, capaz de identificar ataques praticados por pares de defraudadores, bem como uma solução para coligações de tamanhos arbitrários.

Em 2007, foi proposta uma nova técnica para o combate ao problema da fraude em modelos PPC, denominado *Premium Clicks* [Juels et al., 2007]. Consideram uma abordagem que apenas aceita cliques legítimos, validando-os através da autenticação anónima dos visitantes, transparente aos utilizadores e que não necessita de mudanças do lado do cliente.

Em [Zhang and Guan, 2008], Zhang e Guan propuseram uma técnica que deteta cliques duplicados através de modelos baseados em janelas, como por exemplo, janelas saltitantes e janelas deslizantes.

Uma outra solução foi proposta em [Kantardzic et al., 2008], que analisa as atividades detalhadas de cada utilizador, e se baseia na recolha de dados de diferentes fontes, argumentando que uma maior quantidade de informação sobre cada clique permite uma melhor avaliação da qualidade do tráfego de cliques. Os autores recorrem a *multi-source data fusion* (MSDF) para juntar os dados provenientes do cliente com os recolhidos do lado do servidor.

Um modelo colaborativo de deteção e prevenção de fraude em PPC em tempo real baseado em multi-sensores, visto como um problema de fusão de evidências, foi proposto em [Kantardzic et al., 2010].

Uma outra técnica para a deteção de fraude em PPC foi proposta em [Haddadi, 2010], através da introdução de anúncios falsos misturados com os reais. Estes anúncios podem ser direcionados e com texto irrelevante, ou não direcionados e com texto altamente relevante, tornando-os não atrativos a utilizadores reais.

Desta forma, seria possível distinguir visitantes legítimos de *Botnets*.

Em [Antoniou et al., 2011] foi proposto um sistema de deteção em tempo real de fraude em PPC utilizando estruturas de dados avançadas baseadas em árvores *Splay* e explorando as suas vantagens relativamente ao espaço e tempo necessário.

Recentemente, em [Costa et al., 2012] é defendido o uso de CAPTCHAs (*Completely Automated Public Turing test to tell Computers and Humans Apart*) clicáveis como medida de defesa contra ferramentas automáticas, capazes de efetuar cliques nos anúncios, simulando assim o comportamento humano.

Capítulo 4

Deteção de Anomalias

4.1 Definição

Os algoritmos de deteção de anomalias têm como propósito encontrar padrões em dados que não se adequam ao comportamento esperado ou dito normal. Esses padrões desviantes são muitas vezes denominados como anomalias ou *outliers* embora seja possível encontrar referências a observações discordantes, exceções, aberrações, surpresas, peculiaridades, ou contaminantes, dependendo do contexto em que estão inseridos [Chandola et al., 2009].

[Hawkins, 1980] define *outliers* da seguinte forma:

> *Um outlier é uma observação que se desvia tanto das restantes observações que levanta suspeitas de ter sido gerada por um mecanismo diferente.*

Estas anomalias podem surgir acidentalmente ou de forma intencional. Situações como a leitura de valores invulgares em sensores ou tentativas de fraude são exemplos que podem originar dados anómalos num determinado contexto.

Contudo, é importante distinguir deteção de anomalias de técnicas de identificação de ruído nos dados. Muitos algoritmos tentam eliminar ou minimizar a

influência das anomalias. Contudo, esta abordagem pode levar a uma perda de informação essencial para o problema. Como referido em [Han, 2005] "o ruído de uma pessoa pode ser o sinal de outra". O ruído pode ser definido como informação não desejada, sem interesse para o problema, e que tende a ser eliminada ou ignorada aquando da análise. Por outro lado, uma anomalia tende a ser considerada informação relevante e, em certos contextos, crítica.

4.2 Tipos de Anomalias

4.2.1 Anomalias pontuais

Este é o tipo mais comum e mais estudado. Se um dado objeto é anómalo relativamente a todo o *dataset*, então é denominado de anomalia pontual.

A título de exemplo, considere-se a quantidade diária de pedidos recebidos por um dado servidor *Web*. Se num dado dia for registada uma anormal quantidade de pedidos comparativamente com o que é considerado normal, este registo será considerado uma anomalia pontual.

4.2.2 Anomalias Contextuais

Se um dado objeto for anómalo relativamente ao seu contexto e não relativamente a todo o universo de objetos, este é denominado uma anomalia contextual.

Os problemas que atentam a este tipo de anomalias requerem a presença, no seu conjunto de dados, de atributos capazes de introduzir a noção de contexto, como por exemplo tempo, posição geográfica ou temperatura. Desta forma, os métodos de deteção tentam identificar anomalias recorrendo a atributos denominados comportamentais tendo em consideração os de contexto.

Retomando o exemplo anterior, considere-se que este servidor aloja um sítio na

Web dedicado à venda de brinquedos e que é comum um aumento do número de pedidos registados na altura do natal. Se num determinado dia do mês de março for registado uma quantidade de pedidos semelhante ao verificado no natal, embora não seja um valor nunca antes visto, este é considerado uma anomalia contextual, pois naquela data e comparativamente a anos anteriores, este valor apresenta um desvio do que é considerado normal.

4.2.3 Anomalias Coletivas

Se um conjunto de objetos, relacionados entre si, for anómalo relativamente ao restante conjunto de dados, estes podem ser classificados como anomalias coletivas.

O estudo deste tipo de anomalias é mais comum em contextos que incluem dados sequenciais, espaciais ou baseados em grafos.

Considerando ainda o exemplo apresentado anteriormente. Atente-se a um conjunto P que contém todos os possíveis pedidos HTTP que o servidor *Web* pode receber por parte dos visitantes, onde $\{x,y,z,w\} \in P$. Assume-se que qualquer um dos possíveis pedidos é perfeitamente comum e inofensivo. Contudo, a ocorrência da sequência y,x,w,z pode indiciar uma tentativa de intrusão no servidor por parte de um utilizador, constituindo assim uma anomalia coletiva.

4.3 Tipos de Supervisão

Dependendo da existência ou não de dados devidamente catalogados, informando se estes são considerados anómalos ou não, diferentes métodos de deteção de anomalias podem ser aplicados. A obtenção de um conjunto de dados corretamente rotulado e capaz de representar todo o tipo de possíveis padrões é, em certos casos, impossível ou demasiado dispendioso.

Considere-se, a título de exemplo, o processo de criação de um conjunto de

dados relativo à qualidade da carne vendida numa grande superfície comercial. Assumindo que a carne testada não poderá ser vendida e que todo este processo é feito necessitando de recursos humanos especializados é possível concluir que se trata de um procedimento dispendioso.

Sendo que, por definição, anomalias são situações que ocorrem excecionalmente, a obtenção deste tipo de situações é, em determinados contextos, de muito difícil obtenção. Se se considerar ainda que, para determinados contextos, surgem constantemente novos tipos de anomalias, como por exemplo, técnicas de intrusão em dispositivos remotos, é possível concluir que o processo de criação de um modelo capaz de identificar todas as possíveis situações ou comportamento se torna inviável.

Devido a estas questões, existem métodos de deteção de anomalias capazes de operar em modo *supervisionado*, *semi-supervisionado* ou *não supervisionado*.

Nos métodos *supervisionados*, é assumida a presença de dados catalogados de ambos os tipos, isto é, normais e anómalos.

Em modo *semi-supervisionado*, o *dataset* é composto apenas por um tipo de dados, tipicamente dados considerados normais, pois na maioria dos casos são de mais fácil obtenção.

Por último, os algoritmos capazes de operar em modo *não supervisionado* não necessitam de nenhum tipo de catalogação dos dados presentes no *dataset*. No entanto, este tipo de método assume que os dados normais estão presentes em muito número do que os anormais.

4.4 Aprendizagem *Online* ou *Offline*

Em [Laxhammar, 2011], as diversas técnicas de deteção de anomalias são classificadas segundo o seu modo de aprendizagem. O autor identifica dois tipos de

aprendizagem, nomeadamente *online* e *offline*. A maioria das técnicas efetua uma aprendizagem em modo *offline*, ou seja, os modelos são aprendidos com base em conjuntos de treino estáticos, e posteriormente utilizados para analisar novas instâncias. Para a criação de um modelo robusto é, tipicamente, necessário recorrer a um conjunto de treino relativamente grande.

No entanto, esta informação pode não estar disponível *a priori* ou, em determinados contextos, a informação a recolher pode estar em constante mudança e evolução. Por estes motivos, surge a necessidade de uma aprendizagem *online*. Neste caso, o modelo vai evoluindo e aprimorando com base nos novos dados recolhidos.

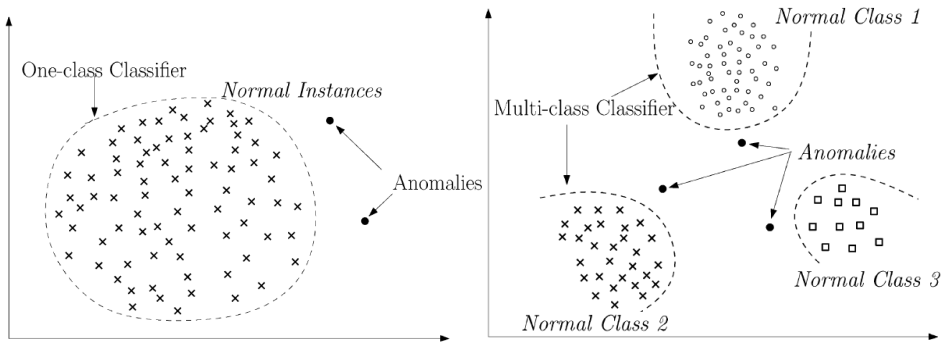
4.5 Tipos de Algoritmos

As técnicas de deteção de anomalias podem ser agrupados com base na sua abordagem. De seguida são apresentadas as principais abordagens, bem como os algoritmos mais representativos e apropriados à tarefa proposta.

4.5.1 Baseados em Classificadores

Os métodos baseados em classificadores operam segundo um processo composto por duas fases distintas. Numa primeira fase, os algoritmos de classificação constroem um modelo (classificador). Nesta etapa do processo, denominada fase de treino ou de aprendizagem, o modelo é construido a partir de um conjunto de dados de treino, onde cada instância tem associada a si um conjunto de atributos e uma etiqueta identificativa da sua classe.

Na segunda fase do processo, o modelo criado é submetido a um conjunto de teste para determinar a sua capacidade de classificação. Este conjunto deve ser diferente do usado na fase anterior, pois caso contrário, a avaliação seria tendenciosa



(a) One-Class (b) Multi-Class

Figura 4.1: Métodos de classificação [Chandola et al., 2009]

uma vez que estes modelos podem sofrer de um problema denominado *overfitting*. Se o modelo criado na primeira fase for demasiado rigoroso e considerar todas as hipóteses presentes no conjunto de treino, este sofrerá de *overfitting* ou sobreajustamento.

De forma a solucionar este problema, os algoritmos de classificação recorrem a técnicas de *prunning* (poda) de modo a criar modelos capazes de classificar instâncias relativamente semelhantes às utilizadas na fase de aprendizagem.

Assim, após criado o modelo, este pode ser submetido a novos conjuntos de forma a classificar as novas instâncias como normais ou anormais.

No entanto, devido à necessidade da existência da identificação da classe a que cada tuplo pertence, na fase de treino, os métodos de classificação são apenas capazes de operar em modo supervisionado ou semi-supervisionado. Do mesmo modo, estas técnicas podem ser divididas em classificação *one-class* ou *multi-class*.

A Figura 4.1 ilustra os dois tipos de classificação referidos. No primeiro caso (a), algoritmos como *One-Class Support Vector Machine* [Schölkopf et al., 2001] ou *one-class Kernel Fisher Discriminants* [Roth, 2004], todas as instâncias presentes no conjunto de dados possuem a mesma identificação da classe. Deste modo, o algoritmo analisa as instâncias dessa mesma classe e cria uma fronteira em torno

desse conjunto. Assim, aquando do teste de novos objetos para com o modelo criado, estes serão considerados anómalos se não estiverem dentro dessa fronteira.

Seguindo o mesmo raciocínio, os algoritmos de deteção de anomalias baseados em classificação *multi-class* [De Stefano et al., 2000; Barbará et al., 2001] criam um modelo através da análise de um conjunto de dados composto por instâncias catalogadas com mais do que uma classe.

Os algoritmos de deteção de anomalias baseados em classificadores sofrem de duas limitações graves. Como referido anteriormente, estes métodos requerem que cada instância seja acompanhada de um identificador da sua classe, o que nem sempre e possível, como referido em 4.3. Uma outra contrapartida desta abordagem é o seu tipo de resultado. Os algoritmos de classificação atribuem a cada instância testada uma identificação, indicando a que classe esta pertence. Deste modo, não é permitido ao analista verificar o grau de anormalidade de cada instância.

4.5.2 Métodos Estatísticos

No ceio da comunidade estatística, o conceito de deteção de anomalias ou *outliers* é conhecido e estudado há bastante tempo. Tipicamente, este tipo de abordagem assenta em modelos probabilísticos ou de distribuição, como por exemplo distribuições normais ou Poisson, sendo que os objetos que se desviem desses modelos, recorrendo a testes de discordância, são considerados como anomalias [Hawkins, 1980; Barnett and Lewis, 1994].

No entanto, este tipo de abordagem, denominadas paramétricas, requerem conhecimento prévio do *dataset* e respetiva distribuição, como por exemplo, a sua média e desvio padrão. Isto limita a sua aplicabilidade em determinados contextos, uma vez que em certos problemas a distribuição dos dados não é conhecida.

Uma outra limitação deste tipo de abordagens é que estas, na sua maioria, são

apenas aplicáveis a conjuntos de dados univariados. Esta limitação impossibilita a sua aplicação em casos mais complexos que resultam em *dataset* multidimensionais.

Existem métodos não-paramétricos capazes de operar sobre conjuntos de dados multivariados, como por exemplo, técnicas baseadas em histogramas [Kruegel and Vigna, 2003; Endler, 1998; Goldstein and Dengel, 2012].

Neste tipo de abordagem, o grau de anomalia de uma dada instância é obtido através da soma das pontuações obtidas por cada atributo.

No entanto, embora seja simples a sua implementação, estas revelam-se incapazes de detetar relações entre os diferentes atributos [Chandola et al., 2009]. Esta incapacidade leva a que, numa dada instância, um conjunto de atributos que possua uma combinação rara, mas que individualmente apresentam valores comuns, seja considerada normal.

4.5.3 Baseados em *Clustering*

Os algoritmos baseados em *clustering* procuram dividir um dado conjunto de dados em diferentes grupos, denominados como classes ou *clusters*, com base na similaridade entre os diversos objetos presentes no *dataset*. Esta similaridade é medida com base no valor dos atributos constituintes de cada objeto [Han, 2005].

Contrariamente aos métodos de classificação, os algoritmos de *clustering* tendem a operar principalmente em modo não supervisionado, ou seja, sem que haja necessidade de a cada instância esteja associada uma classe.

Segundo [Chandola et al., 2009], estes algoritmos de *clustering* podem ser subdivididos em três tipos distintos:

1. Algoritmos que assumem que as instâncias normais devem pertencer a um *cluster*, enquanto que as anómalas não;

2. Algoritmos que defendem que os objetos normais se situam próximos do

centróide de um dado *cluster*, contrariamente aos anómalos;

3. Algoritmos que consideram que os dados normais pertencem a *clusters* densos e de grande dimensão, enquanto que as anomalias recaem em *clusters* pequenos ou esparsos.

O primeiro tipo de técnicas consideram que as instâncias anómalas não devem pertencer a nenhum *cluster*. O autor apresenta alguns algoritmos representativos deste tipo, como por exemplo, DBSCAN [Ester et al., 1996] ou ROCK [Guha et al., 1999]. No entanto, sendo a descoberta de *clusters* o principal objetivo destes algoritmos, estes não são especializados na deteção de anomalias.

O segundo tipo segue uma metodologia constituída por duas fases. Numa primeira etapa, é aplicado um algoritmo típico de *clustering* a fim de obter informações relativas aos *clusters* e respetivos centróides. Após obtidas estas informações, na segunda fase é calculada a distância a que cada instância se situa relativamente ao seu centróide mais próximo. Em [Barbará et al., 2003] é apresentada uma aplicação deste tipo de abordagem, aplicada à problemática da deteção de intrusões.

Contudo, este tipo de técnica falha se as anomalias presentes no conjunto de dados formar um *cluster*. Além disso, sendo esta uma abordagem global (ver 4.5.4) as anomalias locais não serão detetadas.

Assim, surge um terceiro tipo de abordagem, que baseia as suas decisões com base no tamanho e/ou densidade de cada *cluster*. Estes métodos após a análise do conjunto de dados e respetiva criação dos *clusters*, avaliam as características de cada um e comparam-no com os restantes. Se uma dada instância pertencer a um grupo de pequenas dimensões a sua probabilidade de ser considerada uma anomalia será maior comparativamente àquelas pertencentes a grandes grupos.

De seguida é apresentada uma abordagem proposta em [He et al., 2003] que segue a definição apresentada no terceiro tipo de métodos.

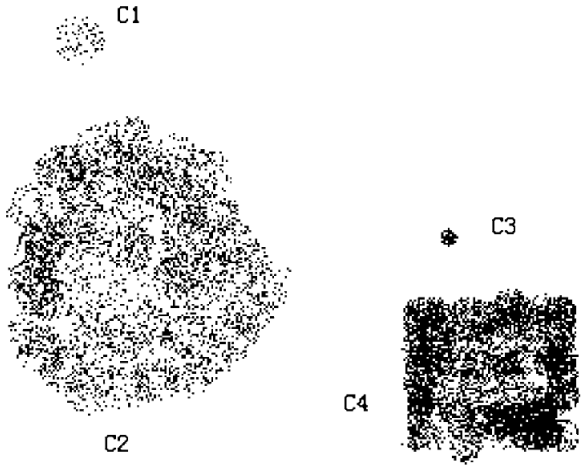


Figura 4.2: CBLOF conjunto de dados bidimensional [He et al., 2003]

Cluster-Based Local Outlier Factor

Em [He et al., 2003] é apresentado uma nova definição de anomalia, denominada *cluster-based local outlier*. Do mesmo modo, é introduzida uma nova medida para detetar este tipo de anomalia, intitulada *Cluster-Based Local Outlier Factor* (CBLOF).

Para uma melhor compreensão deste novo tipo de anomalias considere-se a Figura 4.2. Nesta é possível visualizar um conjunto de dados composto por quatro *clusters*, C_1, C_2, C_3 e C_4. Os autores defendem que os elementos contidos em C_1 e C_3 devem ser considerados anómalos uma vez que não estão contidos em nenhum dos grandes *clusters*, isto é, em C_2 ou C_4.

Sendo este método baseado na abordagem proposta em [Breunig et al., 2000], os autores consideram que as anomalias baseadas em *clusters* devem ser locais relativamente a um outro *cluster*, como por exemplo, as instâncias de C_1 serem anomalias locais relativamente a C_2.

Seja C um conjunto de *clusters* obtidos após a execução de um dado algoritmo de *Clustering* sobre um *dataset* D, isto é, $C = \{C_1,...,C_k\}$ onde $C_i \cap C_j = \emptyset$ e $C_1 \cup ... \cup C_k = D$.

Considere-se a seguinte cardinalidade: $|C_1| \geq |C_2| \geq ... \geq |C_k|$. Dados dois parâmetros numéricos α e β, b é denominado como a fronteira entre os grandes e os pequenos *clusters* se uma das seguintes condições se verificar:

$$(|C_1| + ... + |C_b|) \geq |D| * \alpha \tag{4.1}$$

$$|C_b|/|C_{b+1} \geq \beta \tag{4.2}$$

Assim, o conjunto dos grandes *clusters* é definido por $LC = \{C_i | i \leq b\}$. Analogamente, o conjunto composto pelos menores *clusters* é dado pela expressão $SC = \{C_j | j \geq b\}$.

Se se definir $\alpha = 90\%$, então serão denominados como *grandes clusters* aqueles que contenham 90% das instâncias presentes num conjunto de dados, assumindo assim que a grande maioria dos objetos são normais.

A segunda cláusula indica que deverão existir diferenças de cardinalidade significativas entre os conjuntos denominados grandes comparativamente aos pequenos. Se $\beta = 2$, então qualquer grande *cluster* será, pelo menos, duas vezes maior do que qualquer elemento de SC.

Categorizados os *clusters*, através da sua dimensão, é possível calcular o CBLOF para cada ponto $p \in D$. Este valor é obtido recorrendo à seguinte expressão:

$$CBLOF(p) = \begin{cases} |C_i| * min(d(p,C_j)) & \text{se } p \in C_i, C_i \in SC \\ & \text{e } C_j \in LC \text{ para } j{=}1 \text{ até } b \\ |C_i| * (d(p,C_i)) & \text{se } p \in C_i \text{ e } C_i \in LC \end{cases} \tag{4.3}$$

Assim, o grau de anomalia de uma dada instância será calculado através do

tamanho do *cluster* C_i a que pertence e distância ao maior e mais próximo *cluster* C_j, no caso de C_i ser de pequenas dimensões. Se a instância pertencer a um grande conjunto, o seu grau será calculado com base no tamanho e distância ao centróide desse mesmo conjunto.

4.5.4 Baseados em *Nearest-Neighbor*

Os métodos de deteção de anomalias baseados em *Nearest-Neighbor* podem ser agrupados em duas abordagens distintas. Estas podem ser *locais* ou *globais*:

- As abordagens *globais* medem o grau de normalidade de um dado objeto relativamente a todo o *dataset*. Em [Knorr and Ng, 1998] e [Knorr and Ng, 1999] estes são definidos da seguinte forma. Um objeto O num *dataset* T é um DB(p,D)-*outlier* se pelo menos uma fração p de objetos em T estiver a uma distância superior a D de O.

- Relativamente às abordagens *locais*, estas baseiam a sua medição através do cálculo da densidade da vizinhança de cada instância face à dos seus vizinhos, ou seja, no seu grau de isolamento comparativamente aos seus k vizinhos mais próximos [Breunig et al., 2000].

As abordagens baseadas em distância possuem uma grande vantagens comparativamente às abordagens estatísticas. Estas não necessitam de efetuar suposições relativamente à distribuição do *dataset*. Além disso, os métodos baseados em *Nearest-Neighbor* são escaláveis, uma vez que a análise de cada elemento está apenas dependente dos seus objetos vizinhos. Como consequência, este tipo de abordagem é apropriada para a descoberta de anomalias em grandes conjuntos de dados.

Seguindo a definição de *outlier* proposta por [Hawkins, 1980], os autores de [Knorr and Ng, 1998] propuseram um método baseado na distância, denominado

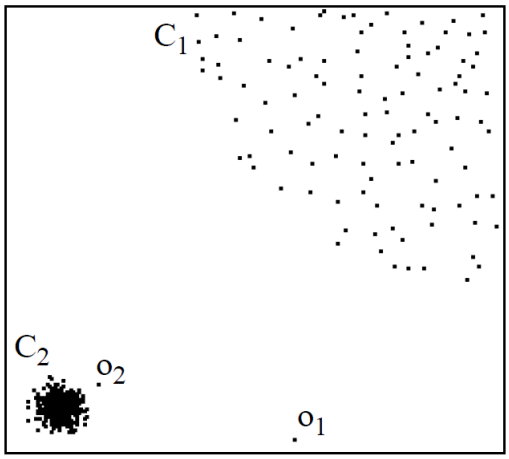


Figura 4.3: Conjunto de dados de duas dimensões [Breunig et al., 2000]

DB(p,D)-*outlier*. Abordagem similar foi proposta em [Ramaswamy et al., 2000], onde os autores calculam o grau de anomalia de um dado objeto com base na sua distância ao seu k-ésimo vizinho mais próximo. Estas duas abordagens, apesar de simples, possuem uma desvantagem considerável, uma vez que são incapazes de tratar corretamente conjuntos de dados compostos subconjuntos de diferentes padrões e/ou densidades.

Em [Breunig et al., 2000] é apresentado um exemplo (Figura 4.3) capaz de ilustrar as vantagens das abordagens locais relativamente às globais. O conjunto possui 502 objetos, dos quais 400 se inserem no *cluster* C_1 e 100 no *cluster* C_2, e dois objetos o_1 e o_2. Analisando os *clusters*, é possível verificar que C_1 possui maior densidade comparativamente a C_2. Por definição, os objetos o_1 e o_2 devem ser considerados como anomalias, contrariamente aos *clusters* C_1 e C_2. Se se considerar que todos os objetos q em C_1 possuírem uma distância relativa aos seus vizinhos mais próximos superior à distância entre o_2 e C_2, isto é, ao seu vizinho mais próximo sendo que este pertence a C_2, então, se for aplicado um método global baseado na distância, apenas o_1 será considerado como anomalia pois satisfaz a condição DB(p,D)-*outlier*. No entanto, não existem valores apropriados para os

parâmetros p e D de modo que o_2 seja considerado DB(p,D)-*outlier* sem que todos os objetos em C_1 o sejam também.

De seguida será apresentado um conjunto de algoritmos representativos destas duas abordagens descritas.

Local Outlier Factor

Método proposto em [Breunig et al., 2000], que atribui um dado grau de anormalidade a cada objeto presente no conjunto de dados. Este grau de desvio é denominado pelos autores como *Local Outlier Factor* (LOF).

Sejam o, q e p objetos de um dado *dataset* D e que a distância entre os objetos p e q é denotada por $d(p,q)$.

Seja k um inteiro positivo, k-*distance*(p) define a distância $d(p,o)$ entre p e $o \in D$ sendo que:

- pelo menos k objetos $o' \in D \setminus \{p\}$, onde $d(p,o') \leq d(p,o)$
- pelo menos $(k-1)$ objetos $o' \in D \setminus \{p\}$, onde $d(p,o') < d(p,o)$

O conjunto de objetos cuja distância a p seja igual ou inferior a k-*distance*(p) é definido por:

$$N_k(p) = \{q \in D \setminus \{p\} \mid d(p,q) \leq k\text{-}distance(p)\} \tag{4.4}$$

Estes objetos são então denominados com *k-vizinhos mais próximos* de p.

A distância de alcance (*reachability distance*) de um objeto p relativamente a um objeto o é definida como:

$$reach\text{-}dist_k(p,o) = max\{k\text{-}distance(o), d(p,o)\} \tag{4.5}$$

Para uma melhor compreensão, considere-se o exemplo da Figura 4.4. Aquando

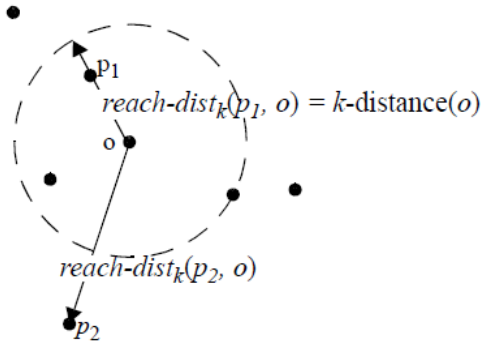


Figura 4.4: Exemplo do cálculo da distância de alcance [Breunig et al., 2000]

do cálculo da distância de alcance entre o objeto o e *p1* e *p2*, a primeira será dada pela sua *k-distance(p1)* enquanto que a segunda será a sua distância real, usando $d(o,p2)$. Isto permitirá que todos os objetos contidos em $N_k(o)$ vejam os seus valores homogeneizados.

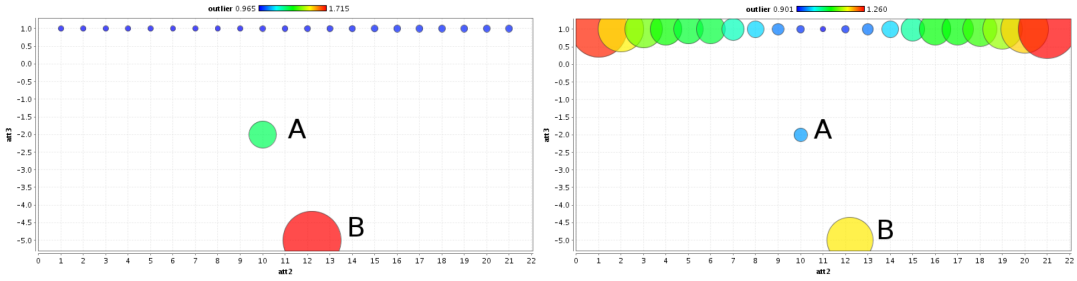
A densidade de alcance local (*local reachability density*) de p é definida como:

$$ldr_k(p) = \left(\frac{\sum_{o \in N_k(p)} reach\text{-}dist_k(p,o)}{|N_k(p)|} \right)^{-1} \tag{4.6}$$

Esta é definida pelo inverso da distância de alcance média. Por fim, o cálculo do *Local Outlier Factor* de p é dado pela seguinte expressão:

$$LOF_k(p) = \frac{\sum_{o \in N_k(p)} \frac{ldr_k(o)}{ldr_k(p)}}{|N_k(p)|} \tag{4.7}$$

Como é possível verificar pela fórmula, o valor de LOF do objeto p será tanto maior, quanto mais baixa for a sua densidade relativamente aos seus k-vizinhos mais próximos.



(a) Resultados de COF (b) Resultados de LOF

Figura 4.5: Comparação entre as abordagens COF e LOF [Amer, 2011]

Connectivity-Based Outlier Factor

Em [Tang et al., 2002], é apresentada uma nova variante do método referido anteriormente, denominado *Connectivity-Based Outlier Factor* (COF), que propõe melhorar a eficiência da abordagem LOF.

[Amer, 2011] introduz um exemplo ilustrativo dos problemas da referida abordagem. Este exemplo (Figura 4.5) é baseado num conjunto bidimensional apresentado em [Tang et al., 2002], onde é possível visualizar a aptidão da abordagem COF face a LOF no tratamento de anomalias que se desviam de padrões caracterizados por baixas densidades, para o valor de $k = 10$. Por definição, A e B são considerados elementos anómalos, sendo A menos anómalo comparativamente a B, uma vez que os restantes elementos formam um padrão composto por uma linha contínua. No entanto, o método LOF falha na sua abordagem ao problema. Na figura, o diâmetro e a cor de cada elemento reflete o grau de anomalia atribuído por cada um dos métodos. Como é possível verificar em (b), é atribuído um maior grau aos elementos nas extremidades da linha do que a B, o que esta errado. Por outro lado, o método COF (b) é capaz de atribuir classificações de acordo com a definição de anomalias.

A diferença entre os dois métodos prende-se com a forma de cálculo dos k-vizinhos de uma dada instância. Em COF, esta é feita de forma incremental,

iniciando-se com a adição do objeto mais próximo ao conjunto dos k-vizinhos. Os restantes *(k-1)* elementos são selecionados da seguinte forma:

Seja $P,Q \subseteq D$, $P \cap Q = \emptyset$ e $P,Q \neq \emptyset$. A distância entre P e Q é representada como $d(P,Q)$ e definida por:

$$d(P,Q) = min\{d(x,y) : x \in P \ \ \& \ \ y \in Q\} \tag{4.8}$$

Para um dado $q \in Q$, q é o vizinho mais próximo de P em Q se existir um $p \in P$ onde $d(p,q) = d(P,Q)$.

O autor denomina *SBN-path* (notação para *set based nearest path*) como sendo um conjunto composto pela sequência $\langle p_1,p_2,...,p_r\rangle$, onde para todos os valores de $1 \leq i \leq r-1$, p_{i+1} é o vizinho mais próximo do conjunto $\{p_1,...,p_i\}$ sendo que pertence a $\{p_{i+1},...,p_r\}$.

Considere-se uma sequência *SBN-path* $s = \langle p_1, p_2, ..., p_r\rangle$. Seja *SBN-trail* (notação para *set based nearest trail*) uma sequência de vértices $\langle e_1,..., e_{r-1}\rangle$, onde para todos os valores de $1 \leq i \leq r-1$, $e_i = (o_i, p_{i+1})$ para $o_i \in \{p_1,...,p_i\}$, e $d(e_i) = d(o_i,p_{i+1}) = d(\{p_1,...,p_i\},\{p_{i+1},...,p_r\})$

Seja $G = \{p_1,...,p_r\}$ um subconjunto de D, a distância média ponderada de p_1 a $G - \{p_1\}$ é dada pela seguinte expressão:

$$ac\text{-}dist_G(p_1) = \frac{1}{r-1} \cdot \sum_{i=1}^{r-1} \frac{2(r-i)}{r} \cdot d(e_i) \tag{4.9}$$

De referir que os vértices inicias obtêm um maior preso do que os restantes. Deste modo, se os vértices mais próximos de p_1 obtiverem valores mais altos comparativamente com os mais distantes, então estes serão mais influentes para o valor de $ac\text{-}dist_G(p_1)$.

Por fim, seja $p \in D$ e k um inteiro positivo, o valor de COF de p relativamente aos seus k-vizinhos é definido por:

$$COF_k(p) = \frac{|N_k(p)| \cdot ac\text{-}dist_{N_k(p)}(p)}{\sum_{o \in N_k(p)} ac\text{-}dist_{N_k(o)}(o)} \tag{4.10}$$

Assim, quanto maior o valor de $COF_k(p)$ atribuído ao objeto p, maior será a sua probabilidade de ser uma anomalia.

Local Correlation Integral

Em [Papadimitriou et al., 2003] é apresentada uma nova variante do método LOF, denominada *Local Correlation Integral* (LOCI), capaz de detetar objetos e *micro-clusters* anómalos. Este método utiliza uma medida denominada *Multi-Granularity Deviation Factor* (MDEF), cujo seu valor para uma dada instância corresponde ao desvio relativo da densidade local da própria instância face à média da dos seus vizinhos mais próximos.

O método LOCI apresenta uma grande vantagem comparativamente aos anteriormente mencionados pois não existe a necessidade do utilizador definir o valor de k, uma vez que este é parâmetro critico e não trivial. Atente-se à abordagem LOF para um dado conjunto de dados, onde estão presentes apenas por dois *clusters* C_1 e C_2 com densidades similares e compostos por β e $\beta + 1$ objetos, respetivamente. Se for definido um $k = \beta$, os objetos presentes em C_1 obterão um maior valor de LOF, comparativamente a C_2.

Considere-se o conjunto de objetos $P = \{p_1,...,p_N\}$, onde N representa o número de elementos no *dataset*. A vizinhança da instância p_i, incluindo o próprio, é dada pela seguinte expressão:

$$N(p_i, r) = \{p \in P | d(p,p_i) \leq r\} \tag{4.11}$$

Onde $d(p,p_1)$ representa a distância do ponto p a p_i, sendo p um r-vizinho. O

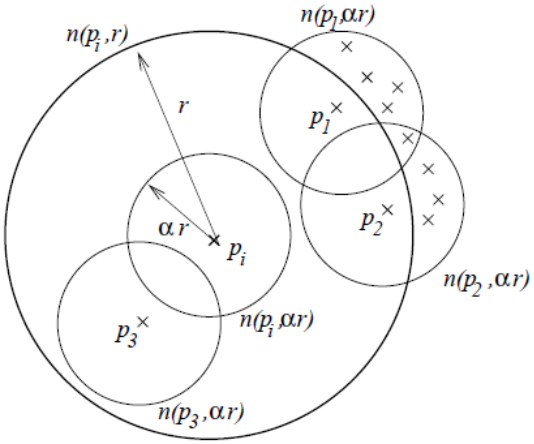


Figura 4.6: Exemplo do cálculo de $n(p_i, r)$ e $n(p_i, \alpha r)$ [Papadimitriou et al., 2003]

número de r-vizinhos de p_i é dado pela expressão:

$$n(p_i, r) = |N(p_i,r)| \tag{4.12}$$

Seja α um número real entre 0 e 1, a média de $n(p, r\alpha)$ sobre a sua r-vizinhança, é dada pela seguinte fórmula:

$$\hat{n}(p_i, r, \alpha) = \frac{\sum_{p \in N(p_i,r)} n(p, r\alpha)}{n(p_i,r)} \tag{4.13}$$

Para uma melhor perceção, considere-se o exemplo da Figura 4.6. Neste, é possível verificar que $n(p_i, r) = 4$, $n(p_3, r\alpha) = 1$ e que $n(p_1, r\alpha) = 6$. Assim, o valor de $\hat{n}(p_i, r, \alpha) = (6 + 5 + 1 + 1)/4 = 3.25$.

Do mesmo modo, o desvio padrão é dado por:

$$\sigma_{\hat{n}}(p_i,r,\alpha) = \sqrt{\frac{\sum_{p \in N(p_i,r)} (n(p,\alpha r) - \hat{n}(p_i,r,\alpha))^2}{n(p_i,r)}} \tag{4.14}$$

Por fim, o cálculo de $MDEF(p_i,r,\alpha)$ e respetivo desvio padrão, são definidos

da seguinte forma:

$$MDEF(p_i,r,\alpha) = 1 - \frac{n(p_i,\alpha r)}{\hat{n}(p_i,r,\alpha)} \tag{4.15}$$

$$\sigma_{MDEF}(p_i,r,\alpha) = \frac{\sigma_{\hat{n}}(p_i,r,\alpha)}{\hat{n}(p_i,r,\alpha)} \tag{4.16}$$

Assim, para qualquer objeto p_i, este será considerado uma anomalia se:

$$MDEF(p_i,r,\alpha) > k_\sigma \sigma_{MDEF}(p_i,r,\alpha) \tag{4.17}$$

Onde k_σ determina o que é significante relativamente ao desvio. Os autores deste método definem $k_\sigma = 3$.

Influenced Outlierness

Em [Jin et al., 2006] é apresentado um novo método de deteção de anomalias, baseado em LOC. Este método surge com o objetivo de tratar casos onde *clusters* com diferentes densidades se situam muito próximos.

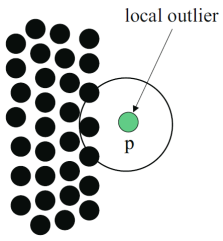


Figura 4.7: Anomalia local [Jin et al., 2006]

Para uma melhor perceção deste problema, considere-se a Figura 4.7. A figura apresenta um conjunto de dados composto por duas dimensões, onde é possível afirmar que o ponto p se trata de uma anomalia local, para um valor de $k = 3$, uma vez que este apresenta uma menor densidade relativamente aos restantes elementos presentes no conjunto.

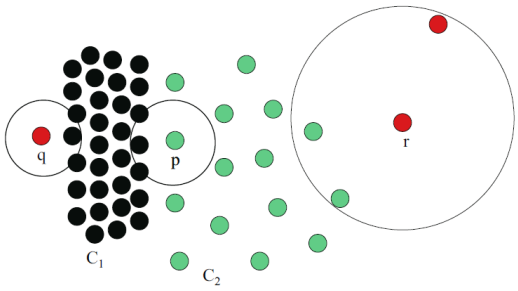


Figura 4.8: Possíveis anomalias p,q e r [Jin et al., 2006]

Considere-se agora o caso ilustrado na Figura 4.8. Nesta é possível verificar a existência de dois *clusters* próximos, contudo distintos, uma vez que C_2 é composto por objetos mais dispersos do que C_1. No presente caso, é possível afirmar que os pontos q e r são, de facto, anomalias. No entanto, o ponto p, pertencendo a C_2, corre o risco de ser considerado como uma anomalia se aplicado o método LOF, uma vez que, comparativamente a q, estes possuem aproximadamente a mesma densidade, mas q situa-se mais próximo de C_1. Além disso, apesar de r apresentar uma menor densidade do que p, a densidade média dos seus vizinhos será mais baixa do que a dos de p, aumentando assim a possibilidade de este ser considerado mais anómalo segundo LOF.

Com o objetivo de resolver os problemas apresentados, foi proposta uma nova medida, denominada *INFLuenced Outlierness* (INFLO), que tem como base o relacionamento simétrico entre um dado elemento e os seus vizinhos aquando do cálculo da densidade local deste. Para o cálculo de INFLO é tido em conta não só os vizinhos mais próximos de um dado objeto (*Nearest Neighbors*, NN), mas também a quantidade de elementos que têm o objeto em análise como vizinho (*Reverse Nearest Neighbors*, RNN).

Seja o método de cálculo dos k-vizinhos mais próximos de um ponto p a mesma da abordagem LOF, o cálculo dos RNN de p é dado pela expressão:

$$RN_k(p) = \{q \mid q \in D \ \& \ p \in N_k(p)\} \tag{4.18}$$

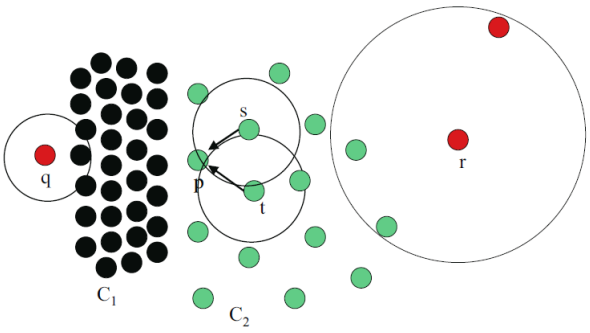


Figura 4.9: Identificação dos RNN de p [Jin et al., 2006]

Considerando esta abordagem, e retomando o exemplo apresentado acima, atente-se à Figura 4.9. Nesta é possível observar que p possui dois RNNs, s e t. Contrariamente, é possível verificar que q não possui RNNs e que r apenas tem como único RNN um outro objeto anómalo.

Através da combinação dos NN e RNN de p é obtido o espaço de influência (*k-influence space*), isto é:

$$IS_k(p) = N_k(p) \cup RN_k(p) \tag{4.19}$$

Sendo a densidade do ponto p definida por:

$$den_k(p) = \frac{1}{k\text{-}distance(p)} \tag{4.20}$$

A medida INFLO é dada pela expressão:

$$INFLO_k(p) = \frac{den_{avg}(IS_k(p))}{den_k(p)} \tag{4.21}$$

Onde,

$$den_{avg}(IS_k(p)) = \frac{\sum_{o \in IS_k(p)} den_k(p)}{|IS_k(p)|} \tag{4.22}$$

Assim, o valor INFLO será obtido através da comparação da densidade média dos elementos presentes no espaço de influência de p, e a sua própria densidade. Desta forma, o ponto p obterá um maior grau de anormalidade quanto mais baixa for a sua densidade relativamente à dos objetos presentes em IS. Do mesmo modo é possível constatar que valores $INFLO \approx 1$ denotam objetos ditos normais, tipicamente presentes no interior de *clusters*.

Local Outlier Probablity

Em [Kriegel et al., 2009] é apresentada uma outra variante dos métodos de densidade locais, denominada *Local Outlier Probablity* (LoOp). Tal como os métodos acima apresentados, este algoritmo mede o grau de anormalidade de cada objeto, ou seja, o fator de anomalia. Contudo, os autores realçam os problemas inerentes à interpretação dos valores atribuídos às instâncias por parte de utilizadores não familiarizados com o método de deteção em questão. Assim, é proposta uma nova abordagem que atribui aos diversos objetos uma pontuação contida no intervalo [0,1], representando desta forma a probabilidade de um dado elemento ser ou não uma anomalia. Para tal, os autores introduzem o conceito de distância probabilística (*pdist*), onde dado um objeto $o \in D$ e um conjunto $S \subseteq D$ centrado em o, esta é definida pela expressão:

$$pdist(\lambda,o,S) = \lambda \cdot \sigma(o,S) \tag{4.23}$$

onde λ representa um fator de normalização e

$$\sigma(o,S) = \sqrt{\frac{\sum_{s \in S} d(o,s)^2}{|S|}} \tag{4.24}$$

define a distância padrão do objeto o a S. O rácio da estimativa da densidade, denominado de *Probabilistic Local Outlier Factor* (PLOF) é definido pela expressão:

$$PLOF_{\lambda,S}(o) = \frac{pdist(\lambda,o,S) \cdot |S|}{\sum_{s \in S} pdist(\lambda, s, S)} - 1 \tag{4.25}$$

Sendo que um dos principais objetivos deste método é uma mais fácil interpretação, é necessário aplicar um fator de escala aos resultados, tornando-os independentes de qualquer distribuição. Para tal é apresentada a medida $nPLOF$:

$$nPLOF(\lambda) = \lambda \cdot \sqrt{\frac{\sum_{o \in D} PLOF_{\lambda,S}(o)^2}{|D|}} \tag{4.26}$$

Após isto, é aplicada uma função de erro Gaussiana (erf) para obter o valor probabilístico de um dado objeto o ser anómalo, denominado *Local Outlier Probability* (LoOP):

$$LoOP_S(o) = max\{0, \text{erf}\left(\frac{PLOF_{\lambda,S}(o)}{nPLOF \cdot \sqrt{2}}\right)\} \tag{4.27}$$

Através da aplicação desta medida, os objetos situados em zonas de alta densidade obterão um valor próximo de 0. Por outro lado, aos que se situam em áreas menos densas será atribuída uma pontuação próxima de 1, evidenciando-os como prováveis anomalias.

KNN-kth

Método proposto por Ramaswamy et al., que efetua a deteção de anomalias através da medição da distância de um objeto p ao seu k-ésimo vizinho mais próximo e,

como resultado final, apresenta um top-n de objetos anómalos.

Assim, seja D um conjunto de dados, k e n dois inteiros positivos. Para um dado objeto $p \in D$, este será considerado uma anomalia se não existem mais de $(n-1)$ elementos p' cuja $k\text{-}distance(p') > k\text{-}distance(p)$.

Sendo que os métodos baseados na distância tendem a efetuar grandes quantidades de comparações a fim de identificar qual o k-ésimo vizinho mais próximo de um dado objeto, estas abordagens sofrem de problemas de eficiência e originando computações quadráticas.

Com o objetivo de solucionar este problema, [Bay and Schwabacher, 2003] propõem um algoritmo, implementado com recurso a ciclos aninhados, que, no pior caso, executa em tempo quadrático. No entanto, permite uma execução em tempo quase linear através da introdução de uma regra de *pruning* se o *dataset* recebido estiver aleatoriamente ordenado.

Este algoritmo remove objetos da análise se a distância desse objeto ao seu (até ao momento) k-vizinho mais próximo for menor do que um dado valor de referência. Este valor é definido como distância mínima do *outlier* mais fraco presente no top-n nesse momento. Sendo que, quanto mais instâncias forem analisadas, mais fortes anomalias são encontradas. O valor de referência será cada vez mais elevado aumentando assim o número de elementos excluídos da análise.

KNN-avg

Este método foi introduzido em [Angiulli and Pizzuti, 2002] e assente no conceito de vizinho mais próximo.

Para a deteção de anomalias, esta abordagem calcula a distância média entre cada objeto pertencente ao conjunto dados e os seus k-vizinhos mais próximos.

Assim, seja $p \in D$, onde D representa um dado conjunto, este método utiliza

a seguinte expressão para o cálculo do grau de anomalia de p.

$$knn(p) = \frac{\sum_{o \in N_k(p)} d(p,o)}{|N_k(p)|} \tag{4.28}$$

4.6 Tipos de Resultados

Tipicamente, existem dois tipos distintos de resultados obtidos através da aplicação das diferentes técnicas de deteção de anomalias. Em [Chandola et al., 2009], estes são denominados *rótulos* e *pontuações*.

No primeiro caso, os algoritmos retornam como resultado um rótulo associado a cada registo presente no *dataset*. Na maioria dos casos esta atribuição é binária. Ou seja, é atribuída uma designação aos objetos considerados normais e outra aos desviantes desse padrão, como por exemplo, "normal" e "anormal".

Este tipo de resultado é o que possibilita uma mais fácil compreensão por parte do analista, pois aquando da revisão dos resultados, a distinção binária é mais intuitiva.

Por outro lado, existem métodos que retornam como resultado a pontuação atribuída a cada registo do *dataset*. Esta pontuação reflete o grau de desvio de cada objeto perante o que é considerado normal. Quanto maior essa pontuação, mais anómala será a observação. Existem várias formas de apresentar o resultado da análise efetuada. Por exemplo, através da apresentação de um top-k, sendo k um número inteiro positivo previamente definido. Pode também ser uma lista cuja pontuação de cada registo seja superior a um dado valor, ou através da apresentação da lista integral.

Sendo o método de atribuição de uma pontuação a cada registo mais flexível e passível de personalização, é, por outro lado, menos intuitivo ao analista, pois caberá a este definir o valor a partir do qual considerará um dado objeto como

anómalo.

4.7 Síntese

No presente capítulo foi abordado o tema da Deteção de Anomalias. Com base na literatura foi definido o conceito de anomalia, bem como os diversos tipos existentes de anomalias.

Foram analisadas diversas abordagens que tentam dar resposta a este tipo de problema, bem como quais os diversos métodos de supervisão existentes. Nesta fase, foram identificados quatro tipos principais de abordagens, nomeadamente, as baseadas em classificadores, em modelos estatísticos, em técnicas de *clustering* e, finalmente, as que têm como base o conceito de vizinho mais próximo, isto é, *Nearest-Neighbor*. As abordagens assentes em *Nearest-Neighbor* foram subdivididas em métodos locais e globais, onde no primeiro caso, um dado objeto é analisado tendo em conta os seus vizinhos mais próximos e, no segundo caso, a análise do objeto tem em conta todo o conjunto de dados.

Foram ainda apresentados dois tipos de resultados possíveis obtidos através da aplicação das referidas abordagens.

Capítulo 5

Descoberta de Conhecimento

Sendo o principal objetivo deste projeto de dissertação a identificação de comportamentos fraudulentos nos modelos de PPC, é fundamental transformar a informação disponível em conhecimento. Será através desta conversão que determinadas decisões poderão ser suportadas aquando da identificação deste tipo de comportamentos.

O processo de descoberta de conhecimento em bases de dados, do inglês *Knowledge Discovery in Databases* (KDD), proposto por Fayyad et al. [Fayyad et al., 1996], é composto por cinco etapas sequenciais:

- Seleção - etapa na qual, após a definição do objetivo do problema, são selecionados os dados necessários à resolução do problema.
- Pré-processamento - onde, após a seleção dos dados, estes são tratados, eliminando ruído, omissões, inconsistências, etc.
- Transformação - etapa responsável pela manipulação dos dados, como por exemplo, agregações, redução do número de dimensões, etc.
- *Data Mining* - fase de seleção e aplicação de métodos e técnicas capazes de revelar padrões presentes nos dados.

- *Interpretação* - etapa na qual são analisados e discutidos os resultados produzidos pelos métodos aplicados, dando origem à extração de conhecimento.

O esquema gráfico representante deste processo pode ser visualizado na Figura 5.1. Este é um processo fundamental em qualquer projeto de descoberta de conhecimento, e será por isso adotado no desenvolvimento da solução pretendida.

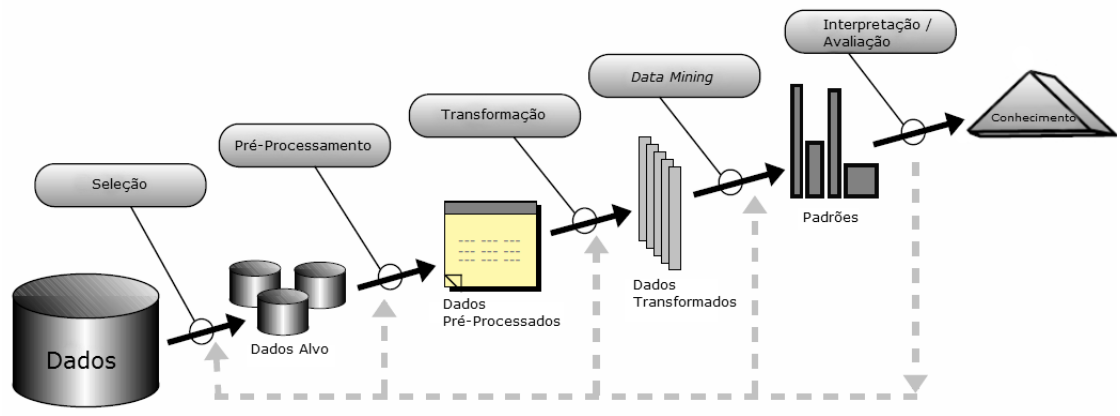


Figura 5.1: Etapas do processo de KDD [Fayyad et al., 1996]

5.1 Seleção

Definido o âmbito desta dissertação (Capítulo 2), segue-se o momento da seleção dos dados.

Uma correta seleção dos dados a utilizar é um dos passos mais importantes para o sucesso de qualquer sistema de deteção de fraude. Deste modo, é necessária uma análise cuidada de todos os dados disponíveis, a fim de identificar quais os mais relevantes para o problema. Além disso, é fundamental que esta análise tenha em conta que as informações a selecionar podem influenciar o desempenho e a qualidade dos resultados produzidos pelos métodos a aplicar.

A Tabela 5.1 lista os dados disponíveis, recolhidos do contexto real apresentado na Secção 2.5.

Atributo	Tipo
Sessão HTTP	*String*
HTTP Cookie	*String*
ID Utilizador	Inteiro
Endereço Host	*String*
Nome Host	*String*
User Agent	*String*
ID Região	Inteiro
ID Sub-região	Inteiro
Data do Pedido	Data
URL Acedido	*String*
Referência	*String*
ID Clip	Inteiro

Tabela 5.1: Atributos disponíveis e respetivos tipos de dados

O atributo *Sessão HTTP* representa a chave alfanumérica gerada pelo servidor *Web* e enviada para o visitante, segundo o protocolo HTTP, a fim de o poder identificar em futuras interações. Estas sessões têm, tipicamente, um curto período de vida, uma vez que expiram ao fim de um determinado tempo de inatividade ou quando o utilizador encerra o seu navegador (em inglês *Web Browser*). Estes dados revelam significativa importância pois permitem identificar o utilizador na fase seguinte, ou seja, na transformação.

O *HTTP Cookie* é o atributo responsável por guardar a chave identificativa do *Cookie* do visitante. O seu funcionamento é semelhante ao parâmetro anterior, mas este tem a capacidade de guardar informações adicionais e perdura mesmo após o encerramento do navegador. De igual forma, esta informação revela-se importante, pois permite detetar o mesmo utilizador em visitas distintas (sessões diferentes).

ID Utilizador é uma chave numérica utilizada para identificar as ações realizadas por um utilizador após a sua autenticação. Analisando os cliques realizados pelos utilizadores, verifica-se que na sua grande maioria (96.34%) o utilizador não

é conhecido, ou seja, nunca se autenticou na plataforma. Este valor pode ser explicado pelo facto de que, só os cliques posteriores à autenticação obtêm a identificação do utilizador, deixando os registos anteriores por atualizar. Assim, foi decidido ignorar este atributo.

Endereço e *Nome do Host* são dois atributos que identificam qual o dispositivo através do qual o utilizador se liga na rede. Sendo de certa forma redundantes, apenas o *Endereço* foi selecionado, uma vez que permitirá analisar se o utilizador revela intenções fraudulentas, alterando constantemente o seu IP (*Internet Protocol*).

No atributo *User Agent* são guardadas as informações do produto e da versão do navegador utilizado. Sendo um comportamento típico dos utilizadores mal intencionados a mudança de *Browser*, este parâmetro revela-se fundamental.

Os atributos numéricos *ID Região* e *ID Sub-região* são utilizados para identificar o local geográfico do qual o visitante efetua os seus pedidos *Web*. Analisando o atributo *ID Região*, verifica-se que aproximadamente 73% são cliques provenientes de Portugal e 14% do Brasil. A França, Reino Unido e EUA, representam respetivamente 1.5%, 1.3% e 1% dos cliques. Relativamente ao *ID Sub-região* verifica-se que dos 73% de cliques provenientes de Portugal, 41% tiveram origem na zona centro (com especial impacto da zona da grande Lisboa), 9% provieram do distrito do Porto e 30% têm origem desconhecida.

Data de Pedido contém a informação do momento em que foi realizada determinada ação, como por exemplo, um clique. Este atributo será fundamental para a análise comportamental do visitante.

URL Acedido e *Referência* são dois atributos alfanuméricos responsáveis por acolherem *Uniform Resource Locators* (URL). *URL Acedido* representa o endereço pedido pelo utilizador enquanto que a *Referência* indica qual a origem do pedido. É através do *URL Acedido* que se procederá à análise comportamental do visitante,

Atributo	Tipo
Sessão HTTP	*String*
HTTP Cookie	*String*
Endereço Host	*String*
User Agent	*String*
ID Região	Inteiro
ID Sub-região	Inteiro
Data do Pedido	Data
URL Acedido	*String*
ID Clip	Inteiro

Tabela 5.2: Atributos selecionados e respetivos tipos de dados

sendo assim um parâmetro essencial.

O atributo *ID Clip* indica qual o editor através do qual foi efetuada determinada ação. Se o visitante tiver intenções de defraudar o sistema, poderá recorrer a múltiplos editores para agilizar o processo, sendo este atributo fundamental para a identificação desses casos.

Analisados os dados, os atributos selecionados encontram-se na Tabela 5.2.

5.2 Pré-Processamento

Selecionados os dados, é fundamental tratá-los de forma a obter um conjunto de dados consistente e capaz de providenciar informação relevante para a fase seguinte do processo.

Após a análise efetuada na etapa anterior, foi possível concluir que não existiam atributos com valores omitidos ou incoerentes.

Do ponto de vista de relevância dos dados, foram eliminados todos os registos cuja sessão (*Sessão HTTP*) não possuía cliques, pois estas são consideradas irrelevantes para o problema de deteção de cliques fraudulentos. Assim, de um total de 127197 sessões disponíveis, apenas 7508 foram consideradas pertinentes para o

problema em questão.

5.3 Transformação

Filtrada e tratada a informação disponível, foi necessário adaptá-la de forma a que fosse possível obter um *dataset* propício à deteção de fraude neste tipo de modelo de negócio. Sendo o principal objetivo a deteção de utilizadores mal intencionados, é necessário assegurar que o *dataset* a construir seja composto por informações capazes de evidenciar indícios comportamentais anómalos, relativamente aos considerados normais.

Questões como:

- Através de quantas categorias navegou o visitante?
- Através de quantos editores?
- Com que frequência efetuou cliques?
- Quantos cliques fez?
- Quantos endereços de IP usou?
- Quantos *cookies* lhe foram atribuídos?
- Quantas sessões efetuou?
- Com quantos *browsers* navegou?
- A partir de quantos locais?

devem ser respondidas analisando o *dataset* produzido.

Numa primeira fase, foram processados todos os URL aos quais os utilizadores acederam, de forma a transformar um endereço *Web*, numa ação efetuada por um

IDSessão	NClips	NCat	NCliq	TEC	NIP	NCookies	NSessões	NBrowser	NLocal
0lovwobj2	4	1	3	21	1	1	1	1	1
0obezrrrr	1	1	2	35	1	1	1	1	1
0oz01w2rk	3	1	5	35,75	1	1	1	1	1

Tabela 5.3: Excerto do *dataset* produzido

utilizador. Após esta transformação, foi contabilizado o número de ações efetuadas por cada utilizador, a fim de responder às questões acima descritas relativas à quantidade de cliques e categorias visitadas numa dada sessão, originando os parâmetros *NCliq* e *NCat*, respetivamente.

Sendo que a transformação dos dados foi produzida tendo como base o código da sessão, e com vista a obter o número de sessões efetuadas, foi feito o cruzamento de dados através do *Cookie*, a fim de identificar sessões com códigos diferentes do atual que possuíssem o mesmo *Cookie*. Assim, foi possível contabilizar o número anterior de visitas efetuadas por um dado utilizador, representado pelo parâmetro *NSessões*.

Relativamente à frequência com que um dado visitante efetua cliques, foi contabilizada a diferença temporal entre os registos deste tipo de ação, recorrendo ao atributo *Data do Pedido*, dando origem ao parâmetro *TEC* (Tempo entre cliques).

Quanto às restantes questões, estas foram respondidas através da contabilização dos valores distintos de cada atributo, dando origem aos parâmetros *NClips*, *NIP*, *NCookies*, *NBrowser* e *NLocal*, respetivamente, número de clips, IP, *cookies*, *browsers* e locais a partir dos quais o visitante acedeu ao serviço.

A Tabela 5.3 apresenta um excerto do *dataset* produzido. Este poderá ser classificado como sendo multi-variado e composto por valores contínuos.

5.4 *Data Mining* - Deteção de Anomalias

Após concluídas todas as etapas necessárias à obtenção de um conjunto de dados devidamente tratado e propício à extração de conhecimento, é necessário identificar quais os métodos mais adequados para o problema em questão.

Sendo que em momento algum é possível afirmar inquestionavelmente se um dado visitante tem pretensões fraudulentas, é através da análise do seu comportamento que se pode retirar ilações sobre quais os seus objetivos. Assim, o comportamento de um dado visitante apenas poderá ser considerado anormal relativamente aos restantes.

Uma vez que os dados disponíveis foram extraídos de um contexto real e tendo em conta o referido anteriormente, não é possível a criação de um conjunto de dados de treino contendo registos devidamente classificados como normais ou anormais. Deste modo, apenas os métodos de *Data Mining* não-supervisionados podem ser aplicados a este problema.

Do mesmo modo, e com base no *dataset* criado, apenas métodos capazes de operar sobre *datasets* multivariados (múltiplos atributos) e compostos por valores contínuos podem ser selecionados.

Considere-se o capítulo 4 e abordagens nele expostas. Relativamente aos métodos de classificação, estes revelam-se inapropriados para o problema em questão uma vez que, na sua grande maioria, estes operam de forma supervisionada ou semi-supervisionada. Além disso, expõem o seu resultado na forma de *labels*, isto é, através da atribuição de um rótulo a cada instância presente no conjunto de dados, classificando-as como normais ou anómalas. Apesar de este tipo de resultado ser mais imediato e intuitivo para o analista, esta abordagem retira sensibilidade na análise dos resultados.

Tal como os métodos de classificação, as abordagens assentes em modelos estatísticos revelam-se inadequadas face ao problema exposto. Relativamente aos

métodos paramétricos, a sua necessidade de conhecimento prévio dos parâmetros é, neste contexto, impraticável. Além disso são, na sua maioria, aplicáveis apenas a conjuntos de dados univariados. Quanto às abordagens não paramétricas, estas revelam-se incapazes de detetar relações entre atributos [Chandola et al., 2009].

Relativamente aos três tipos de abordagens baseadas em *clustering* (ver 4.5.3), os métodos do tipo um sofrem de algumas limitações uma vez que não são especializados na deteção de anomalias. Os métodos do tipo dois, que medem a distância de uma dada instância ao centróide mais próximo, revela-se incapaz de detetar anomalias que no seu conjunto formem *clusters*. Além disso, sendo abordagens globais, mostram-se limitadas na deteção de anomalias locais. Por outro lado, os algoritmos de deteção de anomalias baseados em *clustering* do tipo três revelam uma abordagem interessante para o problema em questão. Desta forma, foi selecionado o algoritmo CBLOF, proposto em [He et al., 2003] e abordado anteriormente nesta dissertação (Secção 4.5.3).

Dada a grande variedade de métodos existentes deste tipo de abordagem, a obtenção de resultados apresentados sob forma de pontuação, representativa do seu grau de anormalidade, e a sua natureza não-supervisionada, os métodos baseados em *Nearest-Neighbor* são a abordagem que melhor se adequa ao problema apresentado.

Assim, os seguintes métodos irão ser aplicados ao conjunto de dados obtido após a fase de transformação:

- Baseados em densidade
  - *Local Outlier Factor* (LOF)
  - *Connectivity-Based Outlier Factor* (COF)
  - *Local Correlation Integeral* (LOCI)
  - *Local Outlier Probablity* (LoOP)

  - *Influenced Outlierness* (INFLO)
  - *Cluster-Based Local Outlier Factor* (CBLOF)

- Baseados em distância
  - KNN-kth
  - KNN-avg

Local Outlier Factor foi a primeira abordagem baseada na densidade local. Como consequência, diversas variantes foram propostas (COF, LOCI, LoOp e INFLO). Assim, este método poderá ser usado como termo de comparação com os restantes métodos que propuseram corrigir falhas presentes nesta técnica. Deste modo será também possível verificar se as falhas em LOF e as melhorias propostas pelas diversas variantes se verificam no presente contexto.

Relativamente aos métodos que baseiam a sua medição na distância, estas permitirão comparar os resultados das abordagens globais face às locais. O algoritmo KNN-kth utiliza como medida a distância ao k-ésimo vizinho mais próximo. Por outro lado, KNN-avg recorre ao cálculo da média da distância de um dado ponto aos seus k vizinhos.

Sendo que este tipo métodos de deteção de anomalias operam em modo não-supervisionado, e que não existem dados devidamente catalogados, não será possível avaliar os resultados obtidos com métricas que recorrem, por exemplo, ao rácio entre falsos positivos e falsos negativos ou a curvas ROC (*Receiver Operating Characteristics*). Por este motivo, foram introduzidos no *dataset* instâncias intencionalmente elaboradas de modo a, aquando da obtenção dos resultados, ser possível comparar os diversos algoritmos de um ponto de vista crítico e de interpretação. Estas instâncias devem ser diversificadas, de forma a representarem várias situações passíveis de acontecer no contexto real. Para a definição destas

novas instâncias a inserir no *dataset* foi necessário o conhecimento prévio da gama de valores de cada atributo pertencente ao conjunto de dados.

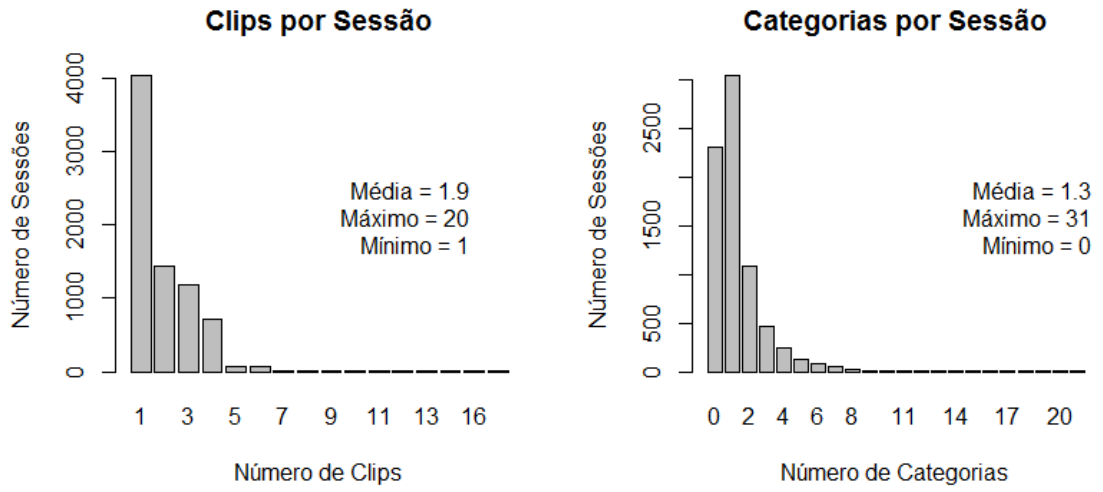


Figura 5.2: Número de clips e categorias por sessão

Relativamente aos valores assumidos pelos atributos NClips e NCat, observe-se a Figura 5.2. No primeiro caso é possível constatar que os seus valores variam entre 1 e 20, sendo que sua média é aproximadamente de 2 clips por sessão e onde a grande maioria dos visitantes utiliza menos de 5 clips.

Quanto ao número de categorias visitadas, estas podem variar entre 0 e 31. As visitas que não acederam a nenhuma categoria são consequência de cliques efetuados em *banners* publicitários que direcionam o visitante diretamente para um dado anúncio. A maior parte dos utilizadores visita menos de quatro categorias.

Analisando a Figura 5.3 é possível observar que a gama de valores relativa ao número de cliques efetuados varia entre 1 e 519, sendo que o mais comum é que os visitantes apenas efetuem um clique por sessão e que a média se situa nos 3.6.

Relativamente ao tempo entre cliques, este valor varia entre os 0 e os 429224 segundos (aproximadamente 5 dias de diferença). O valor mais inferior pode ser

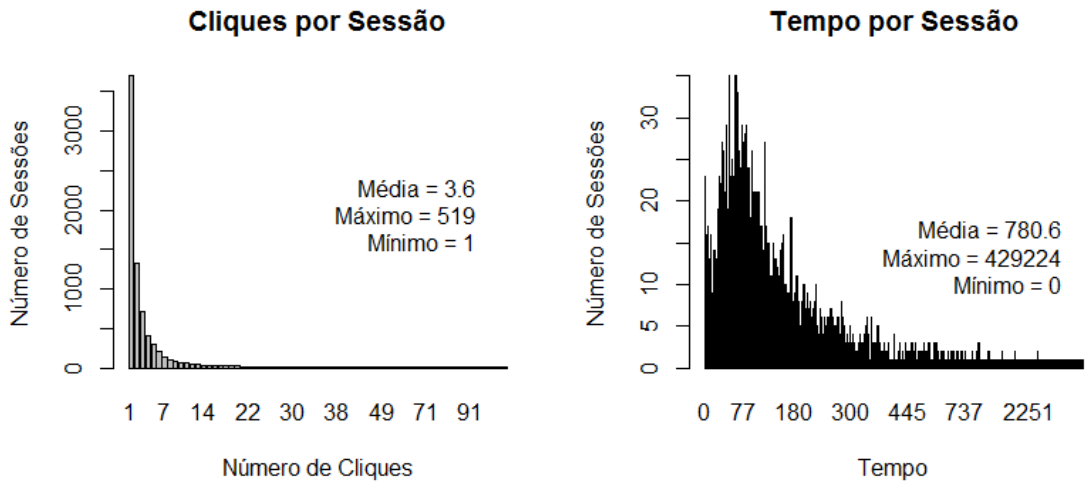


Figura 5.3: Número de cliques e tempo entre cliques por sessão

resultante de duplos cliques, ou seja, dois cliques efetuados quase no mesmo instante. Já o valor máximo deste atributo poderá resultar de um utilizador que tenha mantido o site aberto durante cinco dias. De notar que não é possível deduzir pelos dados disponíveis se o utilizador esteve ativo durante este período. Isto é, se esteve efetivamente a visualizar o anúncio. Este tipo de situações fazem com que a média de tempo entre cliques se situe nos 780 segundos, ou seja, 13 minutos. Sendo que também não será possível calcular este valor para visitas que possuam apenas um clique (a grande maioria, como visto anteriormente) este atributo poderá prejudicar o processo de deteção de anomalias.

Relativamente ao número de endereços utilizados pelos visitantes numa dada sessão, é possível observar que, como esperado, o valor mínimo é de 1 IP por sessão e o máximo de IPs utilizados por um dado visitante foi 24 (Figura 5.4). Como a esmagadora maioria de utilizadores apenas utiliza um endereço, a média deste atributo é de aproximadamente 1 IP/Sessão.

Analisando o número de *cookies* utilizados por cada sessão, o seu comporta-

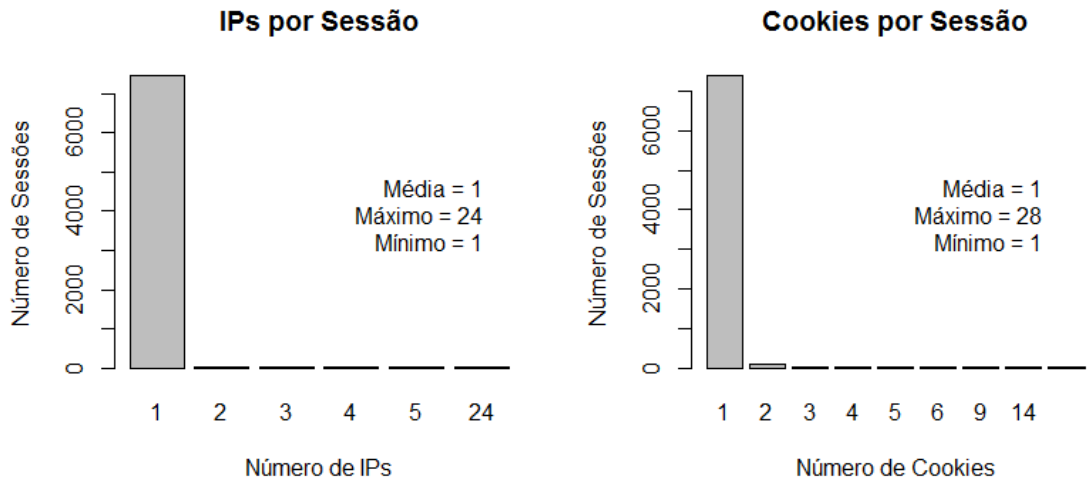


Figura 5.4: Número de IPs e *cookies* por sessão

mento é muito semelhante ao número de IPs. O seu valor varia entre 1 e 28 *cookies* por sessão e o valor médio do atributo é de 1 *cookie*.

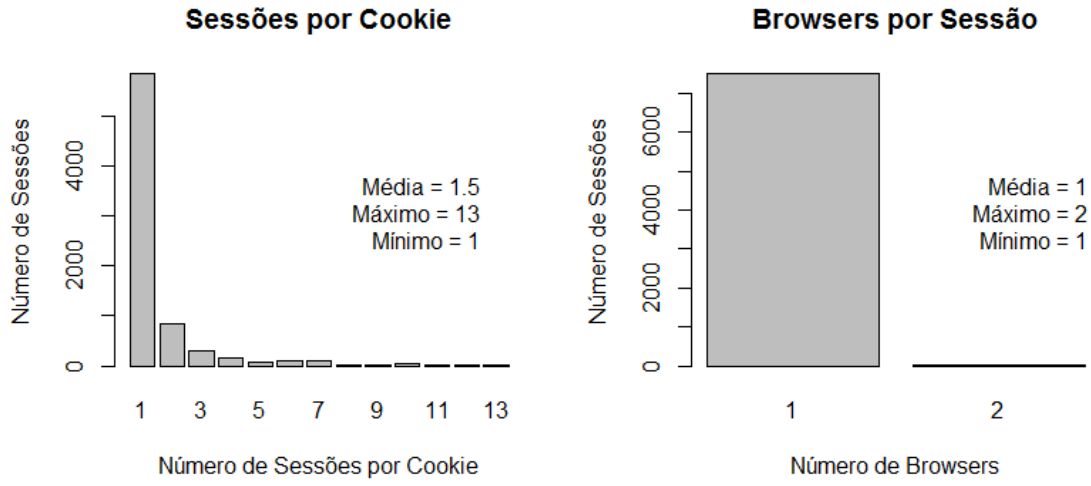


Figura 5.5: Número de sessões por *cookie* e *browser* por sessão

Considere-se a Figura 5.5. Analisando o número de sessões por *cookie* de cada

visitante é possível verificar que a sua gama de valores varia entre 1 e 13 sessões/*cookie*. O valor mais frequente deste atributo é o 1 e em média é de 1.5 sessões por *cookie*. Valores altos revelam que o utilizador já visitou o site anteriormente.

Relativamente ao número de *browsers* usados por cada sessão, verifica-se que no máximo um visitante utiliza 2 navegadores na mesma sessão. A grande maioria dos visitantes apenas utiliza um *browser*. Sendo que por norma a sessão expira quando o navegador é fechado, um número elevado neste atributo poderá indicar que o utilizador está a manipular a informação enviada. No entanto isto também pode acontecer caso o *browser* em questão possua um modo de compatibilidade.

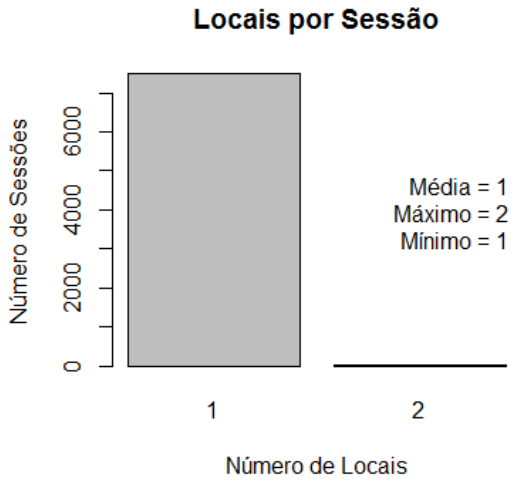


Figura 5.6: Número de locais por sessão

Por último considere-se a Figura 5.6. Nesta é possível visualizar que, tal como o atributo referente ao número de navegadores, a grande maioria dos utilizadores apenas acede ao site de um local numa dada sessão. No presente *dataset* este atributo possui um valor máximo de 2.

Analisados os dados presentes no *dataset* proposto, é agora possível criar um conjunto de instâncias heterogéneas de modo a possibilitar uma análise crítica dos

resultados obtidos através da aplicação dos diferentes algoritmos selecionados. Estas instâncias, serão posteriormente adicionadas ao conjunto inicial, subdivididas em três grupos distintos e originando assim três versões deste.

Assim, duas das instâncias a introduzir no conjunto de dados possuirão os valores médios e mais frequentes de cada atributo. Isto permitirá comparar o comportamento dos diversos algoritmos face a objetos mais comuns e não extremados. Serão também adicionadas duas instâncias contendo os valores mínimos e máximos, de forma a verificar se estas serão consideradas anomalias. A esta versão do conjunto de dados, foi atribuída a designação de *dataset_1*.

Será adicionado um pequeno conjunto de dez elementos com valores próximos entre si, de modo a criar um *cluster* de pequenas dimensões. O objetivo desta adição deve-se à intenção de analisar o comportamento dos métodos selecionados face a grupos de objetos anómalos. Neste caso é esperado que o algoritmo assente em LOCI seja capaz de detetar este tipo de anomalias. O conjunto resultante da adição deste *micro-cluster* ao conjunto inicial foi denominado *dataset_2*.

Por último, serão adicionados ao *dataset* proposto nove novas instâncias, cada uma contendo um valor elevado num dado atributo e o valor mais frequente nos restantes. Com a análise destas instâncias espera-se ser possível verificar qual o atributo mais relevante para cada algoritmo. Deste modo, foi obtido a versão *dataset_3*.

Deste modo, serão adicionados ao conjunto de dados obtido anteriormente 23 novas instâncias, representadas na Tabela 5.4.

5.5 Síntese

Neste capítulo foram abordadas as quatro primeiras etapas do processo de descoberta de conhecimento, sugerido em [Fayyad et al., 1996]. Como resultado da

IDSessão	NClips	NCat	NCliq	TEC	NIP	NCookies	NSessões	NBrowser	NLocal
media	2	1	4	781	1	1	2	1	1
frequente	1	1	1	48	1	1	1	1	1
minimo	1	0	1	0	1	1	1	1	1
maximo	20	31	519	429224	24	28	13	2	2
miniclust_1	21	31	519	429224	24	28	13	2	2
miniclust_2	20	32	519	429224	24	28	13	2	2
miniclust_3	20	31	520	429224	24	28	13	2	2
miniclust_4	20	31	519	429225	24	28	13	2	2
miniclust_5	20	31	519	429224	25	28	13	2	2
miniclust_6	20	31	519	429224	24	29	13	2	2
miniclust_7	20	31	519	429224	24	28	14	2	2
miniclust_8	20	31	519	429224	24	28	13	3	2
miniclust_9	20	31	519	429224	24	28	13	2	3
miniclust_10	20	31	519	429224	24	28	13	2	2
max_NClips	20	1	1	48	1	1	1	1	1
max_NCat	1	31	1	48	1	1	1	1	1
max_NCliq	1	1	519	48	1	1	1	1	1
max_TEC	1	1	1	429224	1	1	1	1	1
max_NIP	1	1	1	48	24	1	1	1	1
max_NCookies	1	1	1	48	1	28	1	1	1
max_NSessoes	1	1	1	48	1	1	13	1	1
max_NBrowser	1	1	1	48	1	1	1	2	1
max_NLocal	1	1	1	48	1	1	1	1	2

Tabela 5.4: Instâncias criadas

execução desta metodologia, foi obtido um conjunto de dados contendo informações devidamente tratadas e que se espera serem relevantes para o problema em questão. Para a obtenção deste conjunto, foi necessária um seleção prévia das informações existentes, seguida de um processo de limpeza de algum ruído e transformação dos dados, levando desta forma à obtenção de um *dataset* propício à extração de conhecimento.

Foram também selecionados os métodos de deteção de anomalias que irão ser aplicados sobre o conjunto criado. Sendo que todos estes métodos irão ser aplicados de modo não-supervisionado, a análise irá ser feita comparando os resultados dos diversos métodos.

Com vista a uma mais fácil interpretação e comparação dos resultados a obter, foram criadas diversas instâncias heterogéneas, que se espera serem úteis, por

exemplo, na identificação dos atributos mais importantes para cada método ou na perceção de como estes lidam com *clusters* de pequenas dimensões constituídos por objetos anómalos. Para a criação deste conjunto de instâncias, foi necessária uma análise prévia dos diversos atributos pertencentes ao *dataset* resultante das etapas anteriores.

Capítulo 6

Resultados

Concluído todo o processo de seleção, pré-processamento, transformação e identificação dos métodos mais propícios à deteção de anomalias no problema apresentado, procedeu-se à aplicação dos referidos métodos sobre o conjunto de dados obtido em etapas anteriores.

Uma vez que os vários métodos escolhidos se baseiam na vizinhança, os testes inicialmente efetuados utilizaram 10 instâncias como o número de vizinhos a considerar ($k = 10$). Este foi escolhido com base no valor apresentado por defeito nas diversas implementações selecionadas.

Como resultado de uma primeira execução dos diversos métodos selecionados, foi detetado um problema relacionado com o atributo responsável pela informação relativa ao tempo médio entre cliques efetuados por sessão. Com a inclusão deste atributo, os vários métodos definiam pontuações, para o conjunto de dados, que variavam entre 0 e $+\infty$. Como referido na Secção 5.4, este dado possui valores com variações muito elevadas. Além disso, as sessões que possuem apenas um clique não têm um valor definido. De forma a resolver este problema, e tal como previsto na Figura 5.1, optou-se por uma nova transformação, neste caso pela discretização do referido atributo. Sendo que quanto menor for o tempo entre cliques mais

provável será de o utilizador possuir intenções menos lícitas, decidiu-se segmentar o atributo nos intervalos [0,5[, [5,15[, [15,30[, [30,∞[. Para os casos em que as sessões possuem apenas um clique, este atributo foi designado como *Indef*. Como resultado desta discretização, o total de instâncias por intervalo pode ser visto na Figura 6.1. Tal como esperado, apenas 69 instâncias possuem tempos entre [0,5[, 116 entre [5,15[, 188 entre [15,30[, 3432 entre [30,∞[e 3702 instâncias não possuem tempo definido.

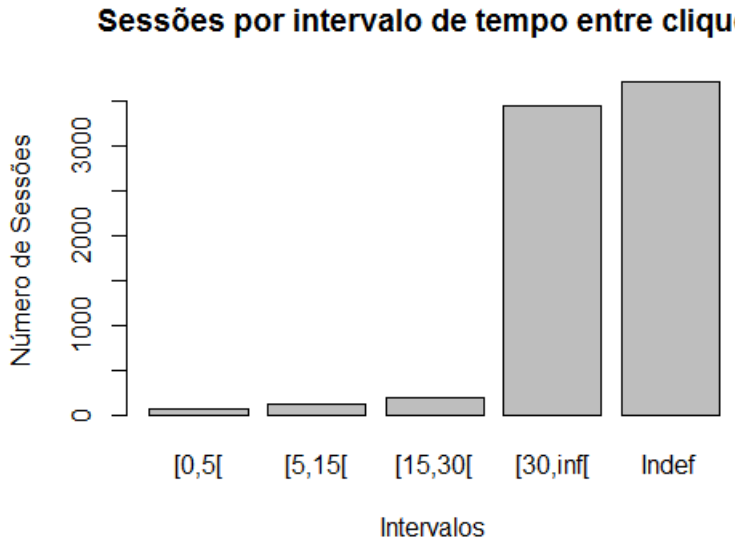


Figura 6.1: Número de sessões por intervalo de tempo entre cliques

Solucionado este problema, existiu a necessidade de reformular as instâncias apresentadas na Tabela 5.4. Assim, as instâncias inseridas no conjunto de dados inicial encontram-se na Tabela 6.1.

De forma a tentar perceber quais os atributos mais relevantes para cada algoritmo, foram inicialmente testadas as instâncias contendo os valores mais frequentes e com um dos atributos com o valor máximo. Isto é, as instâncias nomeadas de *max_NClips* até *max_NLocal*. Para esta análise foram adicionadas ao conjunto

IDSessão	NClips	NCat	NCliq	TEC	NIP	NCookies	NSessões	NBrowser	NLocal
media	2	1	4	[30,inf[	1	1	2	1	1
frequente	1	1	1	Indef	1	1	1	1	1
minimo	1	0	1	[0,5[	1	1	1	1	1
maximo	20	31	519	[30,inf[	24	28	13	2	2
miniclust_1	21	31	519	[30,inf[	24	28	13	2	2
miniclust_2	20	32	519	[30,inf[	24	28	13	2	2
miniclust_3	20	31	520	[30,inf[	24	28	13	2	2
miniclust_4	20	31	519	Indef	24	28	13	2	2
miniclust_5	20	31	519	[30,inf[	25	28	13	2	2
miniclust_6	20	31	519	[30,inf[	24	29	13	2	2
miniclust_7	20	31	519	[30,inf[	24	28	14	2	2
miniclust_8	20	31	519	[30,inf[	24	28	13	3	2
miniclust_9	20	31	519	[30,inf[	24	28	13	2	3
miniclust_10	20	31	519	[30,inf[	24	28	13	2	2
max_NClips	20	1	1	[30,inf[	1	1	1	1	1
max_NCat	1	31	1	[30,inf[	1	1	1	1	1
max_NCliq	1	1	519	[30,inf[	1	1	1	1	1
max_TEC	1	1	1	[0,5[	1	1	1	1	1
max_NIP	1	1	1	[30,inf[	24	1	1	1	1
max_NCookies	1	1	1	[30,inf[	1	28	1	1	1
max_NSessoes	1	1	1	[30,inf[	1	1	13	1	1
max_NBrowser	1	1	1	[30,inf[	1	1	1	2	1
max_NLocal	1	1	1	[30,inf[	1	1	1	1	2

Tabela 6.1: Instâncias redefinidas

de dados produzido estas nove instâncias.

Como se pode constatar pela Tabela 6.2, onde se encontram expostas as pontuações (Pont) e posições (Pos) obtidas, e que a negrito se realçam as instâncias que possuem a pontuação mais elevada, uma das instâncias que obteve maior pontuação na maioria dos métodos foi a *max_NCliq*. Assim, é possível verificar que o atributo responsável pela informação relativa ao número de cliques efetuados pelos visitantes possui um maior peso comparativamente aos restantes atributos. É também possível constatar que o segundo elemento com mais influência é o atributo que regista o número de *cookies* utilizados. O terceiro atributo mais influente revelou-se ser o responsável pela quantificação do número de endereços IP utilizados pelo visitante. Por outro lado, verifica-se que a instância destinada a verificar a influência do atributo tempo é a menos influente na maioria dos casos. Verifica-se

IDSessão	LOF		COF		LOCI		LoOP		INFLO		KNN-avg		KNN-kth		CBLOF	
	Pont	Pos	Pont	Pos	Pont	Pos	Pont	Pos	Pont	Pos	Pont	Pos	Pont	Pos	Pont	Pos
max_NClips	2.3	13	1.5	72	1.5	41	0.6	20	2.6	10	9.4	28	11.4	28	13.3	12
max_NCat	2.5	11	1.9	14	1.5	50	0.8	10	2.8	8	13.3	14	16.1	21	**21.8**	**2**
max_NCliq	10.2	3	**4.8**	**2**	**7.8**	**2**	**1.0**	**3**	9.7	3	**358.3**	**2**	**433.0**	**2**	0.1	7516
max_TEC	1.0	2801	1.6	39	0.0	7516	0.1	460	1.0	3415	1.0	1040	1.0	2389	1.6	5454
max_NIP	6.0	5	3.1	5	1.9	26	0.9	7	6.1	5	21.6	10	25.1	14	19.8	4
max_NCookies	**11.3**	**1**	3.5	3	5.0	4	**1.0**	**1**	**11.0**	**1**	21.2	11	22.0	16	16.9	5
max_NSessoes	1.2	331	1.4	279	0.7	367	0.1	650	1.2	281	1.0	1041	1.0	2390	8.5	155
max_NBrowser	1.1	378	1.6	32	0.1	1148	0.3	87	1.2	233	1.4	515	1.4	958	1.8	3581
max_NLocal	1.1	379	1.6	33	0.1	1147	0.3	88	1.2	234	1.4	516	1.4	959	1.8	3582

Tabela 6.2: Pontuação atribuída pelos métodos selecionados às instâncias *max_[atributo]*

também que os atributos referentes ao número de *browsers* e de locais a partir dos quais foram efetuadas as visitas obtiveram pontuações muito semelhantes.

As pontuações atribuídas pelo método CBLOF foram as mais díspares comparativamente aos restantes. Sendo que este método necessita de um algoritmo de *clustering*, foi utilizado o *X-Means*, apresentado em [Pelleg and Moore, 2000], uma vez que este não necessita de uma definição prévia do número de *clusters*, sendo capaz de definir o número ótimo de sub-conjuntos. Assim, estas pontuações podem ser consequência dessa mesma divisão. Após a execução do algoritmo *X-Means* foi possível constatar que foram criados quatro sub-conjuntos, com as seguintes dimensões: C_0 - 7324 itens; C_1 - 190 itens; C_2 e C_3 - 1 item. Relembre-se que a fórmula de cálculo do grau de anomalia utilizado pelo CBLOF utiliza a dimensão dos *clusters* como um fator de peso associado à distância. Neste caso, o único elemento de C_2 ou C_3 terá um peso de 1 associado à sua distância ao maior *cluster* vizinho. No entanto, os elementos pertencentes a C_0 obterão um peso associado à sua distância ao centróide de 7324. Considere-se o seguinte exemplo. Seja $p \in C_0$ e $q \in C_2$. Supondo que $d(p,C_0) = 1$ e que $d(q, C_0) = 100$, pela fórmula da equação 4.3, p obterá uma pontuação de 7324 e q de 100, o que está errado. Em [Amer, 2011] é apresentada uma solução para este problema que passa pela remoção dos pesos de cada *cluster*. Os novos resultados obtidos após a execução desta abordagem podem ser visualizados na Tabela 6.3

IDSessão	CBLOF	
	Pont	Pos
max_NClips	18.2	172
max_NCat	29.8	72
max_NCliq	**516.3**	**2**
max_TEC	2.2	5639
max_NIP	23.1	115
max_NCookies	27.1	90
max_NSessoes	11.7	347
max_NBrowser	2.5	3767
max_NLocal	2.5	3768

Tabela 6.3: Pontuação atribuída pelo método CBLOF sem pesos

IDSessão	LOF		COF		LOCI		LoOP		INFLO		KNN-avg		KNN-kth		CBLOF	
	Pont	Pos	Pont	Pos	Pont	Pos	Pont	Pos	Pont	Pos	Pont	Pos	Pont	Pos	Pont	Pos
media	1.1	717	1.1	1586	0.0	7509	0.1	1315	1.1	1212	0.9	1283	1.0	2383	1.7	6759
frequente	1.0	2580	1.5	44	0.0	7510	0.1	644	1.0	2299	1.0	1033	1.0	2384	2.2	5633
minimo	1.0	5478	1.6	25	0.0	7511	0.2	334	1.0	2560	1.0	1034	1.0	2385	2.6	3556
maximo	**10.2**	**1**	**5.1**	**1**	**7.8**	**1**	**1.0**	**1**	**9.8**	**1**	**365.1**	**1**	**436.0**	**1**	**518.8**	**1**

Tabela 6.4: Pontuação atribuída pelos métodos selecionados às instâncias com *IDSessão media, frequente, minimo e maximo*

Como é possível verificar, a remoção dos pesos associados a cada *cluster* originou resultados muito mais aceitáveis e que vão de encontro aos produzidos pelos restantes métodos. Por este motivo, decidiu-se utilizar esta nova abordagem em detrimento da anterior.

Aplicando os diversos métodos sobre o conjunto de dados que inclui as instâncias *media, frequente, minimo e maximo*, foram obtidos os resultados apresentados na Tabela 6.4.

Tal como esperado, a instância que contempla os valores máximos de cada atributo foi considerada pelas várias abordagens como sendo o elemento mais anómalo de todo o conjunto de dados. Embora a instância denominada como *minimo* possua valores estremados tal como a *maximo*, a grande parte dos restantes objetos pertencentes ao conjunto de dados possui valores também baixos, como visto anteriormente. Por este motivo, as pontuações atribuídas a esta instância serão substancialmente mais baixas comparativamente ao registo *maximo*. Relativamente às instâncias *media* e *frequente*, a maioria dos métodos atribui-lhe uma pontuação

IDSessão	LOF		COF		LOCI		LoOP		INFLO		KNN-avg		KNN-kth		CBLOF	
	Pont	Pos	Pont	Pos	Pont	Pos	Pont	Pos	Pont	Pos	Pont	Pos	Pont	Pos	Pont	Pos
miniclust_1	1.0	5480	0.7	7460	3.9	3	0.0	7508	1.0	2478	5.9	40	47.1	9	518.9	5
miniclust_2	1.0	5801	0.7	7461	3.9	4	0.0	7509	1.0	2445	6.0	38	47.3	7	518.9	2
miniclust_3	1.0	3800	0.7	7462	3.9	5	0.0	7510	1.0	6075	5.9	45	46.8	13	519.8	1
miniclust_4	1.0	3801	0.7	7463	3.9	6	0.0	7511	1.0	6076	5.9	46	46.8	14	518.8	9
miniclust_5	1.0	5802	0.7	7464	3.9	7	0.0	7512	1.0	2446	6.0	39	47.3	8	518.9	4
miniclust_6	1.0	5878	0.7	7465	3.9	8	0.0	7513	1.0	2300	6.0	37	47.4	6	518.9	3
miniclust_7	1.0	4530	0.7	7466	3.9	9	0.0	7514	1.0	2482	5.9	41	47.1	10	518.8	6
miniclust_8	1.0	3803	0.7	7467	3.9	10	0.0	7515	1.0	6058	5.9	42	46.8	11	518.8	7
miniclust_9	1.0	3804	0.7	7468	3.9	11	0.0	7516	1.0	6059	5.9	43	46.8	12	518.8	8
miniclust_10	1.0	3798	0.7	7469	3.9	12	0.0	7517	1.0	6097	5.9	51	46.8	15	518.8	10

Tabela 6.5: Pontuação atribuída pelos métodos selecionados às instâncias pertencentes ao mini-*cluster*

mais reduzida. Isto pode ser explicado pelo facto de, embora os seus valores sejam frequentes, a combinação dos diferentes atributos pode não ser muito comum. Em relação às posições obtidas por estas instâncias com o método CBLOF, estas foram muito mais baixas. Isto poderá ser consequência do facto de o algoritmo *X-Means* definir os seus centróides com base na média de cada *cluster*. Como visto anteriormente, o algoritmo de *clustering* utilizado definiu um grande subconjunto contendo mais de 90% dos registos e onde foram inseridas as instâncias *media* e *frequente*. Assim, era esperado que as pontuações fossem baixas uma vez que a distância destas ao contróide é reduzida.

Após a aplicação dos métodos selecionados sobre o conjunto de dados, ao qual foram adicionadas as instâncias responsáveis por formar um *cluster* de pequenas dimensões, foram obtidos os resultados apresentados na Tabela 6.5.

Analisando as pontuações atribuídas pelo algoritmo LOF, é possível verificar que este método, tal como esperado, não é indicado para a deteção de *micro-clusters*, uma vez que a pontuação atribuída indica que estas instâncias são normais, isto é, $LOF \approx 1$. Este resultado pode ser explicado pelo facto de ter sido utilizado um valor de $k = 10$, ou seja, o método LOF efetuou os seus cálculos com base na densidade de todos os restantes nove elementos do *cluster* criado artificialmente e uma outra instância real que possui um número de cliques se-

melhante, sendo considerada como vizinha pelo algoritmo. Se fosse utilizado um valor de k superior, este algoritmo iria necessitar de analisar instâncias fora deste subconjunto, o que iria aumentar a sua pontuação. Semelhantes resultados foram produzidos pelos algoritmos COF, LoOP e INFLO.

Quanto ao algoritmo KNN-avg, embora as posições sejam elevadas, a sua pontuação é relativamente baixa em comparação com o top-10. Se este algoritmo for executado com $k = 9$ a sua pontuação seria muito mais baixa (aproximadamente 1) uma vez que não seria necessária a análise de um elemento exterior ao *micro-cluster*.

Do mesmo modo, o método que utiliza como forma de medição do grau de anomalia a distância ao k-ésimo vizinho apresenta valores elevados para estas instâncias. Este resultado está diretamente dependente do valor de k escolhido. Tal como no método anterior, com a utilização de $k = 9$ a pontuação atribuída baixa para valores 1.4, aproximadamente, e a sua posição no *ranking* dos mais anómalos ronda a 900.

Para este teste, os melhores resultados foram obtidos através da execução dos algoritmos LOCI e CBLOF. Neste último, as instâncias foram classificadas como sendo as mais anómalas em todo o *dataset*. No caso do LOCI, e tal como esperado, estas foram também identificadas como das instâncias mais anómalas do conjunto.

Após esta análise sobre o comportamento dos diversos algoritmos sobre as instâncias criadas, foi procedida a aplicação destes sobre o conjunto de dados original, com o propósito de detetar anomalias reais bem como analisar qual ou quais dos métodos selecionados produzem resultados mais interessantes tendo em conta o contexto em que o problema se insere.

Tendo em conta que os atributos com mais influência foram os responsáveis por identificar o número de cliques e de *cookies* usados em cada sessão, as figuras a apresentar de seguida utilizarão como eixos estes dois parâmetros, onde a di-

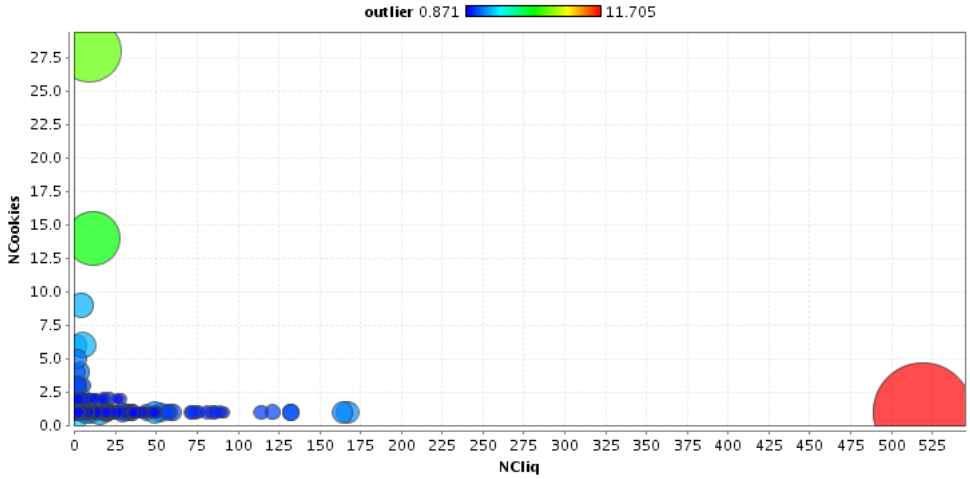


Figura 6.2: Pontuações atribuídas pela abordagem LOF a cada uma das instâncias

NClips	NCat	NCliq	NIP	NCookies	NSessões	NBrowser	NLocal	TEC	Pontuação
4	8	519	1	1	1	1	1	[30,inf[	11.7
6	1	9	1	28	7	1	1	[30,inf[	7.3
1	0	11	1	14	1	1	1	[30,inf[	6.6
13	4	1	1	1	10	1	1	Indef	3.2
1	2	5	1	6	1	1	1	[30,inf[	2.8

Tabela 6.6: Instâncias mais anómalas segundo LOF

mensão e cores dos objetos representarão o grau de anomalia atribuído às diversas instâncias pelo algoritmo em questão. Estas figuras foram geradas recorrendo ao *software* RapidMiner [Mierswa et al., 2006].

Na Figura 6.2 estão ilustradas as pontuações atribuídas pelo método LOF. Como é possível verificar, esta abordagem destacou claramente uma instância (a vermelho na figura) como sendo a mais anómala em relação aos restantes elementos do conjunto de dados. A referida instância, ou sessão, apesar de possuir valores bastante frequentes, possui 519 cliques efetuados, tornando-a claramente anómala neste contexto, uma vez que em média um utilizador efetua apenas quatro cliques em anúncios por sessão. As duas instâncias seguintes, a verde na figura, representam sessões que apesar de possuírem 9 e 11 cliques, apresentam um número suspeito de *cookies*. À primeira sessão foram atribuídos 28 *cookies* e o utilizador

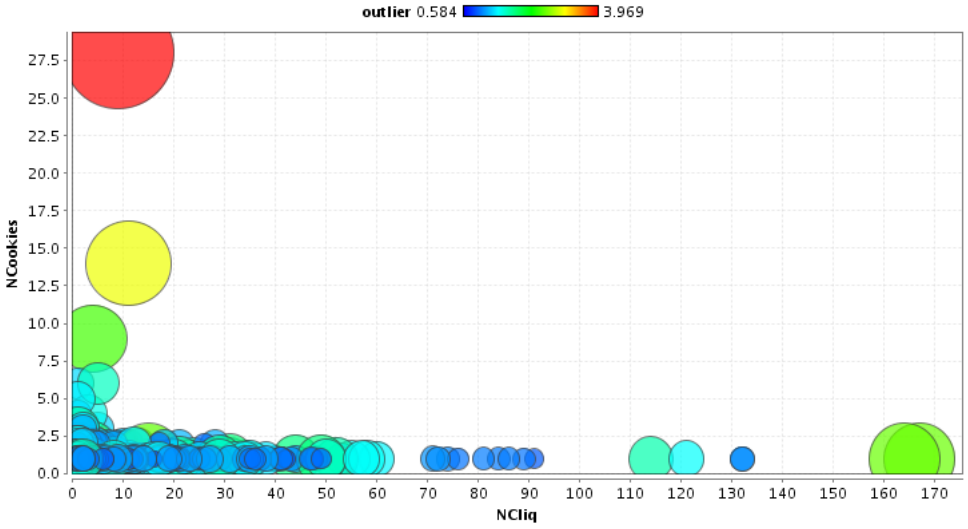


Figura 6.3: Pontuações atribuídas pela abordagem COF a cada uma das instâncias

NClips	NCat	NCliq	NIP	NCookies	NSessões	NBrowser	NLocal	TEC	Pontuação
4	8	519	1	1	1	1	1	[30,inf[	9.0
6	1	9	1	28	7	1	1	[30,inf[	4.0
1	0	11	1	14	1	1	1	[30,inf[	3.1
13	4	1	1	1	10	1	1	Indef	3.1
1	2	167	24	1	1	1	1	[30,inf[	2.6

Tabela 6.7: Instâncias mais anómalas segundo COF

já tinha visitado sete vezes o *site*. A segunda apresentava 14 *cookies*. Embora os seus comportamentos não sejam especialmente graves para o sistema, isto é, não efetuem um elevado número de cliques que possam prejudicar/beneficiar alguma das partes, os visitantes apresentam um comportamento desviante do normal uma vez que parecem demonstrar interesse em não serem "reconhecidos". As cinco instâncias mais anómalas segundo LOF encontram-se na Tabela 6.6.

Sendo que todos os métodos atribuíram a maior pontuação à instância que possui 519 cliques e com o propósito de facilitar a compreensão visual dos graus de anomalia dos registos menos anómalos, nas figuras seguintes foi removida esta instância de forma a que a escala das figuras pudesse ser adequada.

Estas três últimas sessões foram também destacadas como sendo as mais anó-

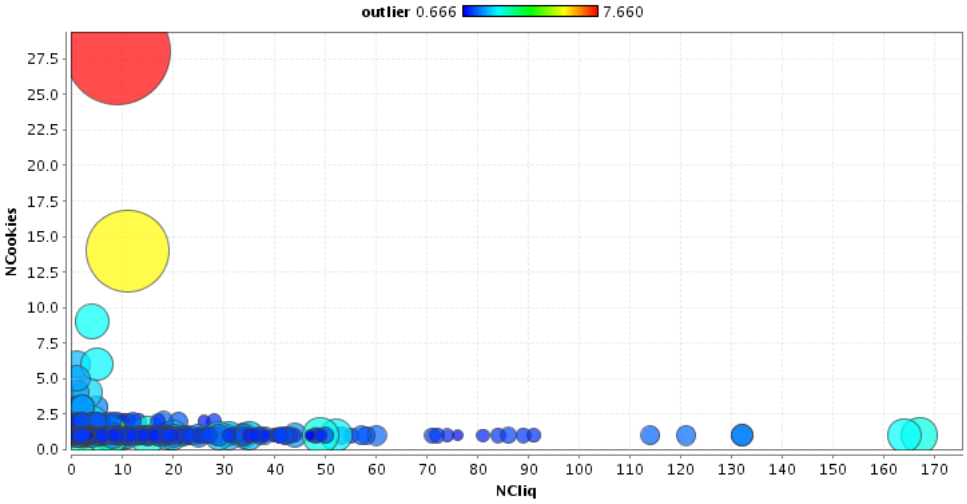


Figura 6.4: Pontuações atribuídas pela abordagem INFLO a cada uma das instâncias

NClips	NCat	NCliq	NIP	NCookies	NSessões	NBrowser	NLocal	TEC	Pontuação
4	8	519	1	1	1	1	1	[30,inf[	11.4
6	1	9	1	28	7	1	1	[30,inf[	7.7
1	0	11	1	14	1	1	1	[30,inf[	5.9
13	4	1	1	1	10	1	1	Indef	3.1
20	2	7	1	1	1	1	1	[30,inf[	2.9

Tabela 6.8: Instâncias mais anómalas segundo INFLO

malas pelos métodos COF e INFLO, como é possível visualizar nas Figuras 6.3 e 6.4 e respetivas Tabelas 6.7 e 6.4. Todos os três métodos anteriores classificam também como anómala uma sessão que possui apenas um clique, mas que visitou o site 10 vezes, recorrendo a 13 clips diferentes.

As pontuações atribuídas pela abordagem LoOP (Tabela 6.9) divergem ligeiramente das referidas anteriormente. Além do mesmo top-3 apresentado anteriormente, nesta abordagem é dada grande importância a instâncias que possuem várias visitas (*NSessoes*). A quarta instância mais anómala segundo esta abordagem é uma sessão que possui 49 cliques, tendo o utilizador visitado o site 12 vezes anteriormente, isto é $NSessoes = 12$. Para uma mais fácil perceção dos resultados obtidos através da execução deste algoritmo, considere-se a Figura 6.5.

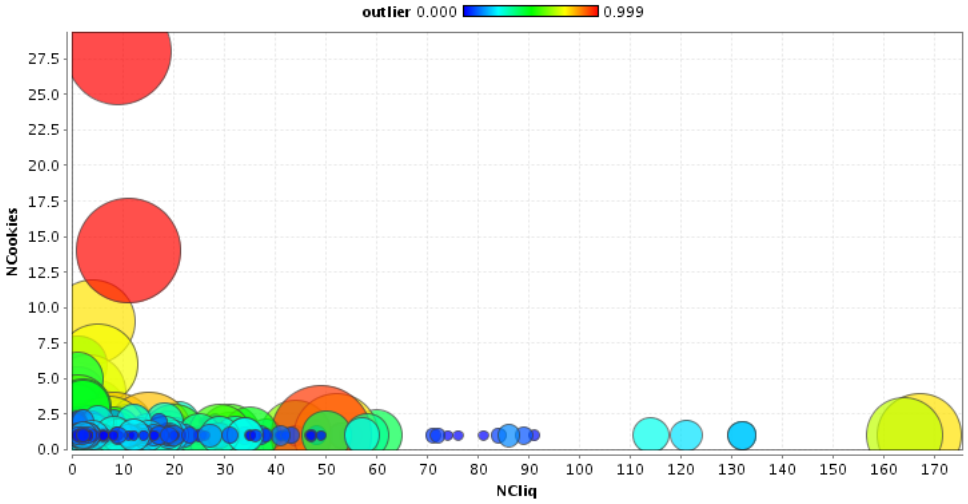


Figura 6.5: Pontuações atribuídas pela abordagem LoOP a cada uma das instâncias

NClips	NCat	NCliq	NIP	NCookies	NSessões	NBrowser	NLocal	TEC	Pontuação
4	8	519	1	1	1	1	1	[30,inf[	1.0
6	1	9	1	28	7	1	1	[30,inf[	1.0
1	0	11	1	14	1	1	1	[30,inf[	1.0
2	1	49	1	1	12	1	1	[30,inf[	0.9
13	4	1	1	1	10	1	1	Indef	0.9

Tabela 6.9: Instâncias mais anómalas segundo LoOP

Estas abordagens apresentam algumas pontuações inadequadas. Sendo que a fraude neste tipo de negócio ocorre através da prática de cliques fraudulentos, o número de cliques deveria ser um indicador realmente importante na pontuação a atribuir a cada sessão. Contudo, as abordagens LOF, COF, INFLO e LoOP falham em casos relativamente graves. Por exemplo, todas estas abordagens consideram muito menos anómala uma sessão que possua 91 cliques do que uma que possua 10 sessões anteriores e apenas 1 clique. Por este motivo, estes métodos mostram-se inadequados para o problema em estudo.

Analisando os resultados produzidos através da aplicação do algoritmo assente em LOCI, é possível constatar através da Figura 6.6 que esta abordagem atribuiu uma maior importância ao número de cliques comparativamente aos métodos an-

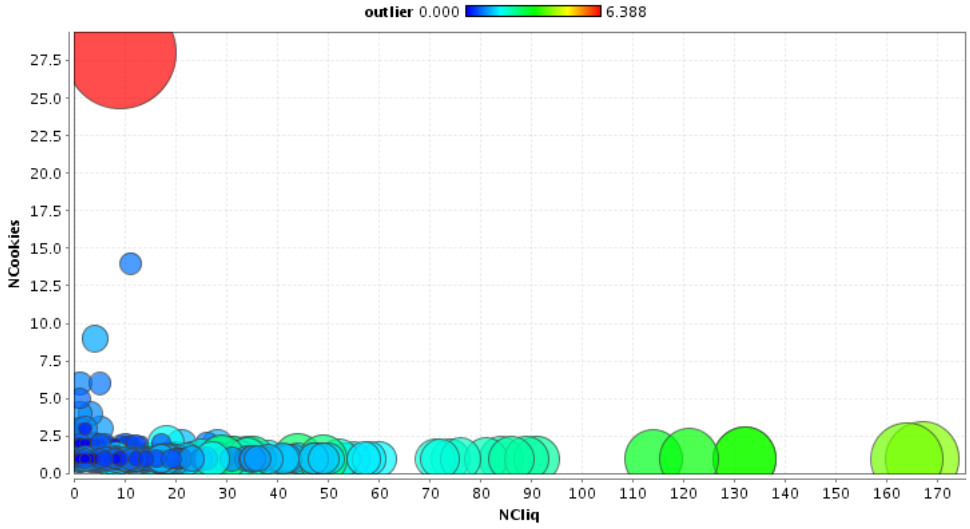


Figura 6.6: Pontuações atribuídas pela abordagem LOCI a cada uma das instâncias

NClips	NCat	NCliq	NIP	NCookies	NSessões	NBrowser	NLocal	TEC	Pontuação
4	8	519	1	1	1	1	1	[30,inf[	9.9
6	1	9	1	28	7	1	1	[30,inf[	6.4
1	2	167	24	1	1	1	1	[30,inf[	4.0
1	2	164	1	1	1	1	1	[30,inf[	3.9
1	1	132	1	1	1	1	1	[30,inf[	3.3

Tabela 6.10: Instâncias mais anómalas segundo LOCI

teriores, sem no entanto desvalorizar a importância do número de *cookies*. Pode-se assim afirmar que esta proposta é uma das mais adequada para a deteção de anomalias no contexto apresentado. Assim, após a execução deste algoritmo com o valor de $\alpha = 0.5$, onde o número mínimo de vizinhos foi definido como $n_{min} = 10$, foram obtidos como mais anómalas as instâncias presentes na Tabela 6.10.

Relativamente ao algoritmo responsável pelo cálculo da distância média de cada instância face aos seus dez vizinhos, esta abordagem revelou atribuir um maior peso ao número de cliques à semelhança do método anterior (ver Tabela 6.11. Para uma mais fácil interpretação dos resultados obtidos, considere-se a Figura 6.7. Assim, além do objeto classificado como sendo o mais anómalo pelos métodos anteriormente apresentados, esta abordagem atribuiu pontuações eleva-

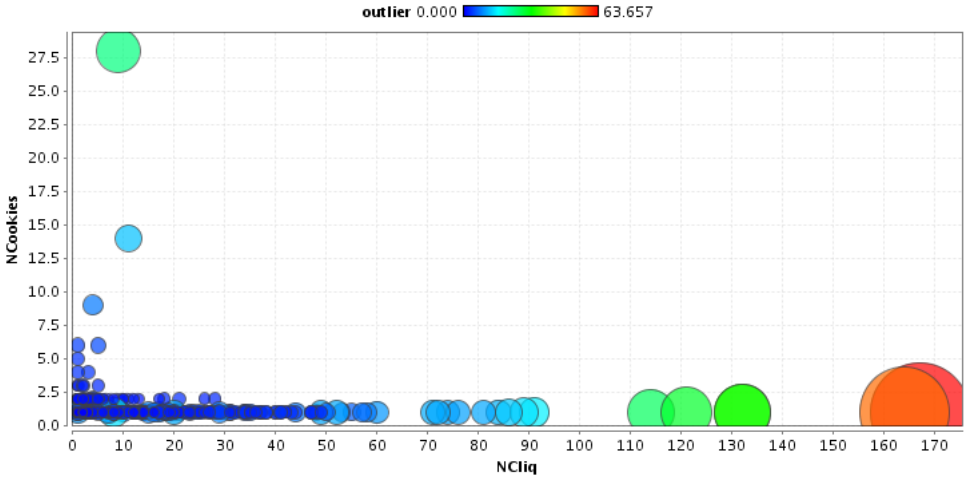


Figura 6.7: Pontuações atribuídas pela abordagem KNN-avg a cada uma das instâncias

NClips	NCat	NCliq	NIP	NCookies	NSessões	NBrowser	NLocal	TEC	Pontuação
4	8	519	1	1	1	1	1	[30,inf[	401.1
1	2	167	24	1	1	1	1	[30,inf[	63.7
1	2	164	1	1	1	1	1	[30,inf[	57.0
1	1	132	1	1	1	1	1	[30,inf[	33.3
1	1	132	1	1	1	1	1	[30,inf[	33.3

Tabela 6.11: Instâncias mais anómalas segundo KNN-avg

das a sessões que possuem entre 114 e 167 cliques. Além disso, na sessão onde foram efetuados 167 cliques, o utilizador mudou de endereço IP 24 vezes, o que poderá indiciar pretensões fraudulentas.

Resultados muito similares foram também obtidos através da execução do método KNN-kth, como é possível verificar pela Tabela 6.12, onde o top-5 obtido é idêntico ao anterior. Relativamente aos restantes elementos, considere-se a Figura 6.8 para uma mais fácil visualização das classificações obtidas.

À semelhança dos métodos LOCI, KNN-avg e KNN-kth, a abordagem assente na medida CBLOF atribuiu ainda maior destaque às sessões que possuem um elevado número de cliques efetuados pelos visitantes, como é possível visualizar na Figura 6.9 e na Tabela 6.13. Mais uma vez, foi retirado o elemento com maior grau de anomalia e foi utilizado o número de categorias como um dos eixos do gráfico.

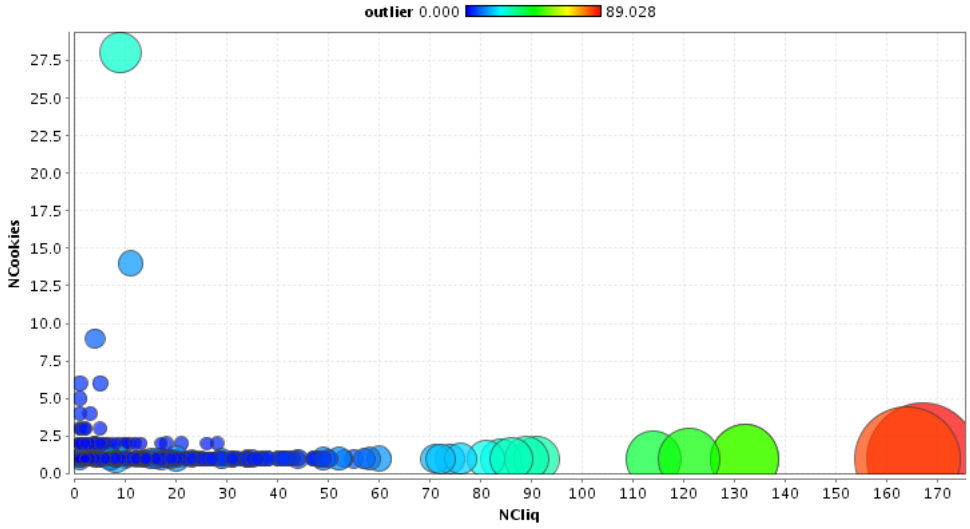


Figura 6.8: Pontuações atribuídas pela abordagem KNN-kth a cada uma das instâncias

NClips	NCat	NCliq	NIP	NCookies	NSessões	NBrowser	NLocal	TEC	Pontuação
4	8	519	1	1	1	1	1	[30,inf[	435.1
1	2	167	24	1	1	1	1	[30,inf[	89.0
1	2	164	1	1	1	1	1	[30,inf[	83.0
1	1	132	1	1	1	1	1	[30,inf[	51.0
1	1	132	1	1	1	1	1	[30,inf[	51.0

Tabela 6.12: Instâncias mais anómalas segundo KNN-kth

Com este método, apenas o 70^{o} objeto mais anómalo foi avaliado por outro atributo que não o número de cliques. A avaliação deste objeto teve como base o número de categorias utilizadas, neste caso 31 categorias, sendo esta sessão que possui mais categorias visitadas. Para a obtenção destes resultados, foi utilizado o algoritmo *X-Means*. Este algoritmo subdividiu o conjunto de dados inicial em quatro *clusters* compostos por 1, 71, 636 e 6799 elementos. Sendo que o algoritmo CBLOF foi executado com os parâmetros $\alpha = 0.9$ e $\beta = 5$, e sabendo que o conjunto de dados possui 7507 instâncias, temos pela inequação 4.1 que $b = 1$, ou seja, o conjunto dos grandes *clusters* é composto apenas pelo maior subconjunto. Assim, o cálculo do grau de anomalia de cada instância presente no *dataset* foi feito medindo a distância ao único grande *cluster*.

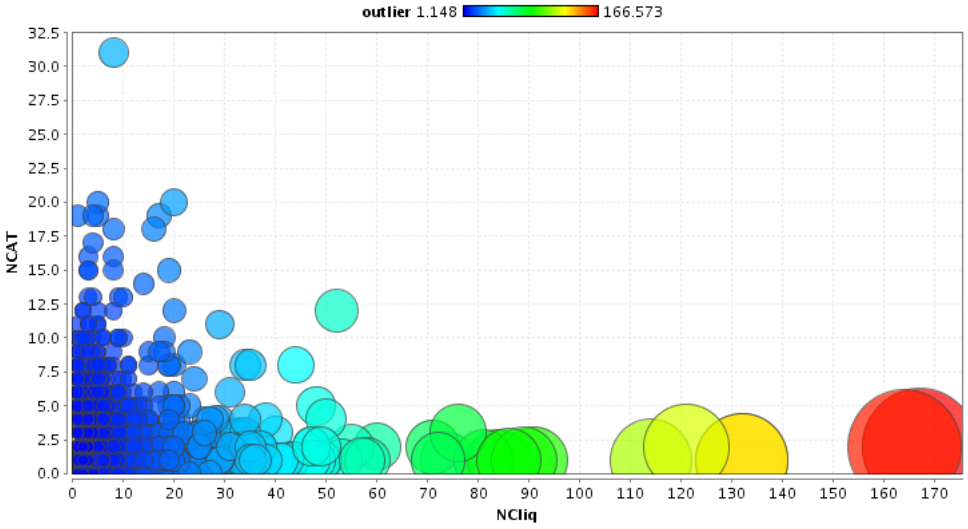


Figura 6.9: Pontuações atribuídas pela abordagem CBLOF a cada uma das instâncias

NClips	NCat	NCliq	NIP	NCookies	NSessões	NBrowser	NLocal	TEC	Pontuação
4	8	519	1	1	1	1	1	[30,inf[	517.0
1	2	167	24	1	1	1	1	[30,inf[	166.6
1	2	164	1	1	1	1	1	[30,inf[	162.0
1	1	132	1	1	1	1	1	[30,inf[	130.0
1	1	132	1	1	1	1	1	[30,inf[	130.0

Tabela 6.13: Instâncias mais anómalas segundo CBLOF

Relativamente ao desempenho apresentado pelos diversos algoritmos selecionados, considere-se a Tabela 6.14. Nesta estão representados aos tempos de execução obtidos, em segundos. Estas execuções foram efetuadas numa máquina com Windows 7 Professional 64-bit SP1, Intel(R) Core(TM) 2 Duo T6670 2.20GHz, 8 GB RAM. Como é facilmente constatável, praticamente todos os métodos processaram todas as instâncias quase instantaneamente, com exceção da abordagem assente na medida LOCI.

Efetuados todos os testes previstos foi possível recolher ilações sobre a viabilidade dos diversos métodos selecionados num contexto de deteção de fraude em *Pay-Per-Click*. Uma das principais conclusões retiradas, após a execução dos testes relativos à deteção de *micro-clusters* anómalos, foi a existência de uma grande

IDSessão	LOF	COF	LOCI	LoOP	INFLO	KNN-avg	KNN-kth	CBLOF
dataset_original	1	1	3375	1	1	1	1	3
dataset_1	1	1	2944	1	1	1	1	3
dataset_2	1	1	3329	1	1	1	1	3
dataset_3	1	1	3319	1	1	1	1	3

Tabela 6.14: Tempo de execução em segundos dos diversos algoritmos sobre as diferentes versões de *datasets*

oscilação das pontuações obtidas tendo em conta o valor de k escolhido. Foi possível constatar que a aptidão para a deteção deste tipo de anomalias estava muito dependente do número de vizinhos considerados, apresentando-se assim como uma grave limitação destes algoritmos. Por esse mesmo motivo, os métodos que não exigem a definição deste valor obtiveram resultados muito mais satisfatórios, neste caso LOCI e CBLOF.

Após a análise dos resultados obtidos através da aplicação dos diferentes métodos sobre o conjunto de dados inicial, foi também possível constatar que os baseados nas medidas LOCI, KNN-kth, KNN-avg e CBLOF, obtiveram melhores resultados. No entanto, os algoritmos KNN-kth e KNN-avg sofrem das limitações já mencionadas, apresentando-se em desvantagem relativamente às abordagens LOCI e CBLOF.

Por último, e observando o tempo necessário para o processamento das diversas instâncias de *datasets* criadas, é possível constatar que o algoritmo LOCI demorou muito mais tempo do que os restantes métodos. Isto poder-se-à revelar num fator de exclusão face ao método CBLOF caso existam restrições temporais relativas ao tempo máximo de execução permitido.

Capítulo 7

Conclusão e Trabalho Futuro

7.1 Conclusão

Ao longo dos capítulos anteriores, foi estudado o modelo publicitário *Pay-Per-Click* bem como os principais problemas inerentes à sua implementação.

Sendo este um tipo de negócio amplamente utilizado, movimenta grandes volumes monetários. Como consequência, este modelo de negócio torna-se um potencial alvo de práticas fraudulentas, sendo necessária uma contínua evolução de mecanismos de defesa contra utilizadores com pretensões menos lícitas.

Como proposto na Secção 1.1, o estado da arte foi apresentado no Capítulo 3, onde foram analisados os problemas inerentes à difusão e expansão da *Internet*, como é exemplo a intrusão em redes de computadores. Neste contexto foram identificadas as principais técnicas utilizadas para a proteção dos sistemas contra este tipo de ataque e estudada a sua viabilidade de aplicação na deteção de fraude num modelo de negócio real assente em *Pay-Per-Click*. Para isso foi aprofundado o estado da arte sobre as diversas abordagens existentes na área de Deteção de Anomalias, a fim de identificar quais as mais adequadas ao problema em questão. Uma vez que em momento algum é possível afirmar inquestionavelmente se um

dado visitante tem pretensões fraudulentas e sendo que os dados disponíveis foram extraídos de uma situação real, não foi possível a criação de um conjunto de dados de treino contendo registos devidamente classificados como normais ou anormais. Deste modo, apenas os métodos não-supervisionados poderiam ser aplicados neste problema. Após esta análise, concluiu-se que as abordagens mais indicadas são as assentes em *Clustering* e baseadas na comparação entre vizinhos, isto é, baseadas em *Nearest-Neigbour*. Assim, foram selecionados oito métodos baseados nestas abordagens para posterior comparação de resultados.

Sendo que os resultados de qualquer processo de extração de conhecimento estão diretamente relacionados com os dados submetidos à análise, foi necessária uma seleção cuidadosa da informação disponível. Uma vez que os dados em posse foram retirados de um contexto real e dada a existência de algumas inconsistências, esta foi uma etapa crítica neste projeto de dissertação, da qual resultou um conjunto de dados devidamente tratados e constituído por dez atributos e 7507 instâncias.

Sendo este um problema não-supervisionado e com o propósito de obter informação adicional sobre o comportamento de cada método selecionado, foi criado e adicionado um conjunto de 23 instâncias heterogéneas ao *dataset* previamente obtido. Deste modo foi possível identificar quais os atributos mais influentes bem como a capacidade dos referidos métodos de lidar com *micro-clusters*. Nesta fase, foi possível verificar a existência de resultados anómalos obtidos através da execução do algoritmo baseado em CBLOF. Assim, foi adotada uma solução proposta em [Amer, 2011], que passa pela remoção dos pesos associados a cada *cluster*. Com esta pequena alteração, foi possível alcançar resultados muito mais adequados ao problema.

Através de uma fase inicial de testes, foi possível concluir que os diversos métodos adotados consideravam como mais influentes os atributos responsáveis pela

quantificação de cliques e *cookies* utilizados em cada sessão. Sendo que é através dos cliques que é praticada a fraude neste sistema, o número de cliques efetuados pelo visitante é um fator fundamental no processo de deteção de fraude. Do mesmo modo, o número de *cookies* pode dar indicações sobre se o utilizador pretende, ou não, ser reconhecido perante o sistema. Em caso negativo, este visitante poderá estar a tentar efetuar novos cliques, fazendo-se passar por um novo visitante, numa tentativa de não levantar suspeitas sobre o seu comportamento. Um outro método típico utilizado para evitar o reconhecimento passa pela alteração do endereço de IP. A instância responsável pelo teste da influência deste fator foi considerada pelas diversas abordagens como a terceira mais anómala.

Foram também realizados testes à capacidade de deteção de *micro-clusters* dos diversos métodos. Nesta fase, as abordagens LOCI e CBLOF destacaram-se pela positiva, atribuindo elevados graus de anormalidade às instâncias pertencentes ao referido subconjunto. Os algoritmos que medem a distância e a média das distâncias (KNN-kth e KNN-avg, respetivamente) também destacaram estas instâncias. No entanto, este destaque revelou-se consequência do valor de k definido, uma vez que em testes realizados com valores de k inferiores, a estas instâncias foi atribuída uma pontuação muito mais baixa. Estes testes permitiram concluir que o desempenho e pontuações atribuídas pelos métodos LOF, COF, LoOP, INFLO, KNN-kth e KNN-avg estão muito dependentes do valor de k escolhido. Sendo que a escolha deste valor não é trivial, esta dependência representa uma desvantagem considerável destes métodos face aos métodos LOCI e CBLOF.

Por último, foram aplicados os diversos métodos selecionados ao conjunto de dados inicial, isto é, sem qualquer instância adicional. Após uma análise realizada aos resultados obtidos, foi possível constatar que os métodos baseados em densidade, com exceção de LOCI, apresentam falhas consideráveis na medição do grau de anomalia de determinadas instâncias, como por exemplo, as referentes a sessões

que tenham realizado entre 70 e 91 cliques, onde lhes foi atribuída uma pontuação inferior à esperada. Neste contexto, os algoritmos que alcançaram melhores resultados foram os baseados nas medidas LOCI, KNN-kth, KNN-avg e CBLOF. Contudo, e como mencionado anteriormente, as medidas KNN-kth e KNN-avg possuem a desvantagem de necessitarem da definição do valor de k e, consequentemente, a sua capacidade de deteção de subconjuntos anómalos estar dependente dessa escolha.

Relativamente ao método LOCI, apesar dos seus resultados bastante satisfatórios, o seu tempo de execução é muito superior ao dos restantes métodos (uma hora comparativamente a um segundo, aproximadamente). Assim, a abordagem que obteve um bom equilíbrio entre tempo de execução e performance de resultados foi o CBLOF.

Aquando da aplicação deste método no contexto real, será necessária uma escolha de qual a forma de seleção dos objetos anómalos. Como apresentado anteriormente, existem duas formas de seleção, nomeadamente, através da definição dos top-N objetos mais anómalos ou através da definição de um valor a partir do qual estes serão considerados como anomalias. No entanto, estas abordagens apresentam um problema. Sendo que o seu resultado se baseia na atribuição de um grau de anormalidade a cada instância, a seleção de quais os objetos anómalos por parte de utilizadores menos inteirados do funcionamento destes algoritmos torna-se um processo complexo. No caso dos dois métodos que apresentaram melhores resultados, nenhum possui um intervalo definido do grau de anomalia a atribuir, pelo que se torna difícil a escolha de um valor mínimo. Por outro lado, se for definido um top-N, poder-se-ão considerar como fraudulentos objetos que possuem pontuações baixas.

Independentemente do método de seleção dos objetos anómalos e com base no estudo realizado, é possível afirmar que a Deteção de Anomalias é uma área

de *Data Mining* capaz de dar um contributo muito significativo no processo de identificação de potenciais defraudadores neste modelo de negócio.

7.2 Trabalho Futuro

Aquando da construção do conjunto de dados, foi assumido que cada sessão representa um novo utilizador. Contudo, esta assunção nem sempre se verifica, como se pode confirmar através do atributo *NSessões*. Assim, e com vista à obtenção de melhores resultados, será necessário um cruzamento dos dados disponíveis a fim de identificar e avaliar o comportamento dos utilizadores, atendendo à sessão atual mas tendo em conta o seu historial.

Como visto anteriormente, diversos algoritmos atribuem elevadas pontuações a instâncias relativamente normais, mas que possuem um valor elevado, como por exemplo o *NSessões*. Com vista à resolução deste problema, seria interessante analisar o comportamento dos diversos métodos se lhes fosse fornecido o peso que cada atributo deveria ter no cálculo do grau de anormalidade de cada instância.

Após esta análise, a implementação de uma solução capaz de automatizar todo o processo descrito anteriormente será um dos objetivos a abordar futuramente. Desta forma será possível dotar o sistema apresentado no caso de estudo, de uma capacidade de deteção de cliques fraudulentos, auxiliando assim a instauração do sentimento de confiança entre os utilizadores deste modelo de negócio.

Bibliografia

Amer, M. (2011). Comparison of unsupervised anomaly detection techniques. Bachelor's thesis, German University in Cairo (GUC).

Angiulli, F. and Pizzuti, C. (2002). Fast outlier detection in high dimensional spaces. In Elomaa, T., Mannila, H., and Toivonen, H., editors, *Principles of Data Mining and Knowledge Discovery*, volume 2431 of *Lecture Notes in Computer Science*, pages 43–78. Springer Berlin / Heidelberg.

Antoniou, D., Paschou, M., Sakkopoulos, E., Sourla, E., Tzimas, G., Tsakalidis, A., and Viennas, E. (2011). Exposing click-fraud using a burst detection algorithm. In *Computers and Communications (ISCC), 2011 IEEE Symposium on*, pages 1111 –1116.

Anupam, V., Mayer, A., Nissim, K., Pinkas, B., and Reiter, M. K. (1999). On the security of pay-per-click and other web advertising schemes. *Comput. Netw.*, 31:1091–1100.

Barbará, D., Li, Y., Couto, J., Lin, J.-L., and Jajodia, S. (2003). Bootstrapping a data mining intrusion detection system. In *Proceedings of the 2003 ACM symposium on Applied computing*, SAC '03, pages 421–425, New York, NY, USA. ACM.

Barbará, D., Wu, N., and Jajodia, S. (2001). Detecting novel network intrusions

using bayes estimators. In *Proceedings of the First SIAM Conference on Data Mining*.

Barnett, V. and Lewis, T. (1994). *Outliers in statistical data*. Wiley series in probability and mathematical statistics: Applied probability and statistics. Wiley & Sons.

Bay, S. D. and Schwabacher, M. (2003). Mining distance-based outliers in near linear time with randomization and a simple pruning rule. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '03, pages 29–38, New York, NY, USA. ACM.

Breunig, M. M., Kriegel, H.-P., Ng, R. T., and Sander, J. (2000). Lof: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, SIGMOD '00, pages 93–104, New York, NY, USA. ACM.

Cannady, J. (1998). Artificial neural networks for misuse detection. In *National information systems security conference*, pages 368–81.

Chandola, V., Banerjee, A., and Kumar, V. (2009). Anomaly detection: A survey. *ACM Comput. Surv.*, 41:15:1–15:58.

Costa, R., de Queiroz, R., and Cavalcanti, E. (2012). A proposal to prevent click-fraud using clickable captchas. In *Software Security and Reliability Companion (SERE-C), 2012 IEEE Sixth International Conference on*, pages 62 –67.

De Stefano, C., Sansone, C., and Vento, M. (2000). To reject or not to reject: that is the question-an answer in case of neural classifiers. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 30(1):84 –94.

Denning, D. (1987). An intrusion-detection model. *Software Engineering, IEEE Transactions on*, SE-13(2):222 – 232.

Diehl, C. and Hampshire, J.B., I. (2002). Real-time object classification and novelty detection for collaborative video surveillance. In *Neural Networks, 2002. IJCNN '02. Proceedings of the 2002 International Joint Conference on*, volume 3, pages 2620 –2625.

Dorronsoro, J., Ginel, F., Sgnchez, C., and Cruz, C. (1997). Neural fraud detection in credit card operations. *Neural Networks, IEEE Transactions on*, 8(4):827 – 834.

Endler, D. (1998). Intrusion detection applying machine learning to solaris audit data. In *Proceedings of the 14th Annual Computer Security Applications Conference*, ACSAC '98, pages 268–, Washington, DC, USA. IEEE Computer Society.

Ester, M., peter Kriegel, H., S, J., and Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. pages 226–231. AAAI Press.

Fain, D. C. and Pedersen, J. O. (2006). Sponsored search: A brief history. *Bulletin of the American Society for Information Science and Technology*, 32(2):12–13.

Fayyad, U., Piatetsky-shapiro, G., and Smyth, P. (1996). From data mining to knowledge discovery in databases. *AI Magazine*, 17:37–54.

Feily, M., Shahrestani, A., and Ramadass, S. (2009). A survey of botnet and botnet detection. In *Emerging Security Information, Systems and Technologies, 2009. SECURWARE '09. Third International Conference on*, pages 268 –273.

Ferreira, P. G., Alves, R., Belo, O., and Cortesão, L. (2006). Establishing fraud detection patterns based on signatures. In *Industrial Conference on Data Mining*, pages 526–538.

Ghosh, S. and Reilly, D. (1994). Credit card fraud detection with a neural-network. In *System Sciences, 1994. Proceedings of the Twenty-Seventh Hawaii International Conference on*, volume 3, pages 621 –630.

Goldstein, M. and Dengel, A. (2012). Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm. In *Stefan Wölfl (ed.) KI-2012: Poster and Demo Track*, pages 59–63.

Guha, S., Rastogi, R., and Shim, K. (1999). Rock: a robust clustering algorithm for categorical attributes. In *Data Engineering, 1999. Proceedings., 15th International Conference on*, pages 512 –521.

Haddadi, H. (2010). Fighting online click-fraud using bluff ads. *SIGCOMM Comput. Commun. Rev.*, 40:21–25.

Han, J. (2005). *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Hawkins, D. M. (1980). Identification of outliers / d. m. hawkins.

He, J., Zhao, K., Hu, L., , N., and Liu, Z. (2010). A time-stamp frequent pattern-based clustering method for anomaly detection. *IETE Technical Review*, 27(3):220–227.

He, Z., Xu, X., and Deng, S. (2003). Discovering cluster-based local outliers. *Pattern Recognition Letters*, 24(9-10):1641 – 1650.

Hollis, N. (2005). Ten years of learning on how online advertising builds brands. *Journal of Advertising Research*, 45(02):255–268.

IAB and PwC (2011). Iab internet advertising revenue report - an industry survey conducted by pwc and sponsored by the interactive advertising bureau (iab).

Jakobsson, M. and Ramzan, Z. (2008). *Crimeware: understanding new attacks and defenses*. Addison-Wesley Professional, first edition.

Jansen, B. (2007). Click fraud. *Computer*, 40(7):85 –86.

Jin, W., Tung, A., Han, J., and Wang, W. (2006). Ranking outliers using symmetric neighborhood relationship. In Ng, W.-K., Kitsuregawa, M., Li, J., and Chang, K., editors, *Advances in Knowledge Discovery and Data Mining*, volume 3918 of *Lecture Notes in Computer Science*, pages 577–593. Springer Berlin / Heidelberg.

Juels, A., Stamm, S., and Jakobsson, M. (2007). Combating click fraud via premium clicks. In *Proceedings of 16th USENIX Security Symposium on USENIX Security Symposium*, pages 2:1–2:10, Berkeley, CA, USA. USENIX Association.

Kantardzic, M., Walgampaya, C., and Emara, W. (2010). Click fraud prevention in pay-per-click model: Learning through multi-model evidence fusion. In *Machine and Web Intelligence (ICMWI), 2010 International Conference on*, pages 20 –27.

Kantardzic, M., Walgampaya, C., Wenerstrom, B., and Lozitskiy, O. (2008). Improving click fraud detection by real time data fusion. *Signal Processing and Information Technology*, pages 69 – 74.

Kemmerer, R. and Vigna, G. (2002). Intrusion detection: a brief history and overview. *Computer*, 35(4):27 –30.

Knorr, E. M. and Ng, R. T. (1998). Algorithms for mining distance-based outliers in large datasets. pages 392–403.

Knorr, E. M. and Ng, R. T. (1999). Finding intensional knowledge of distance-based outliers. In *In VLDB*, pages 211–222.

Kriegel, H.-P., Kröger, P., Schubert, E., and Zimek, A. (2009). Loop: local outlier probabilities. In *Proceedings of the 18th ACM conference on Information and knowledge management*, CIKM '09, pages 1649–1652, New York, NY, USA. ACM.

Kruegel, C. and Vigna, G. (2003). Anomaly detection of web-based attacks. In *Proceedings of the 10th ACM conference on Computer and communications security*, CCS '03, pages 251–261, New York, NY, USA. ACM.

Laxhammar, R. (2011). Anomaly detection in trajectory data for surveillance applications. Licentiate thesis, School of Science and Technology at Örebro University.

Levy, S. (2011). *In the Plex: How Google Thinks, Works, and Shapes Our Lives.* Simon & Schuster.

Li, X., Zeng, D. D., Liu, Y., and Yang, Y. (2011). Click fraud and the adverse effects of competition. *IEEE Intelligent Systems*, 26:31–39.

Metwally, A., Agrawal, D., and El Abbadi, A. (2007a). Detectives: detecting coalition hit inflation attacks in advertising networks streams. In *Proceedings of the 16th international conference on World Wide Web*, WWW '07, pages 241–250, New York, NY, USA. ACM.

Metwally, A., Agrawal, D., and El Abbadi, A. (2007b). On hit inflation techniques and detection in streams of web advertising networks. In *Distributed Computing Systems, 2007. ICDCS '07. 27th International Conference on*, page 52.

Mierswa, I., Wurst, M., Klinkenberg, R., Scholz, M., and Euler, T. (2006). Yale: Rapid prototyping for complex data mining tasks. In Ungar, L., Craven, M., Gunopulos, D., and Eliassi-Rad, T., editors, *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 935–940, New York, NY, USA. ACM.

Papadimitriou, S., Kitagawa, H., Gibbons, P., and Faloutsos, C. (2003). Loci: fast outlier detection using the local correlation integral. In *Data Engineering, 2003. Proceedings. 19th International Conference on*, pages 315 – 326.

Pelleg, D. and Moore, A. W. (2000). X-means: Extending k-means with efficient estimation of the number of clusters. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, pages 727–734, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Ramaswamy, S., Rastogi, R., and Shim, K. (2000). Efficient algorithms for mining outliers from large data sets. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, SIGMOD '00, pages 427–438, New York, NY, USA. ACM.

Reiter, M. K., Anupam, V., and Mayer, A. (1998). Detecting hit shaving in click-through payment schemes. In *Proceedings of the 3rd conference on USENIX Workshop on Electronic Commerce - Volume 3*, WOEC'98, pages 13–13, Berkeley, CA, USA. USENIX Association.

Roth, V. (2004). Outlier detection with one-class kernel fisher discriminants. In *Advances in Neural Information Processing Systems*.

Schölkopf, B., Platt, J. C., Shawe-Taylor, J. C., Smola, A. J., and Williamson, R. C. (2001). Estimating the support of a high-dimensional distribution. *Neural Comput.*, 13(7):1443–1471.

Tang, J., Chen, Z., Fu, A., and Cheung, D. (2002). Enhancing effectiveness of outlier detections for low density patterns. In Chen, M.-S., Yu, P., and Liu, B., editors, *Advances in Knowledge Discovery and Data Mining*, volume 2336 of *Lecture Notes in Computer Science*, pages 535–548. Springer Berlin / Heidelberg.

Tuzhilin, A. (2006). The lane's gifts v. google report.

Zhang, L. and Guan, Y. (2008). Detecting click fraud in pay-per-click streams of online advertising networks. In *Distributed Computing Systems, 2008. ICDCS '08. The 28th International Conference on*, pages 77 –84.