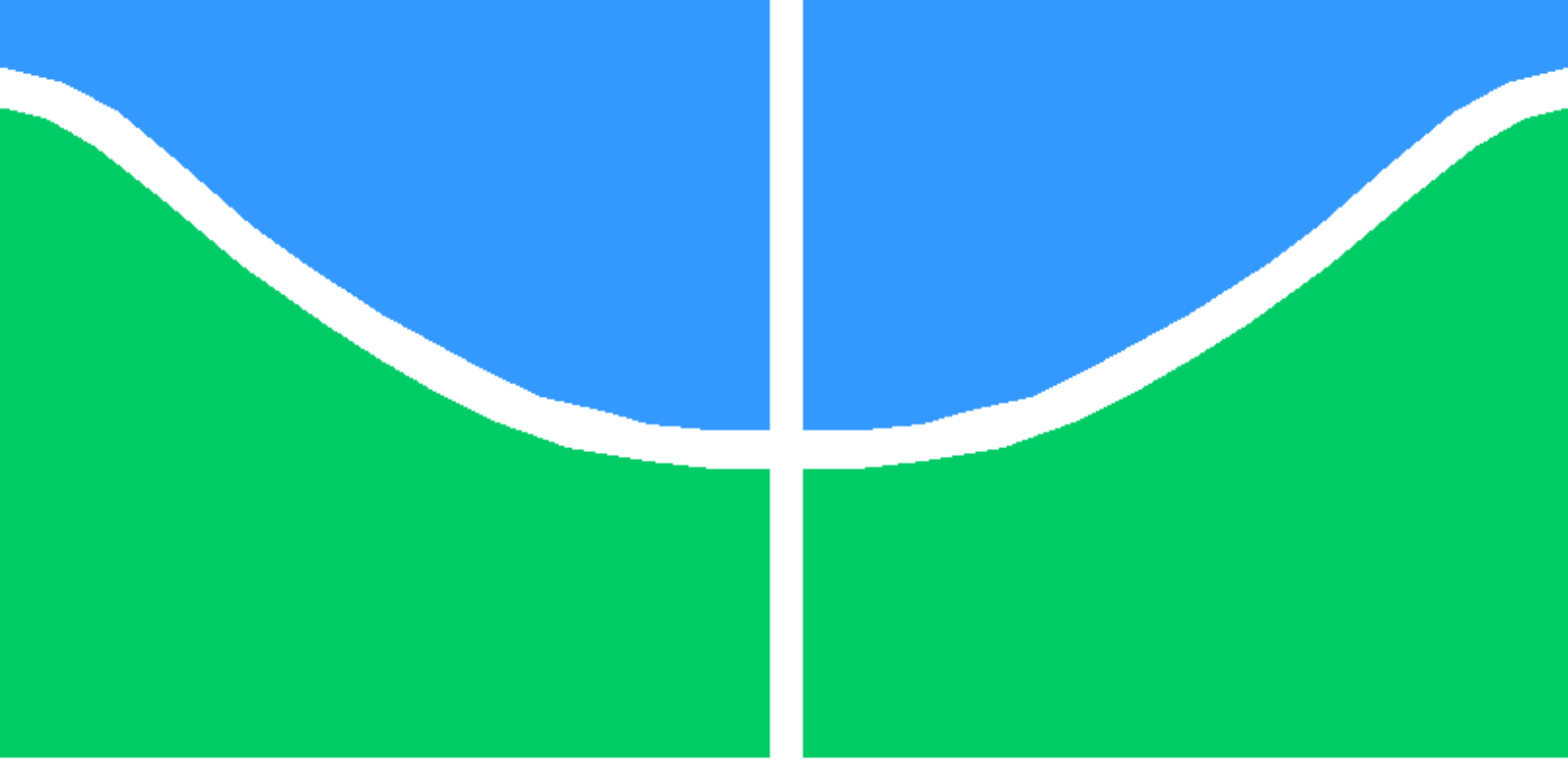


MODELO PARA GERAÇÃO DE LINHAS TEMPORAIS CONTEXTUAIS EM INVESTIGAÇÕES DIGITAIS

REGIS LEVINO DE OLIVEIRA

DISSERTAÇÃO DE MESTRADO EM ENGENHARIA ELÉTRICA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA

UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

MODELO PARA GERAÇÃO DE LINHAS TEMPORAIS CONTEXTUAIS EM INVESTIGAÇÕES DIGITAIS

REGIS LEVINO DE OLIVEIRA

Orientador: DR. BRUNO WERNECK P. HOELZ, POLÍCIA FEDERAL

DISSERTAÇÃO DE MESTRADO EM ENGENHARIA ELÉTRICA

PUBLICAÇÃO PPGENE.DM - 635/2016
BRASÍLIA-DF, 15 DE DEZEMBRO DE 2016.

UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

MODELO PARA GERAÇÃO DE LINHAS TEMPORAIS CONTEXTUAIS EM INVESTIGAÇÕES DIGITAIS

REGIS LEVINO DE OLIVEIRA

DISSERTAÇÃO DE MESTRADO SUBMETIDA AO DEPARTAMENTO DE ENGENHARIA ELÉTRICA DA FACULDADE DE TECNOLOGIA DA UNIVERSIDADE DE BRASÍLIA, COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM ENGENHARIA ELÉTRICA.

APROVADA POR:

Dr. Bruno Werneck P. Hoelz, Polícia Federal
Orientador

Dr. Flavio Elias de Deus, ENE/UnB
Examinador interno

Dr. Paulo Renato da Costa Pereira, Polícia Federal
Examinador externo

BRASÍLIA, 15 DE DEZEMBRO DE 2016.

FICHA CATALOGRÁFICA

REGIS LEVINO DE OLIVEIRA

MODELO PARA GERAÇÃO DE LINHAS TEMPORAIS CONTEXTUAIS EM INVESTIGAÇÕES DIGITAIS

2016, xvii, 62p., 201x297 mm

(ENE/FT/UnB, Mestre, Engenharia Elétrica, 2016)

Dissertação de Mestrado - Universidade de Brasília

Faculdade de Tecnologia - Departamento de Engenharia Elétrica

REFERÊNCIA BIBLIOGRÁFICA

REGIS LEVINO DE OLIVEIRA (2016) MODELO PARA GERAÇÃO DE LINHAS TEMPORAIS CONTEXTUAIS EM INVESTIGAÇÕES DIGITAIS. Dissertação de Mestrado em Engenharia Elétrica, Publicação 635/2016, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 62p.

CESSÃO DE DIREITOS

AUTOR: REGIS LEVINO DE OLIVEIRA

TÍTULO: MODELO PARA GERAÇÃO DE LINHAS TEMPORAIS CONTEXTUAIS EM INVESTIGAÇÕES DIGITAIS.

GRAU: Mestre ANO: 2016

_____

REGIS LEVINO DE OLIVEIRA

QNA 34 CASA 28, TAGUATINGA NORTE, BRASÍLIA - DF

DEDICATÓRIA

À minha família, à minha esposa e aos meus amigos pelo apoio.

AGRADECIMENTOS

Agradeço a Deus em primeiro lugar pela saúde e possibilidade de trabalho.

Agradeço em especial à esposa Taina, que proporcionou esta realização, pelo carinho, atenção, dedicação, incentivo e total apoio nos momentos mais difíceis.

Agradeço aos meus pais, Diogo Gomes de Oliveira e Abigail do Carmo Levino de Oliveira, pelas orações e apoio.

Ao Prof. Dr. Bruno Werneck P. Hoelz pelo imenso apoio dispensado, incentivo, dedicação, autruismo e amizade essenciais para o desenvolvimento deste trabalho.

Agradeço a minha querida tia Lili pelas intensas orações e aos meus irmãos que vibraram para que eu vencesse mais este desafio.

A todos os professores e colegas do Mestrado em Informática Forense, pelo convívio salutar, mútuo auxílio, amizade, que tornaram o mestrado bem prazeroso.

Finalmente, agradeço à Polícia Federal, por intermédio da Diretoria Técnico-Científica, e à Universidade de Brasília, por desenvolverem e apoiarem o projeto de Mestrado Profissional em Engenharia Elétrica com ênfase em Informática Forense e Segurança da Informação, no âmbito do qual esta pesquisa foi desenvolvida, e ao Ministério da Justiça, por fornecer os recursos financeiros necessários ao curso de Mestrado, por meio do Programa Nacional de Segurança Pública com Cidadania – PRONASCI.

Regis Levino de Oliveira

RESUMO

MODELO PARA GERAÇÃO DE LINHAS TEMPORAIS CONTEXTUAIS EM INVESTIGAÇÕES DIGITAIS

Autor: Regis Levino de Oliveira

Orientador: Bruno Werneck P. Hoelz

Programa de Pós-graduação em Engenharia Elétrica

Brasília, dezembro de 2016

Para a elucidação de casos em que o uso de equipamentos digitais está presente, os peritos necessitam realizar a reconstrução dos eventos ocorrida no tempo. Assim, o processo de análise de linhas temporais é uma técnica bastante empregada em exames periciais em ambientes computacionais. No entanto, a maioria dos estudos em linhas temporais concentra-se nos desafios da extração de registros temporais e na normalização desses dados, tratando dos problemas advindos da aquisição de diversas fontes, com menos ênfase em como visualizar e analisar um grande volume desses dados. Este trabalho propõe um modelo para gerar linhas temporais contextualizadas, onde cada rótulo temporal é associado a outras quatro dimensões: local, pessoa, assunto e evento. Um algoritmo de clusterização é então utilizado para gerar linhas temporais com dados similares, que são mais fáceis de visualizar e interpretar. Algoritmos de agrupamento facilitam o descobrimento de novos conhecimentos a partir dos dados analisados. Após obter as linhas temporais contextuais, o perito analisa os dados em conjunto com a linha temporal única, sem contextos, que contém todos os registros temporais extraídos das diversas fontes coletadas, observando os registros que, antes do processo de contextualização, eram mais difíceis de serem observados. Nos resultados obtidos, por meio do estudo de caso, foram obtidas linhas temporais cujos registros apresentam semelhança contextual entre si, reduzindo a interferência de outros registros não relacionados. No experimento proposto, pode-se identificar com mais facilidade os suspeitos com maior interação e os momentos de maior atividade relacionados às condutas investigadas.

ABSTRACT

MODEL FOR THE GENERATION OF CONTEXTUAL TEMPORATIVE LINES IN DIGITAL INVESTIGATIONS

Author: Regis Levino de Oliveira

Supervisor: Bruno Werneck P. Hoelz

Programa de Pós-graduação em Engenharia Elétrica

Brasília, december of 2016

For the elucidation of cases where the use of digital equipment is present, the experts need to perform the reconstruction of the events occurred in time. Thus, the process of analysis of timelines is a technique widely used in expert examinations in computational environments. However, most timeline studies focus on the challenges of extracting temporal records and normalizing these data, addressing the problems of acquiring multiple sources, with less emphasis on how to view and analyze a large volume of such data. This work proposes a model to generate contextualized time lines, where each time label is associated with four other dimensions: location, person, subject and event. A clustering algorithm is then used to generate timelines with similar data, which are easier to visualize and interpret. Grouping algorithms facilitate the discovery of new knowledge from the analyzed data. After obtaining the contextual timelines, the expert analyzes the data in conjunction with the single timeline, without contexts, which contains all the temporal records extracted from the various sources collected, observing the records that, prior to the contextualization process, were more difficult to be observed. In the obtained results, through the case study, temporal lines were obtained whose registers present contextual similarity among themselves, reducing the interference of other unrelated records. In the proposed experiment, it is possible to identify more easily the suspects with greater interaction and the moments of greater activity related to the conducts investigated.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

1 INTRODUÇÃO

Vestígios digitais podem ser a chave para a elucidação de um fato criminoso, como a existência de uma foto (ex.: comprovante de depósito), a troca de mensagens por e-mail, a troca de mensagens por aplicativos de mensageria ou o registro de ações realizadas em um sistema computacional (como logs de acesso). Para auxiliar na compreensão e análise dessa grande diversidade de vestígios, vários recursos de visualização podem ser empregados, dentre eles o da visualização de linhas temporais. A partir da construção de uma linha do tempo, fica mais fácil analisar eventos em torno de pontos de interesse, como o momento em que um servidor de dados foi acessado indevidamente.

Muitas ferramentas estão disponíveis para a geração e análise de linhas temporais. Ferramentas como *Zeitline* [Buchholz e Falk 2005], *CyberForensics TimeLab* [Olsson e Boldt 2009] e *log2timeline* [Guðjónsson 2010] lidam com o desafio de coletar e apresentar os dados temporais em uma única linha temporal.

Elas diferem bastante na forma de exibição e tratamento das informações. Nessas ferramentas, após extração, os dados são dispostos em formato de linha temporal e uma análise de remontagem dos eventos é realizada. A análise de uma linha temporal única acarreta vários problemas que o especialista deve lidar. As diversas fontes de informação, a interpretação de cada sistema para dada estampa de tempo, os desvios no relógio dos dispositivos e a quantidade de material gerado a ser analisada são os problemas mais comuns. Assim, a ocorrência desses diferentes problemas requer o uso de outras abordagens. Nesse sentido, [Chabot et al. 2014a] propõem um modelo para a reconstrução de eventos que inclui definições formais das entidades envolvidas em um incidente e operadores. As definições formais e o uso dos operadores permitem que o conhecimento contido no modelo seja extraído, manipulado e analisado. Já Yu Jin [Jin 2013] propõe o uso de mapas auto-organizáveis para analisar a relação de registros de atividades no sistema operacional Android.

De acordo com as circunstâncias do caso, pode ser necessário visualizar os eventos temporais segundo o contexto a que estão relacionados, reduzindo o ruído associado ao grande volume de registros temporais. Para isso, este trabalho propõe um modelo para a geração de linhas temporais contextuais. Os registros temporais passam por um processo de contextualização, no qual um registro temporal (e seu arquivo de origem) é analisado e associado a entidades previamente definidas nas dimensões pessoa, local, evento e assunto. Posteriormente, algoritmos de agrupamento (clusterização) são aplicados sobre os registros

contextualizados, dos quais resultam linhas temporais que, potencialmente, agrupam eventos relacionados segundo essas entidades. Em seguida, uma análise das linhas temporais contextualizadas é realizada em conjunto com a linha temporal única, que engloba todos os registros temporais coletados.

1.1 Justificativa

Este trabalho propõe um modelo para a geração de linhas temporais contextuais e aplica a teoria de clusterização para realizar o agrupamento de entidades que podem estar associados de acordo com contextos estabelecidos pelo investigador.

Em uma investigação criminal, o uso da técnica de análise dos eventos por meio de uma linha temporal é muito útil para a elucidação de fatos. No entanto, nem sempre há um marco temporal bem definido para ser usado como ponto de partida da análise. Assim, de acordo com as circunstâncias do caso, podemos utilizar como ponto inicial uma pessoa ou grupo de pessoas, uma localização geográfica (uma escola por exemplo), algum evento relevante, ou determinado assunto, como o recebimento de propina, entre outros.

Com a aplicação do agrupamento, o investigador pode visualizar de um modo mais fácil os dados extraídos, associado a uma linha de tempo. Muitos trabalhos foram desenvolvidos para lidar com a extração das estampas de tempo dos dispositivos digitais utilizados nas práticas criminosas. Após a extração, os dados são dispostos em formato de linha temporal e uma análise de remontagem dos eventos é realizada. A análise é feita em uma única linha temporal, tendo muitas dificuldades, como já mencionadas. Em estudos recentes, [Chabot et al. 2014a] propuseram modelos de geração de conhecimento em que a análise dos dados e a geração de conhecimento era o mais automatizado possível, permitindo ao analista utilizar sua expertise para se aprofundar, notadamente, em dados mais refinados.

Com a utilização do modelo proposto neste trabalho, os investigadores poderão dedicar sua expertise nos eventos que possuem algum padrão de relacionamento, utilizando para isso múltiplas linhas temporais que poderão ser criadas a partir do agrupamento (clusterização) segundo outras dimensões como localização, assunto, pessoa, evento, em vez de analisar exclusivamente uma única linha temporal poluída com múltiplos eventos não relacionados entre si, tornando complexo o processo de análise da linha temporal. Conjuntamente com as linhas temporais contextualizadas, o expert analisa a linha temporal única no exame dos registros já filtrados pelas linhas de tempo contextuais.

1.2 Objetivos

Este trabalho tem como objetivo a criação de um modelo para gerar linhas temporais contextuais, onde cada registro temporal é associado a outras quatro dimensões: local, pessoa, assunto e evento. Em seguida, algoritmos de clusterização são então utilizados para gerar linhas temporais com dados similares, que são mais fáceis de visualizar e interpretar. São objetivos específicos:

- desenvolver um modelo de geração de linhas temporais digitais para orientar e auxiliar peritos criminais em investigações criminais digitais durante a reconstrução dos eventos, tomando-se como ponto inicial das investigações pessoas envolvidas, eventos, locais e assuntos;
- desenvolver um estudo de caso em que o modelo proposto possa ser testado e seus resultados possam ser avaliados.

A contribuição deste trabalho é principalmente melhorar a visualização dos eventos quando se utiliza a técnica de análise de linhas temporais, por meio da construção de um modelo de linhas temporais cujas estampas de tempo são analisadas segundo os contextos as quais estão relacionadas. Após a contextualização, os registros são submetidos a um algoritmo de agrupamento *(clustering)* onde o resultado é a formação de linhas temporais contextualizadas, excluindo informações descontextualizadas (ruído) geradas devido ao grande número de dados. A análise de forma exclusiva de uma linha temporal única de várias fontes abrange eventos totalmente desconexos com o caso relacionado (ex.: login em um computador e atualização de software em um celular). A análise contextualizada filtra os ruídos, apresentando ao perito eventos alusivos aos locais, pessoas, assuntos ou eventos mais relacionados aos casos. Os registros filtrados estão disponíveis na linha temporal única, para análise em comparação com as linhas temporais contextualizadas.

1.3 Organização do trabalho

O restante desse trabalho está organizado da seguinte forma: capítulo 2 descreve as referências bibliográficas e trabalhos correlatos e o capítulo 3 apresenta os conceitos teóricos e ferramentas necessários para a aplicação da abordagem proposta; o capítulo 4 apresenta o modelo proposto e o capítulo 5 apresenta os experimentos realizados, mais os resultados obtidos; por fim, o capítulo 6 apresenta as conclusões e trabalhos futuros.

2 INVESTIGAÇÕES DIGITAIS E LINHAS TEMPORAIS

Este capítulo ilustra os conceitos básicos necessários a compreensão do tema tratado nesta dissertação e faz um levantamento das abordagens e mecanismos para a realização da contextualização, da construção das linhas temporais geradas e da reconstrução de eventos utilizado na computação forense.

Computação Forense trata principalmente da investigação de crimes no qual computadores são utilizados [Olsson e Boldt 2009].A Informática Forense consiste de etapas onde os dados relacionados a computadores são coletados, analisados e preservados (cadeia de custódia), a fim de se garantir o valor probatório para a Justiça. [Stacy Jr e Lunsford 2006].

Evidência digital é qualquer informação em formato digital que é transmitida ou que esteja armazenada em algum equipamento eletrônico que possa servir ou tenha valor como prova em uma persecução penal. Muitos tipos de dados podem ser considerados evidências, como, por exemplo, planilhas eletrônicas, agenda de contatos ou de endereços, e-mails, arquivos com gravações em áudio ou em vídeo, arquivos de imagens (inclusive pedofilia), histórico de conversas em programas de comunicação instantânea (chat), entre outros. [Stacy Jr e Lunsford 2006].

Investigação digital, conforme conceitua [Carrier 2005] é um processo onde questões sobre eventos digitais são respondidas por meio de hipóteses devidamente desenvolvidas e testadas. O processo é realizado utilizando-se de métodos científicos onde podem ser desenvolvidas hipóteses utilizando os vestígios e evidencias que são encontrados e então testar a hipótese procurando por evidências que tornam a hipótese impossível de ser realizada.

Para a elucidação crimes envolvendo equipamentos digitais, os investigadores necessitam de metodologias e abordagens que os auxiliem a descobrir o que ocorreu, como ocorreu e quem realizou os eventos em um determinado espaço temporal. Assim, muitos pesquisadores realizaram estudos e métodos que deram origem a análise de vestígios digitais por meio da linha temporal ou *timeline*. Linha temporal é a apresentação de eventos em ordem cronológica, a fim de proporcionar a visualização de um fato ou fatos no decorrer do tempo.

Análise da linha temporal é um processo utilizado por especialistas em informática forense para correlacionar evidências digitais cronologicamente agrupadas, de modo a permitir a extração de informações que elucidem fatos criminosos no mundo real. A análise

por meio da linha do tempo é um componente muito importante de uma investigação de fatos criminosos. Poder afirmar que um evento realmente ocorreu em determinado lugar e a ordem dos acontecimentos pode solucionar um caso ou encurtar bastante o curso de uma investigação. [Guðjónsson 2010].

Para tornar mais fácil a compreensão da importância de se realizar uma análise de linhas temporais, é realizada na próxima seção, uma rápida revisão a respeito dos principais componentes e softwares que integram uma análise de linha de tempo.

2.1 Principais componentes de uma análise de linhas temporais

Para a realização de uma análise de linhas temporais, deve-se entender onde as estampas de tempo *(timestamp)* podem ser coletadas. Pode-se extrair as informações de tempo dos sistemas de arquivos dos sistemas operacionais [Carrier 2005], horários de dentro dos arquivos [Hargreaves e Patterson 2012], arquivos de sistema, logs, *proxy* e *firewall* [Guðjónsson 2010], entre outros.

2.1.1 *Sistemas de arquivos*

O sistema de arquivo é um módulo do sistema operacional responsável por gerenciar as operações sobre os arquivos. A forma como são estruturados, nomeados, acessados, usados, protegidos e implementados é responsabilidade desse módulo [Tanenbaum 2009]. São exemplos de sistemas de arquivos: FAT, FAT32, NTFS, ReFS, EXT3, EXT4, HFS+, UFS, ReiserFS, XFS, ZFS, JFS, etc. O sistema de arquivo é escolhido no momento da formatação do dispositivo de armazenamento. O processo de formatar uma partição ou volume é criar as estruturas de dados usadas pelo sistema de arquivo. Todo arquivo possui muitas informações como nome, data e horário, tamanho do arquivo, entre outros. Essas informações são chamadas de atributos. A tabela 2.1 ilustra um exemplo de possíveis atributos.

Tabela 2.1: Alguns atributos possíveis de arquivos [Tanenbaum 2009]

Atributo	**Significado**
Proteção	Quem tem acesso ao arquivo e de que modo
Senha	Necessidade de senha para acesso ao arquivo
Criador	ID do criador do arquivo
Proprietário	Proprietário atual
Flag de somente leitura	0 para leitura/escrita; 1 para somente leitura
Flag de oculto	0 para normal; 1 para não exibir o arquivo
Flag de sistema	0 para arquivos normais; 1 para arquivos do sistema
Flag de arquivamento	0 para arquivos com backup; 1 para arquivos sem backup
Flag de ASCII/binário	0 para arquivos ASCII; 1 para arquivos binários
Flag de acesso aleatório	0 para acesso somente sequencial; 1 para acesso aleatório
Flag de temporário	0 para normal; 1 para apagar o arquivo ao sair do processo
Flag de travamento	0 para destravados; diferente de 0 para travados
Tamanho do registro	Número de bytes em um registro
Posição da chave	Posição da chave em cada registro
Tamanho do campo-chave	Número de bytes no campo-chave
Momento de criação	Data e hora de criação do arquivo
Momento do último acesso	Data e hora do último acesso do arquivo
Momento da última alteração	Data e hora da última modificação do arquivo
Tamanho atual	Número de bytes no arquivo
Tamanho máximo	Número máximo de bytes no arquivo

Cada sistema de arquivo tem seu formato de registro da estampa de tempo. Na figura 2.1 tem-se um exemplo das informações de tempo no sistema de arquivo NTFS, utilizado em sistemas operacionais da plataforma Windows.

Os sistemas de arquivos guardam muitas informações de data e hora dos arquivos presentes em dispositivos de armazenamento. Estas informações formam os chamados Metadados. Os Metadados podem ser descritos como informações sobre os arquivos e o conteúdo deles, tendo como atributos, por exemplo, o nome, autor, data de criação, data de modificação, entre outros, dos arquivos. No entanto, nem todos os sistemas de arquivos registram as mesmas estampas de tempo, mas costumam ter em comum as datas e horários do último acesso e da última alteração de um arquivo. Os sistemas de arquivos são uma boa fonte de informação para a extração das estampas de tempo. A análise de linha temporal utilizando-se apenas os registros temporais do sistema de arquivo possui alguns problemas. As estampas de tempo dos arquivos podem ser alteradas pela operação normal do sistema operacional, pela execução de algum software antivírus, pela atualização automática dos programas presentes nos dispositivos, pela infecção por softwares maliciosos, etc. Alguns sistemas operacionais possuem a opção de configuração no qual não são gravados as datas e horários do último acesso. Nos sistemas Windows, essa configuração pode ser realizada por meio da chave de registro

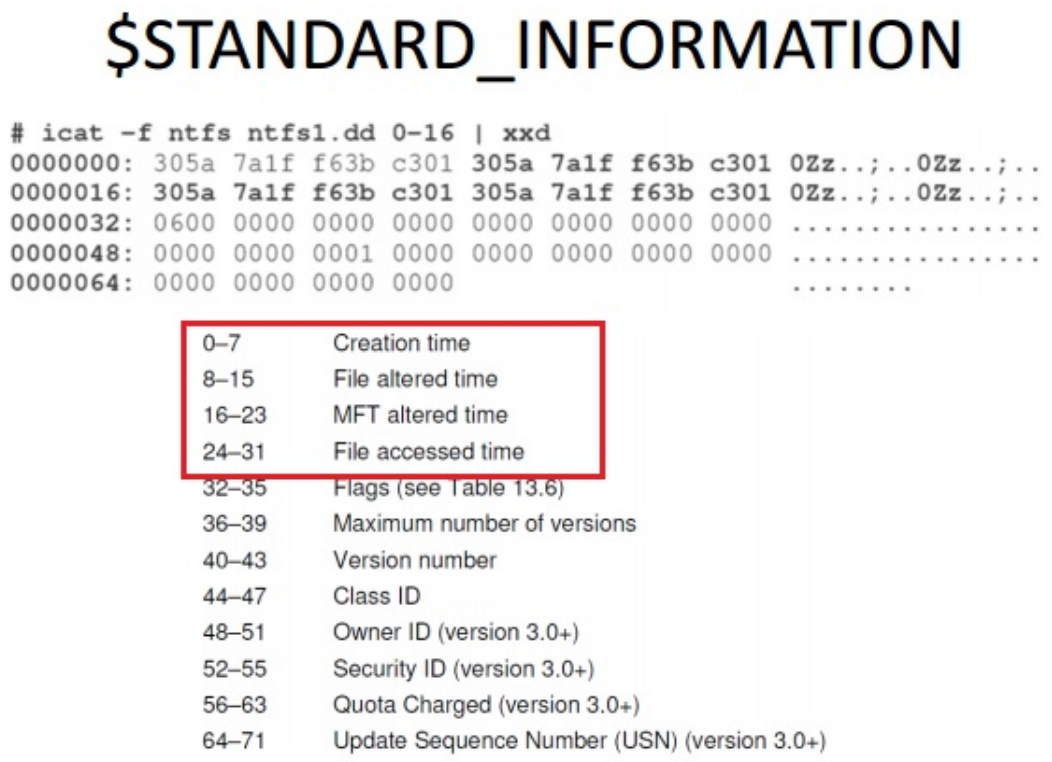


Figura 2.1: Estampas de tempo no sistema de arquivo NTFS [Carrier 2005]

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\FileSystem. No Linux, pode-se realizar a mesma configuração utilizando-se o "fstab". Ferramentas anti-forenses são utilizadas para atrapalhar a análise pelo profissional. Estes softwares são capazes de alterar as estampas de tempo dos sistemas de arquivos e servem para excluir ou incluir informações falsas [Guðjónsson 2010].

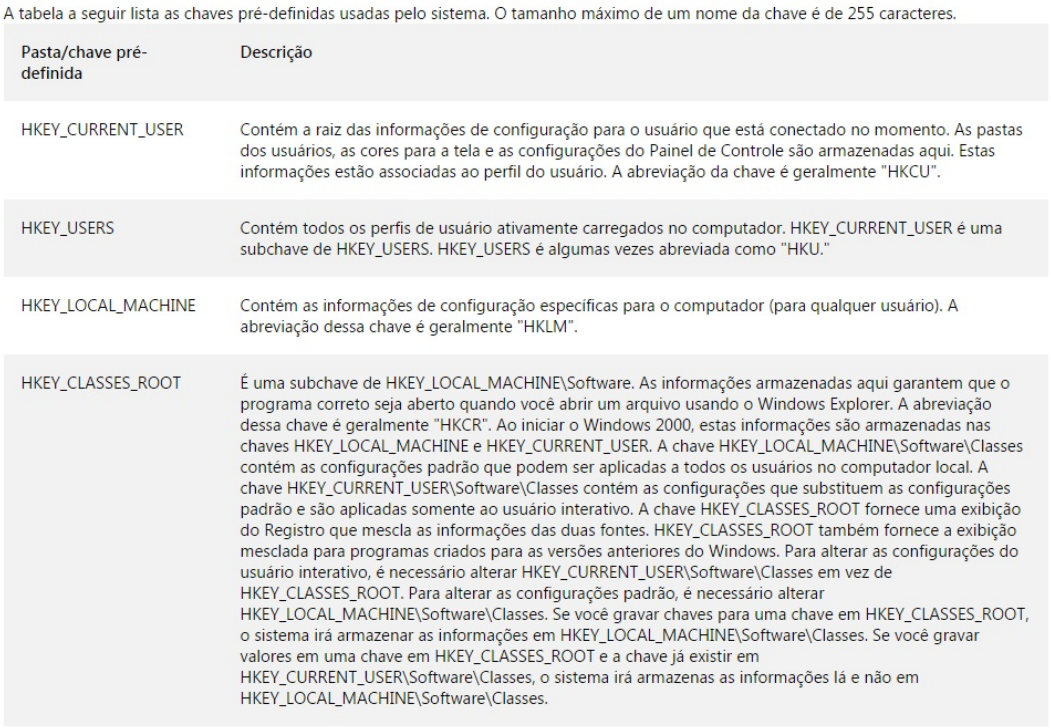
2.1.2 *Registros em Log*

Logs são arquivos que contém informações sobre o comportamento dos sistemas operacionais, seus serviços e atividades. Estes arquivos são utilizados para análise posterior do analista. Os arquivos de *logs* podem gerar informações sobre quais sites na internet um usuário acessa, eventuais falhas no sistema operacional, instalação ou remoção de programas, entre outros.

2.1.3 *Registro do Sistema Windows*

O registro é um banco de dados disposto de forma hierárquica que contém informações necessárias para o funcionamento do sistema operacional, para o funcionamento dos aplicativos nele instalado e para o funcionamento dos dispositivos de hardware. Todas as configurações de aplicativos e drivers de dispositivos instalados no sistema operacional Win-

dows são aí armazenadas. Os registros são conhecidos como “chaves de registro”. Estampas de tempo sobre a instalação, modificação, remoção de software podem ser extraídas do Registro do Windows. A figura 2.2 ilustra as quatro chaves raízes do Registro do Windows.



A tabela a seguir lista as chaves pré-definidas usadas pelo sistema. O tamanho máximo de um nome da chave é de 255 caracteres.

Pasta/chave pré-definida	Descrição
HKEY_CURRENT_USER	Contém a raiz das informações de configuração para o usuário que está conectado no momento. As pastas dos usuários, as cores para a tela e as configurações do Painel de Controle são armazenadas aqui. Estas informações estão associadas ao perfil do usuário. A abreviação da chave é geralmente "HKCU".
HKEY_USERS	Contém todos os perfis de usuário ativamente carregados no computador. HKEY_CURRENT_USER é uma subchave de HKEY_USERS. HKEY_USERS é algumas vezes abreviada como "HKU."
HKEY_LOCAL_MACHINE	Contém as informações de configuração específicas para o computador (para qualquer usuário). A abreviação dessa chave é geralmente "HKLM".
HKEY_CLASSES_ROOT	É uma subchave de HKEY_LOCAL_MACHINE\Software. As informações armazenadas aqui garantem que o programa correto seja aberto quando você abrir um arquivo usando o Windows Explorer. A abreviação dessa chave é geralmente "HKCR". Ao iniciar o Windows 2000, estas informações são armazenadas nas chaves HKEY_LOCAL_MACHINE e HKEY_CURRENT_USER. A chave HKEY_LOCAL_MACHINE\Software\Classes contém as configurações padrão que podem ser aplicadas a todos os usuários no computador local. A chave HKEY_CURRENT_USER\Software\Classes contém as configurações que substituem as configurações padrão e são aplicadas somente ao usuário interativo. A chave HKEY_CLASSES_ROOT fornece uma exibição do Registro que mescla as informações das duas fontes. HKEY_CLASSES_ROOT também fornece a exibição mesclada para programas criados para as versões anteriores do Windows. Para alterar as configurações do usuário interativo, é necessário alterar HKEY_CURRENT_USER\Software\Classes em vez de HKEY_CLASSES_ROOT. Para alterar as configurações padrão, é necessário alterar HKEY_LOCAL_MACHINE\Software\Classes. Se você gravar chaves para uma chave em HKEY_CLASSES_ROOT, o sistema irá armazenar as informações em HKEY_LOCAL_MACHINE\Software\Classes. Se você gravar valores em uma chave em HKEY_CLASSES_ROOT e a chave já existir em HKEY_CURRENT_USER\Software\Classes, o sistema irá armazenas as informações lá e não em HKEY_LOCAL_MACHINE\Software\Classes.

Figura 2.2: Chaves de registro pré-definidas usadas pelo sistema [Microsoft 2012]

2.1.4 *Proxy e Firewall*

O *firewall* é um mecanismo de proteção que controla a passagem de pacotes entre redes, tanto locais como externas. O dispositivo possui um conjunto de regras especificando que tráfego ele permitirá ou negará. Permite a comunicação entre redes de acordo com a política de segurança definida e que são utilizados quando há uma necessidade de que redes com níveis de confiança variados se comuniquem entre si [Stallings e Vieira 2008]. O *firewall* registra uma série de atividades de acordo com configurações pré-definidas, entre outras, a data e a hora dessas atividades. Esses horários podem ser usados para análise em investigações forenses. A figura 2.3 ilustra um exemplo de *firewall* do tipo filtro de pacotes.

Os *proxies* nada mais são do que *firewalls* em nível de aplicação. Os serviços *proxy* são programas aplicativos ou servidores especializados que recebem as solicitações dos usuários e as encaminha para os respectivos servidores reais. Do ponto de vista do cliente, o servidor é o *proxy*; do ponto de vista do servidor, o cliente é o *proxy*. Seu princípio básico de funcionamento está no fato de que este tipo de *firewall* não permite a conexão direta entre as entidades finais da comunicação. A figura 2.4 ilustra um exemplo do funcionamento de um

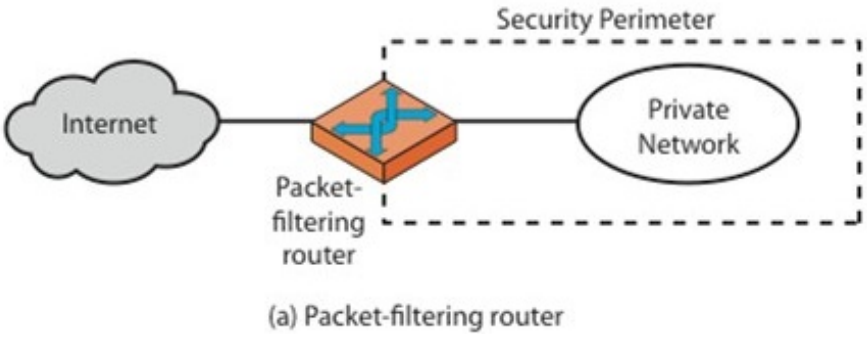


Figura 2.3: Roteador de filtragem de pacotes [Stallings e Vieira 2008]

proxy.

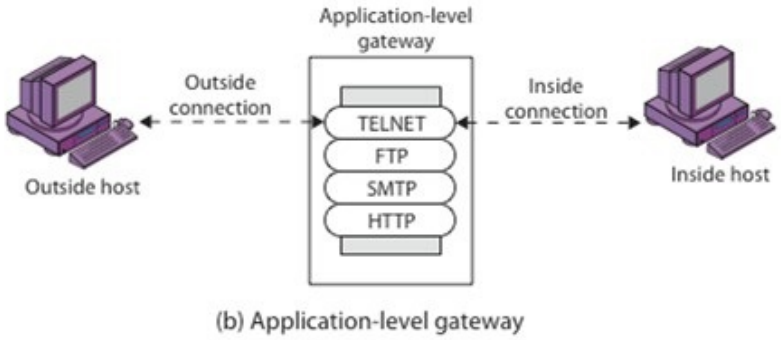


Figura 2.4: Gateway (proxy) em nível de aplicação [Stallings e Vieira 2008]

2.1.5 *Tempos MAC*

Os tempos MAC (ou *MAC times*) referem-se a registros temporais da última atualização em relação a um arquivo e seus atributos. É a última modificação *(mtime)*, acesso *(atime)* e alteração dos metadados *(change)*. Ocorre o *mtime* quando o conteúdo do arquivo é modificado. Ocorre o *atime* quando algum atributo do arquivo ou o conteúdo é acessado. Ocorre o *change* quando apenas os metadados, não o conteúdo do arquivo, é alterado. No Windows com NTFS ou ReFS, cada arquivo tem uma estampa de tempo para *Create, Modify, Access* e *Entry Modified*, também conhecido com o termo abreviado *MACE* [Guðjónsson 2010].

2.1.6 *Estampas de tempo em dispositivos móveis*

Da mesma forma que os assuntos já acima explicados, pode-se realizar a análise de eventos por meio de uma linha de tempo em dispositivos móveis. Vários componentes do sistema de arquivo, do sistema operacional e dos aplicativos instalados trazem informações

úteis para um investigador. Saber em que momento uma mensagem de SMS foi enviada pode ser o ponto chave na determinação do fato ou da autoria de um crime.

2.2 Trabalhos correlatos

Há muitas maneiras de realizar análises de linhas temporais. Muitos pesquisadores desenvolveram metodologias e ferramentas que robustecem uma investigação forense por meio da análise de uma linha do tempo. *Super TimeLine* é um método criado por [Guðjónsson 2010] no qual é proposto a análise da linha temporal por meio de diferentes fontes. Essa forma de análise propõe melhorar o quadro dos fatos para o analista e minimizar a influência das ferramentas anti-forenses. A extensão de uma análise de linha temporal para um *Super TimeLine*, adiciona, para a análise, as informações dos sistemas de arquivos, de logs de eventos do sistema operacional, registro (no caso do sistema operacional Windows), logs das atividades dos sistemas de proteção *firewall, proxy*, sistemas de rede, etc. Dessa forma, espera-se que o investigador tenha uma boa visão geral do dispositivo em análise e possa encontrar rapidamente o vestígio necessário, possibilitando realizar pesquisas mais aprofundadas. A figura 2.5 ilustra a ferramenta gráfica para a entrada de dados.

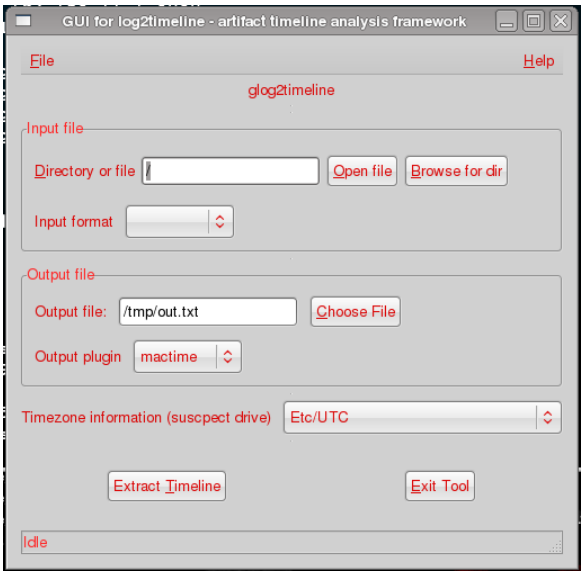


Figura 2.5: Ferramenta gráfica *log2timeline* [Guðjónsson 2010]

O *log2timeline* [Guðjónsson 2010] é uma ferramenta desenvolvida para aplicação do *Super TimeLine* no qual arquivos e diretórios são examinados recursivamente. Esta ferramenta é interessante pois, pela análise de várias fontes como arquivos, informações do sistema de arquivo, logs, firewall, registro do Windows, entre outros, é possível ter uma visão mais ampla, observando ações antes, durante e após a realização dos eventos sob análise. A ferramenta apresenta quatro módulos básicos: Um front-end, bibliotecas compartilhadas,

um módulo de entrada e um módulo de saída. São independentes e desenvolvidos de forma separada. A visão geral de funcionamento da ferramenta é ilustrada na figura 2.6.

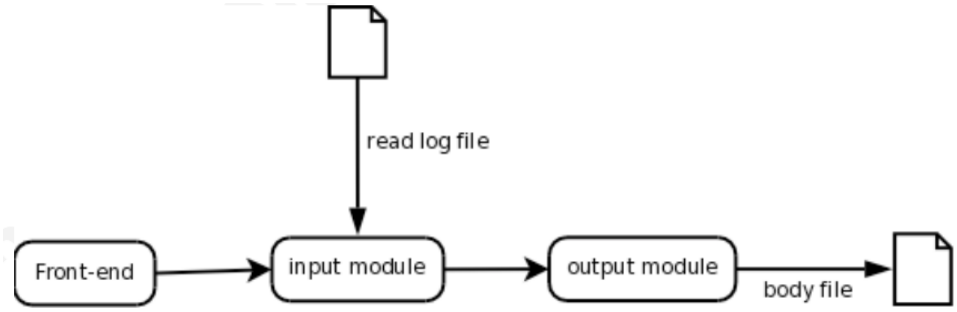


Figura 2.6: Estrutura do log2timeline [Guðjónsson 2010]

Os pesquisadores [Olsson e Boldt 2009] desenvolveram uma técnica para extrair não somente as informações dos metadados, mas também de uma variedade de arquivos como, por exemplo, dados EXIF, arquivos de Link, arquivos MBOX, arquivos do registro do Windows, etc. Para essa técnica, desenvolveram o *CyberForensics TimeLab* (CFTL) [Olsson e Boldt 2009], uma ferramenta que analisa várias fontes, incluindo arquivos em formato JPEG, arquivos do sistema de arquivo, log de eventos do sistema operacional Windows, etc., e em seguida gera o resultado em arquivos XML. Os resultados podem ser visualizados em interface gráfica, os quais são lidos do XML gerado. Os dados são exibidos conforme ilustrado pela figura 2.7.

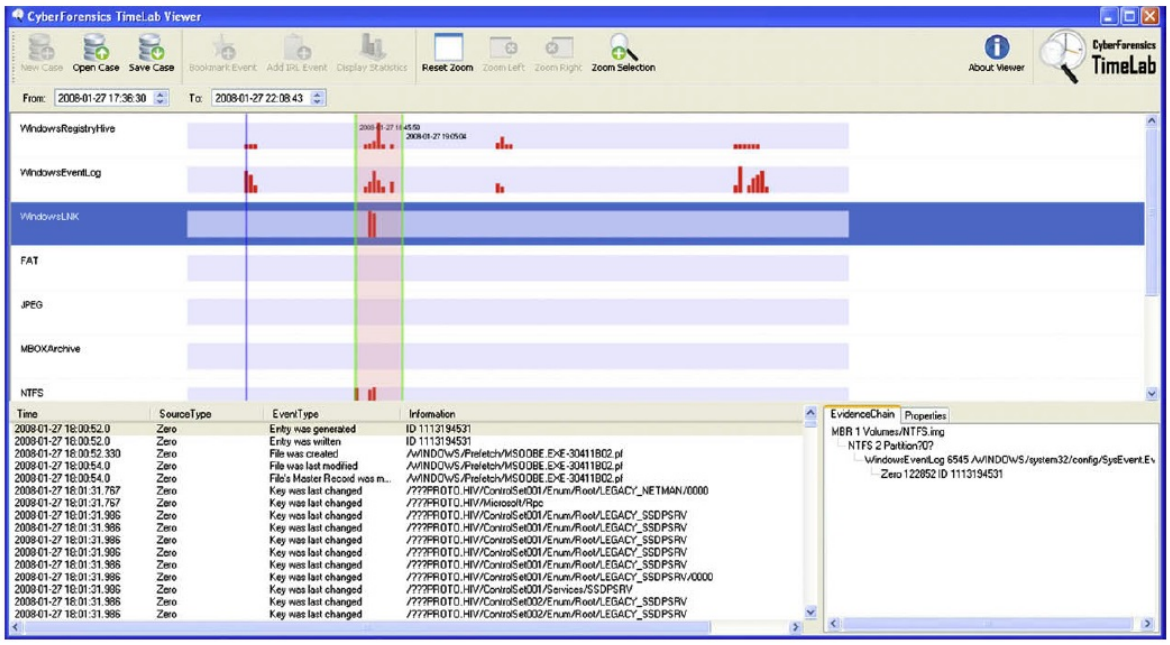


Figura 2.7: Feramenta gráfica *CyberForensics TimeLab* (CFTL) [Olsson e Boldt 2009]

Uma das maiores dificuldades para os analistas forenses que utilizam a técnica de análise de linhas temporais é analisar uma grande quantidade de dados. Assim,

[Hargreaves e Patterson 2012] desenvolveram uma metodologia e ferramenta cujo objetivo é automatizar o processo manual dos peritos, combinando múltiplos eventos de baixo nível em um ou poucos eventos de alto nível, compreensíveis aos humanos. Estampas de tempo dos sistemas de arquivos e dentro dos arquivos são extraídas, formando os eventos de baixo nível. Os eventos são colocados em uma linha de tempo e são armazenados em um banco de dados SQLite para consulta. Os dados brutos são mantidos nesse banco de dados para posterior análise da origem de certa estampa de tempo. Os eventos de alto nível são gerados percorrendo completamente a linha de tempo, em busca do critério especificado. Como os dados brutos ficam armazenados no banco de dados, é possível recuperá-los e verificar a fonte que gerou a estampa de tempo do evento de baixo nível.

Brian Carrier [Carrier 2010] desenvolveu ferramentas que extraem estampas de tempo dos sistemas de arquivos. Essas ferramentas são agora parte do Sleuthkit/Autopsy. Os maiores softwares comerciais para computação forense apresentam módulos que trabalham com linhas temporais. Esses softwares são o FTK [AccessData 2016] e Encase [Encase 2016].

O Zeitline [Buchholz e Falk 2005] é outra ferramenta que coleta informação de várias fontes, mas requer que o especialista adicione, ordene e agrupe as estampas de tempo manualmente para que os dados possam ser analisados. Esta ferramenta trata os eventos em dois tipos: Eventos Atômicos, os quais são importados do sistema, como estampas de tempo MAC, arquivos de log, etc.; e Eventos Complexos, os quais são formados por conjuntos de Eventos Atômicos. A ferramenta lista os eventos como uma árvore, no qual os eventos são exibidos e indexados pelas datas em que ocorreram. Nesta estrutura de árvores, os eventos complexos são exibidos como pais e os eventos atômicos como filhos, conforme ilustrado na figura 2.8.

Todas essas ferramentas e frameworks descritos focam essencialmente na extração das estampas de tempo. Focam essencialmente nos detalhes técnicos da extração, no ajuste e na sincronização dos relógios, entre outros problemas envolvidos em uma análise por meio de linha de tempo. A grande quantidade variada de dispositivos de armazenamento e as altas capacidades de armazenamento fazem com que o número de dados gerados a serem analisados seja muito grande. Assim, pesquisas foram realizadas com foco em modelos que extraem conhecimento da correlação dos eventos e incidentes.

Em suas pesquisas, [Chabot et al. 2014a] propuseram um modelo para a reconstrução de eventos que foca na formalização dessa reconstrução contendo definições formais das entidades envolvidas em um incidente e quatro conjuntos de operadores, permitindo que o conhecimento contido no modelo proposto pelos autores seja extraído, manipulado e anali-

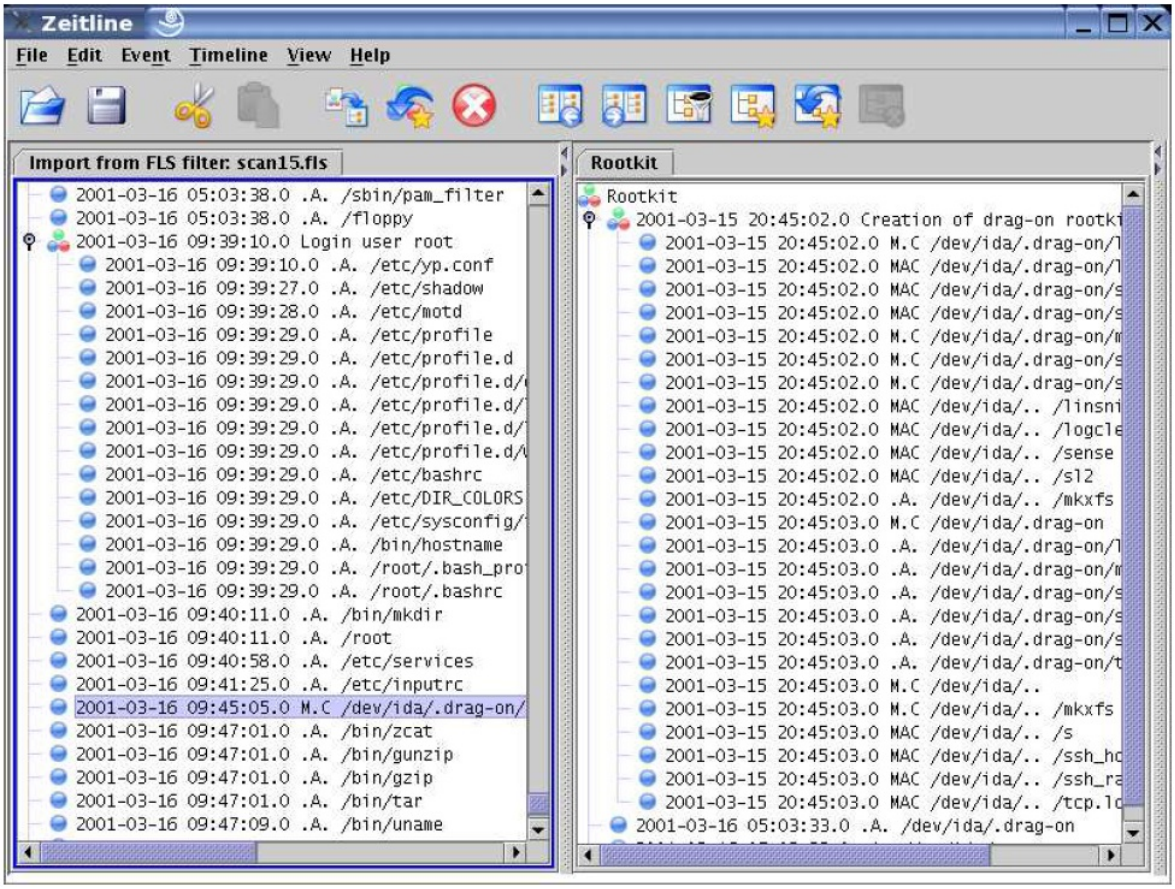


Figura 2.8: Visualização gráfica de eventos no Zeitline [Buchholz e Falk 2005]

sado. Entende-se a reconstrução de eventos como um grande processo no qual tem-se como entrada um conjunto de eventos, cuja saída é uma linha de tempo dos eventos. Conforme explicam [Chabot et al. 2014a], para a criação da linha do tempo, é necessário realizar a extração das estampas de tempo.

Os autores esclarecem que o auxílio aos especialistas não pode se limitar a montar única e exclusivamente uma linha no tempo. É necessário resolver a questão da grande quantidade de eventos gerados, as diferentes origens utilizadas para se extrair as informações, a necessidade de reunir os eventos em um modelo adequado, a preservação da integridade dos dados, a execução de todo o processo de investigação de forma correta e a validação.

Para reunir todas essas características, [Chabot et al. 2014a] propuseram a abordagem SADFC (*Semantic Analysis of Digital Forensic Cases*), no qual é proposto identificar e modelar o conhecimento relacionado a um incidente e o conhecimento relacionado com o processo de investigação, a fim de determinar as circunstancias do incidente e, dessa forma, formalizar a reconstrução dos eventos, definindo formalmente entidades envolvidas no incidente. Também define um modelo de representação de conhecimento em que são usadas definições prévias usadas para armazenar conhecimento sobre um incidente e usado para solucionar o caso. Provê extração de métodos de conhecimento contidos em fontes hete-

rogêneas para preencher o modelo de conhecimento e provê ferramentas que auxiliam os investigadores na análise dos conhecimentos extraídos.

O SADFC usa técnicas de gerenciamento de conhecimento, mineração de dados e semântica da web. O SADFC é a sinergia de três elementos, que são o modelo de conhecimento, o modelo de processo de investigação e a arquitetura orientada a ontologia.

Assim, [Chabot et al. 2014a] propuseram uma representação rica em conhecimento, contendo um grande conjunto de entidades, relações e processos de análise automatizados. Neste modelo, são definidas quatro entidades: sujeito/pessoa, objeto, evento e vestígio, conforme mostrado da figura 2.9.

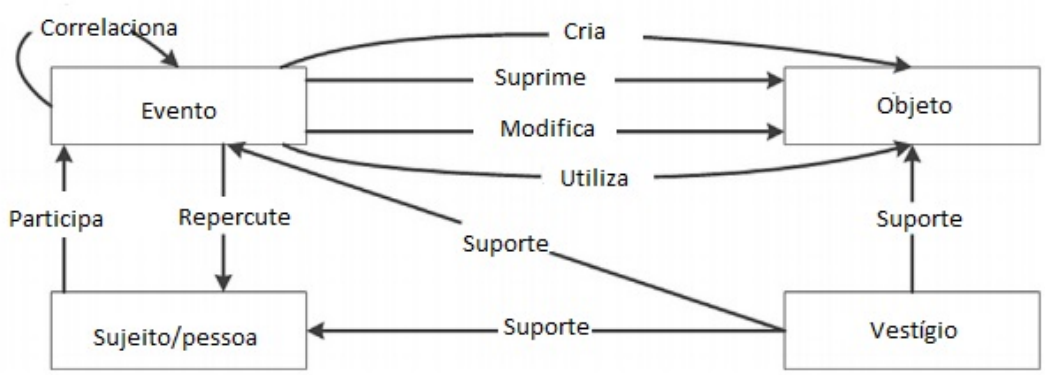


Figura 2.9: Modelo de conhecimento [Chabot et al. 2014a]

Para a reconstrução dos eventos, o SADFC possui quatro operadores, que são: operador de extração, usado para extrair informações dos vestígios de várias fontes; operador de mapeamento, no qual cria as entidades (eventos, objetos, sujeito/pessoa) associadas aos vestígios extraídos; operador de inferência, no qual utiliza conhecimento dos eventos, dos objetos ou dos sujeitos; operadores de análise, que auxiliam os investigadores na análise da linha de tempo.

No trabalho desenvolvido por [Chabot et al. 2014a] é introduzido um operador dedicado para identificação de eventos de correlação. Para a correlação entre dois eventos o operador utiliza da correlação temporal, correlação do sujeito, correlação do objeto e correlação das regras-base. As quatro entidades são assim caracterizadas:

Sujeito - são os atores envolvidos em um dado evento durante o ciclo de vida do evento, que podem ser humanos ou processos como um navegador, sistema operacional, etc. A entidade pode se relacionar a um ou mais eventos.

Objeto - os eventos interagem com um ou mais objetos, que podem ser chaves de registro, um arquivo, uma página web, etc.

Evento - é uma situação fática ocorrida em um intervalo de tempo qualquer. O evento pode se relacionar com um ou mais de um sujeito ou objeto.

Vestígio - são pedaços de informação, rastros de atividades que possibilitam reconstruir um evento passado. Pode-se encontrar vestígios em entradas de log, histórico de navegação da web, atividades de um dado software, etc.

Para ordenação dos eventos, os autores utilizaram a álgebra de [Allen 1983], como ilustrado na figura 2.10.

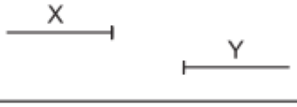
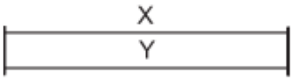
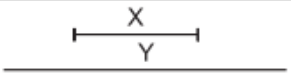
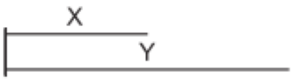
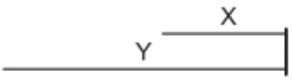
Functions	Constraints	Example
before(X,Y)	$x_{t_{end}} < y_{t_{start}}$	
equal(X,Y)	$x_{t_{start}} = y_{t_{start}}$ && $x_{t_{end}} = y_{t_{end}}$	
meets(X,Y)	$x_{t_{end}} = y_{t_{start}}$	
overlaps(X,Y)	$x_{t_{start}} < y_{t_{start}}$ && $x_{t_{end}} > y_{t_{start}}$	
during(X,Y)	$x_{t_{start}} > y_{t_{start}}$ && $x_{t_{end}} < y_{t_{end}}$	
starts(X,Y)	$x_{t_{start}} = y_{t_{start}}$	
finishes(X,Y)	$x_{t_{end}} = y_{t_{end}}$	

Figura 2.10: Álgebra de Allen [Chabot et al. 2014a]

Na reconstrução dos eventos, [Chabot et al. 2014b] propuseram uma arquitetura em multicamadas que realiza a reconstrução de eventos de forma automática. Para isso, utilizaram o modelo de representação de conhecimento, no qual armazenam ricas informações semânticas dos eventos. A arquitetura está fundada nas seguintes camadas:

Camada de extração - responsável por extrair informações de vestígios de diversas fontes. Após a extração, os dados são filtrados e as informações que não são relevantes são descartadas. Devido a origem de várias fontes, nesta camada, os dados são normalizados para serem traduzidos em um único formato.

Camada semântica - nesta camada, os dados extraídos pela camada anterior necessitam ser compreendidos, interpretados e traduzidos em conhecimento. As informações dos

vestígios extraídas são convertidas em entidades pelo operador de mapeamento. Todos os conhecimentos são reunidos em um único modelo de conhecimento.

Camada de raciocínio - esta camada analisa e melhora o conhecimento armazenado. Do conhecimento extraído pelas camadas anteriores, a camada de raciocínio deduz novos conhecimentos que não puderam ser identificados diretamente na cena do crime. Nesta camada são executadas análises básicas a fim de liberar o tempo do investigador para focar em eventos mais importantes que requerem sua maior habilidade e atenção. Esta ferramenta permite a correlação de eventos por meio de regras pré-definidas pelo investigador.

Camada de interface - é a camada que permite que os investigadores interajam com o modelo e é composto por três módulos:

- Ferramenta de visualização da linha de tempo - ferramenta de visualização gráfica da linha do tempo de todos os eventos armazenados;

- Interface de consultas para comandos SQL - permite ao investigador enviar consultas SPARQL;

- Painel de configurações - permite aos investigadores realizar configurações a respeito das regras usadas para a correlação dos eventos. A arquitetura proposta pelos autores [Chabot et al. 2014b] é ilustrada na figura 2.11.

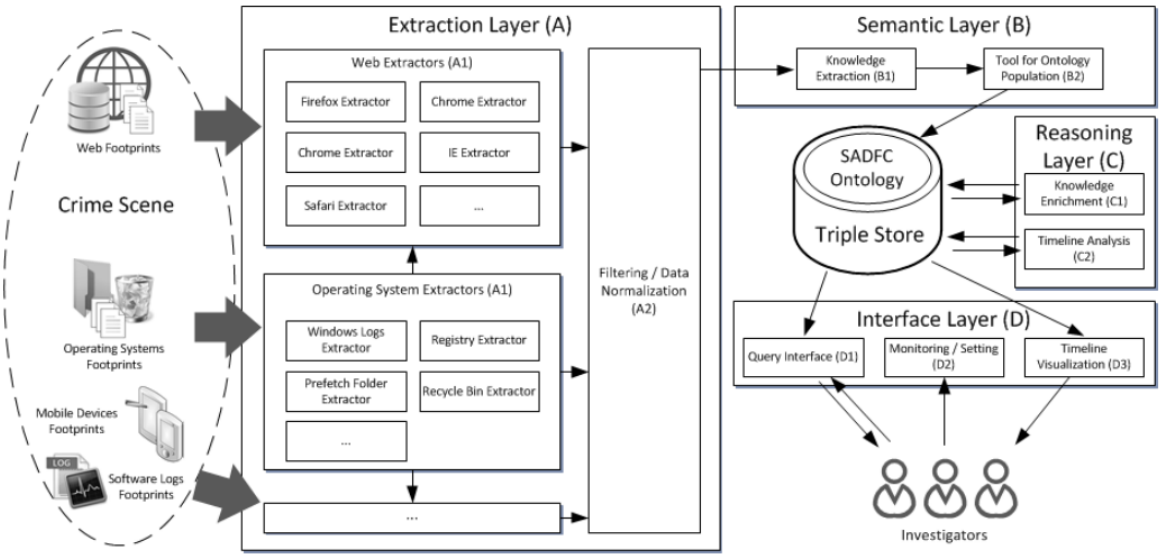


Figura 2.11: Arquitetura do modelo de geração de conhecimento proposta por [Chabot et al. 2014b]

Já Yu Jin [Jin 2013] propõe um framework para a visualização de linhas temporais ao analisar dispositivos com o sistema operacional Android, onde o pesquisador faz o uso

de mapas auto-organizáveis para analisar a relação de registros de atividades no sistema operacional. No trabalho realizado pelo autor, o algoritmo que implementa os mapas auto-organizáveis de [Kohonen 1982] foi escolhido por atender bem aos requisitos da proposta do trabalho, embora outros algoritmos de redes neurais pudessem ser usados. O autor menciona que os mapas auto-organizáveis são capazes de tratar vetores de entrada com mais de três dimensões e ter como saída um mapa de visualização em duas dimensões. Pelas características dos eventos realizados no sistema operacional Android, foi necessário utilizar um algoritmo de aprendizado de máquina não supervisionado, característica presente no algoritmo proposto por [Kohonen 1982]. A figura 2.12 ilustra o uso dos mapas auto-organizáveis no trabalho de [Jin 2013].

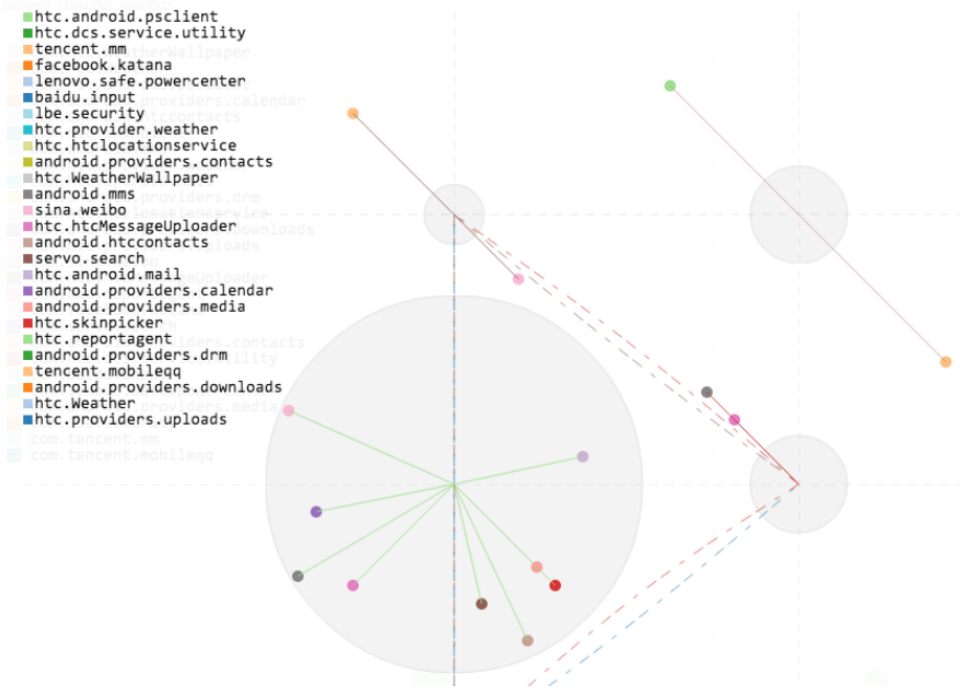


Figura 2.12: Visão do mapa auto-organizável [Jin 2013]

Nos estudos realizados por [Gantz e Reinsel 2012] e apresentados na figura 2.13 o universo digital cresce a um fator de 300, de 130 exabytes a 40.000 exabytes, ou 40 trilhões de gigabytes. O autor explica que de 2012 até o ano de 2020, o universo digital dobrará a cada dois anos.

Esse crescimento ocasiona diversos problemas para as investigações digitais. Quando os materiais são submetidos aos setores de perícia criminal no campo da informática forense, são analisados, em diversos casos, milhares de dados. A ferramenta *CyberForensics TimeLab* (CFTL) [Olsson e Boldt 2009] trouxe a possibilidade de examinar os dados por meio de telas gráficas. Ela analisa várias fontes, incluindo arquivos em formato JPEG, arquivos do sistema de arquivo, log de eventos do sistema operacional Windows, etc.

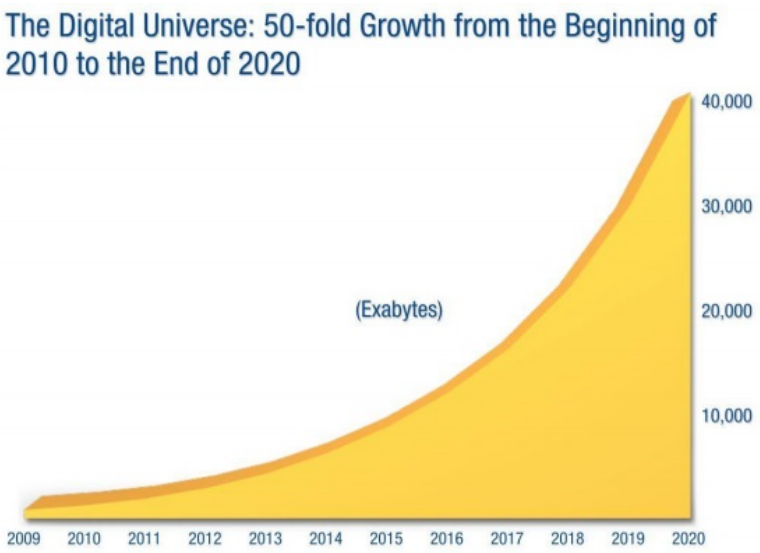


Figura 2.13: Crescimento do universo digital [Gantz e Reinsel 2012]

No entanto, a quantidade de informação gerada e o grande número de eventos da linha temporal não relacionados ao caso investigado dificultam a visualização dos eventos realmente relevantes. Da mesma forma, a ferramenta *log2timeline* [Guðjónsson 2010] possibilita a análise de linhas temporais de várias fontes, possibilitando ter uma visão geral dos eventos.

Porém, a visualização completa dessas fontes variadas em uma única linha temporal é complexa e dificulta o trabalho do perito criminal, tendo que analisar muitos registros descontextualizados, ocasionando muita perda de tempo no processo de análise. A figura 2.14 ilustra o exame de uma linha temporal sem qualquer contextualização, com todos os eventos de variadas fontes ordenadas apenas pelo marco cronológico.

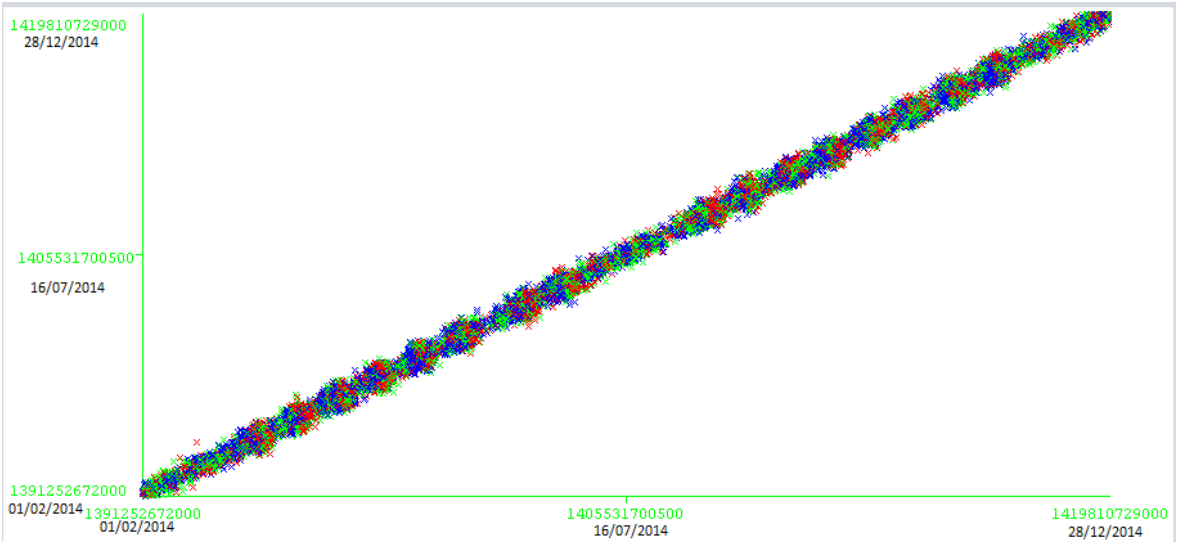


Figura 2.14: Linha temporal única sem contextos

Registros temporais (eventos de baixo nível) também podem ser vinculados automa-

ticamente a um evento (de alto nível), como sugerido por [Hargreaves e Patterson 2012]. Por exemplo, a detecção de alteração em vários arquivos do sistema pode ser vinculada à instalação de um programa malicioso. Mas de modo semelhante às outras técnicas, a visualização dos eventos na linha temporal única é prejudicada pela grande quantidade de informações pois alguns eventos podem não haver qualquer relação com a investigação digital.

No modelo dos pesquisadores [Chabot et al. 2014a], a reconstrução dos eventos é realizada por meio de operadores que extraem, criam entidades e expõem os dados em linhas temporais. Esse modelo melhora a análise do perito em relação a outros modelos pois a análise não circunscreve em apenas uma linha temporal. As linhas temporais são geradas após a aplicação dos operadores, que geram conhecimento e são armazenados em bancos de dados para posterior comparação. No entanto, o modelo não agrupa os eventos sem um marco temporal inicial.

De acordo com as circunstâncias do caso, pode ser necessário visualizar os eventos temporais segundo o contexto a que estão relacionados, reduzindo o ruído associado ao grande volume de registros temporais. Assim, os estudos, ferramentas e *frameworks* abordados neste capítulo não solucionam o problema da análise de uma linha temporal única. Para isso, este trabalho cria o conceito de contextualização. No processo de contextualização, os registros temporais são associados a pessoas, assuntos, eventos ou locais. O modelo para a geração das linhas temporais contextuais, proposto neste trabalho, visa apresentar os dados temporais agrupados segundo outras dimensões que não somente o tempo. Esse agrupamento ocorre utilizando a técnica de clusterização ou agrupamento. Com as linhas temporais contextualizadas, o especialista pode examinar com maior atenção os registros temporais presentes na linha temporal única, localizando registros específicos, já filtrados pelo processo de contextualização e agrupamento e examinar os registros temporais que estão próximos aos registros contextualizados e que não fizeram parte do processo ou que foram ignorados.

3 AGRUPAMENTO OU CLUSTERIZAÇÃO

Quando processos computacionais ou algoritmos possuem a capacidade de aprender, com base em algum critério anteriormente definido, tendo como ponto inicial a própria experiência dos dados de entrada a que são submetidos, pode-se dizer que essas técnicas fazem parte do contexto da teoria de "Aprendizagem de Máquina" [Silva 2009]. No aprendizado de máquina, as conclusões alcançadas tendo como fonte um conjunto de dados é obtida utilizando-se de indução como forma de inferência [Mitchell 1997]. A aplicação de algoritmos de indução é um passo no processo de descoberta do conhecimento [Kohavi e Provost 1998]. Nesse processo de aprendizagem por indução, a classificação pode ser feita pelo modo supervisionado ou pelo modo não supervisionado [Monard e Baranauskas 2003].

Em uma rede neural, a aprendizagem de máquina utilizando métodos supervisionados é realizada classificando-se as entradas em grupos que atendem a certos requisitos, formados por um "supervisor", que monitora o aprendizado. Assim, a rede neural aprende e novas entradas ainda não testadas podem ser classificadas de acordo com seus atributos. A rede neural pode responder de forma diferente a novos estímulos. O processo de aprendizado consiste nos ajustes sinápticos [Fausett 1993].

Nos métodos de aprendizagem não-supervisionados, a classificação dos dados é realizada pelo próprio conjunto de dados de entrada e o sistema procura por padrões ali existentes, agrupando-os, de forma a reunir os dados ou pontos similares. Assim, esses métodos classificam os dados de forma automática, sem ser necessário uma supervisão [Borges 2010]. [Becker 1991] ensina que para o aprendizado não supervisionado, o sistema ajusta seus parâmetros através dos estímulos fornecidos, que formam representações internas das entradas e cria-se classificações novas de forma automática.

Conforme [Witten et al. 2011], há quatro tipos de aprendizagens aplicados em um contexto de mineração de dados: aprendizagem por classificação, aprendizagem por associação, aprendizagem por agrupamento (clustering) e aprendizagem por previsão numérica.

Aprendizagem por agrupamento ou clusterização, é aquela em que são procurados grupos cujas amostras possuem características similares [Bishop 2006]. Este trabalho tem como foco a utilização desse método de aprendizagem.

No campo de estudos em aprendizado de máquina, um cluster é o agrupamento de

dados de certo conjunto de dados de entrada. Esses agrupamentos são realizados por meio de similaridades (dentro de um mesmo cluster) ou pelas diferenças (entidades de um cluster são diferentes de entidades de outro cluster). É desejável que os pontos em um cluster tenham uma distância pequena um do outro e que pontos em clusters diferentes sejam mais distantes. Para a realização desses agrupamentos, muitos algoritmos foram desenvolvidos.

Para [Jain et al. 1999], clusterização é a classificação não-supervisionada de dados, formando agrupamentos ou clusters. Ela representa uma das principais etapas de processos de análise de dados, denominada análise de clusters.

Conforme ensina [Nassif 2012], o agrupamento de dados é utilizado em análises exploratórias de dados quando não se conhece previamente os dados que serão submetidos à análise ou esse conhecimento é pequeno. Na mesma linha, o autor esclarece que a formação dos grupos é uma atividade puramente subjetiva. Como exemplo, pessoas diferentes podem fazer diferentes agrupamentos com o mesmo conjunto de objetos, a depender do critério de semelhança utilizado. A figura 3.1 ilustra a formação de agrupamentos por similaridades.

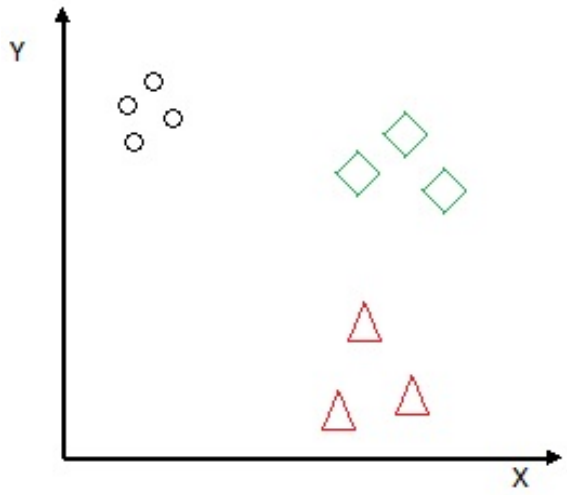


Figura 3.1: Agrupamento por similaridades

A processo de agrupamento é uma técnica utilizada na mineração de dados que objetiva agrupar de modo automático, utilizando o método não supervisionado, as “x” amostras de um conjunto de dados em k grupos, geralmente disjuntos. Os grupos formados são denominados clusters ou agrupamentos. Como a aprendizagem não é supervisionada, o processo de agrupamento não possui grupos pré-definidos ou rotulados com treinamento anterior. O processo de agrupamento é realizado por algum algoritmo que define como os dados serão dispostos nos grupos distintos. Os métodos de agrupamento têm como fundamento a ideia de distância ou de semelhança entre as amostras e definem a correlação dos itens amostrados a cada grupo segundo aquilo que cada amostra tem de semelhança em relação aos outros pertencentes ao mesmo cluster. O ponto central do processo de agrupamento é que elemen-

tos em um mesmo cluster devem ter características semelhantes, ou seja, cada elemento deve apresentar-se similarmente aos elementos dentro do mesmo grupo. Assim, o objetivo de um processo de clusterização é maximizar a similaridade em um mesmo grupo e maximizar as diferenças entre grupos distintos. Conforme [Cruz 2010], quando o número de clusters é conhecido a priori, ele é conhecido como Problema de K-Clusterização ou simplesmente Problema de Clusterização (PC). Caso contrário, quando o número ideal de clusters não é previamente conhecido, é denominado Problema de Clusterização Automática (PCA).

- **Aplicações**

  Uma das aplicações do método de agrupamento é empregá-la com o objetivo de reduzir o número de elementos para um número de subgrupos com suas respectivas características e, assim, direcionar a observação dos elementos conforme as características intrínsecas daquele agrupamento [Cassiano 2014]. O processo de agrupamento também é amplamente utilizado para a identificação de relacionamento entre os elementos do conjunto de dados. A clusterização também encontra aplicação no processo de reconhecimento de padrões. Esta técnica objetiva classificar as amostras (padrões) em um número de categorias ou classes (cluster). Como ensina [Cassiano 2014], utiliza-se o processo de agrupamento para cumprir ao menos um dos seguintes objetivos:

  1. Identificação da estrutura subjacente - para obter 'insights' sobre os dados, gerar hipóteses, detectar anomalias, e identificar características marcantes;
  2. Classificação Natural - identificar o grau de semelhança entre as formas ou organismos (filogenética);
  3. Compressão - como um método para a organização dos dados e resumindo-o através de protótipos do cluster.

- **Limitações**

  Encontrar o melhor agrupamento é um problema NP-Completo (não é computacionalmente possível encontrá-lo) [Hruschka e Ebecken 2003], a menos que *n* (número de elementos) e *k* (número de clusters) sejam extremamente pequenos, uma vez que o número de partições distintas em que podemos dividir *n* objetos em k clusters aumenta aproximadamente com $\frac{k^n}{n!}$ [Cassiano 2014].

- **Medida de Similaridades**

  Realizar o processo de clusterização objetiva agrupar os elementos dentro de grupos que tendem a ser mais similares entre si do que elementos presentes em outros grupos. Para alcançar o objetivo, é necessário verificar quão similares são os elementos

[Cassiano 2014]. A medida de similaridade refere-se tanto a uma similaridade quanto a uma dissimilaridade, ou distância. Assim, a similaridade mede o quanto dois objetos são semelhantes (ou próximos entre si), enquanto a dissimilaridade mede o quanto dois elementos são diferentes (ou distantes entre si). Caso essas medidas forem normalizadas para o intervalo [0,1], uma pode ser obtida subtraindo-se a outra do valor 1 [Nassif 2012].

Os elementos são mapeados em um espaço vetorial conforme seus atributos. Assim, os algoritmos de agrupamento trabalham com a seguinte estrutura de dados, em que *n* elementos cada qual com *p* atributos formando uma matriz *n* x *p* conforme ilustrado na figura 3.2.

$$X = \begin{bmatrix} x_{11} & x_{12} & x_{13} & \cdots & x_{1p} \\ x_{21} & x_{22} & x_{23} & \cdots & x_{2p} \\ x_{31} & x_{32} & x_{33} & \cdots & x_{3p} \\ x_{41} & x_{42} & x_{43} & \cdots & x_{4p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & x_{n3} & \cdots & x_{np} \end{bmatrix}$$

Figura 3.2: Matriz atributo-valor [Cassiano 2014]

Quando os elementos não podem ser mapeados em um espaço vetorial por causa das suas características típicas dos atributos, obtém-se a medida de dissimilaridades entre os elementos por meio de uma matriz de (dis)similaridades onde o valor d*ij* representa a distância ou similaridade entre os elementos x*i* e x*j*.

$$\begin{array}{c} \\ \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \vdots \\ \boldsymbol{x}_n \end{array} \begin{array}{c} \begin{array}{cccc} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_m \end{array} \\ \begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1m} \\ d_{21} & d_{22} & \cdots & d_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ d_{n1} & d_{n2} & \cdots & d_{nm} \end{bmatrix} \end{array}$$

Figura 3.3: Matriz de (dis)similaridades [Nassif 2012]

Ademais, muitos algoritmos de clusterização necessitam apenas dos valores das distâncias entre eles, não sendo preciso, portanto, uma representação vetorial dos elementos. Recebem como entrada a matriz de dissimilaridades [Cassiano 2014].

3.1 O Algoritmo *K-Means*

Conforme ilustra [Borges 2010], os algoritmos de aprendizagem de máquina baseados no método não-supervisionado utilizam as próprias informações da base de dados para realizar a classificação e agrupamento dos dados. Dentre muitos algoritmos baseados no método particional e não-supervisionado, o *k-means* é um dos mais conhecidos. Este trabalho utiliza este algoritmo no experimento para a implementação do modelo e foi escolhido para a etapa de clusterização por ser um algoritmo simples, rápido e de baixa complexidade. No entanto, outros algoritmos com outros métodos de clusterização podem ser utilizados.

O algoritmo *k-means* se utiliza de uma técnica chamada de particional, no qual cria uma partição na tentativa de recuperar a estrutura original dos dados [Huang 1998]. O *k-means* se baseia em centro, ou seja, o agrupamento é representado por um ponto central do grupo, denominado de centroide.

O algoritmo é bastante difundido e utilizado com sucesso em muitas aplicações, sendo o mais popular e mais simples algoritmo particional [Cassiano 2014]. A ideia é posicionar os dados de forma a minimizar a distância dos pontos no mesmo grupo e maximizar a distância de pontos em grupos diferentes [Borges 2010]. Entre os algoritmos do tipo particional, a distância Euclidiana é a mais comum para o cálculo de medidas de dissimilaridades [Macario Filho 2015]. A distância Euclidiana é uma das medidas que podem ser utilizadas. Para o processo de agrupamento, outras medidas como Cosseno, Jaccard, Edição (utilizada em sequencias de DNA) [Fonseca e de Castro Reis 2002], etc., podem ser utilizadas. As etapas que descrevem o *k-means* podem ser definidas como [Araújo 2008]:

1. Selecione *k* instâncias para serem os centroides iniciais dos grupos;
2. Atribua todas as instâncias ao centroide mais próximo;
3. Recalcule o centroide para cada grupo;
    - Calcule a média de todas as instâncias do grupo;
4. Repita os passos 2 e 3 até que os centroides não mudem;

A complexidade do algoritmo *k-means* é *O(nkt)* onde *n* é o número total de objetos, *k* é o número de clusters e o *t* é o número de iterações [Umale e Nilav 2014]. Pela sua simplicidade, ele é indicado para grandes bases de dados [Wu et al. 2008]. A figura 3.4 ilustra o funcionamento do algoritmo *k-means*:

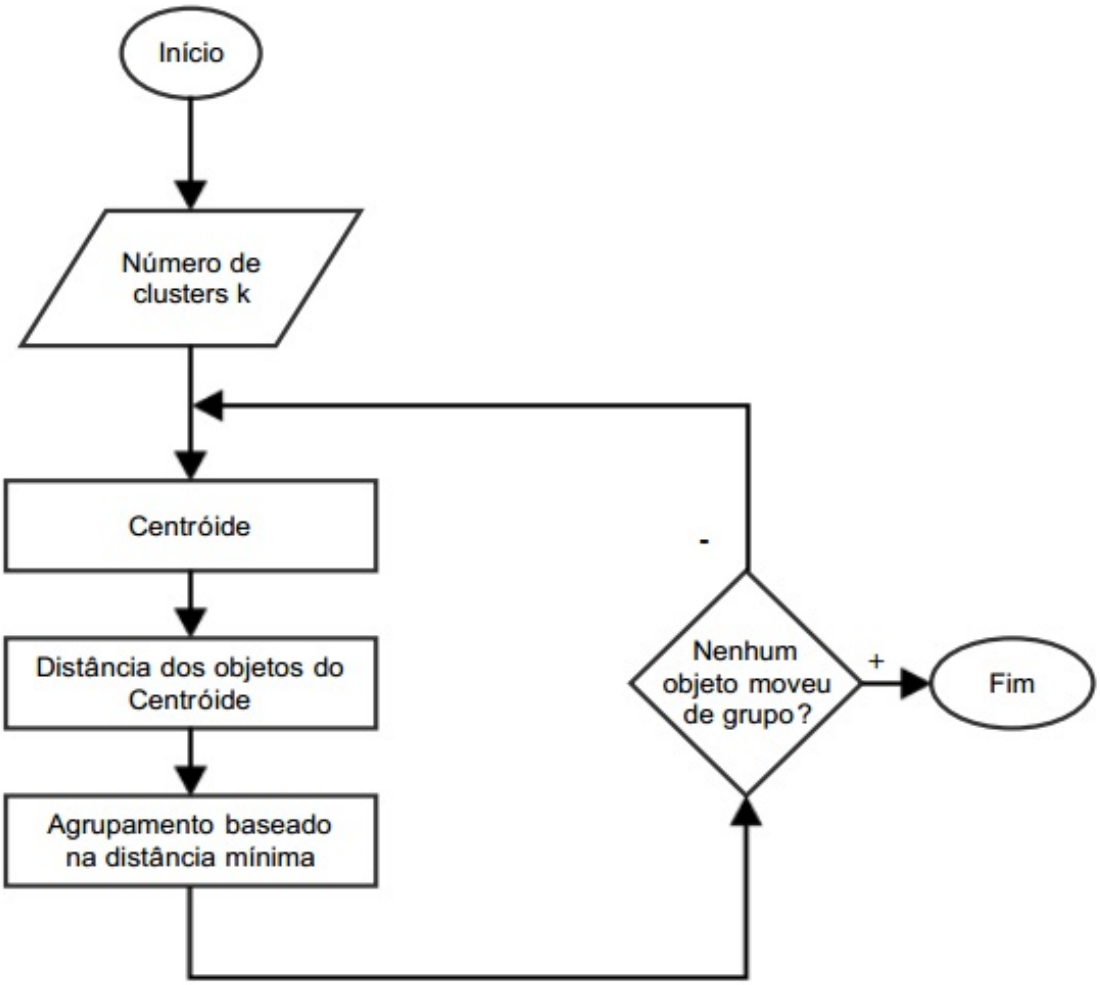


Figura 3.4: Fluxograma do *k-means* [Teknomo 2006]

3.2 Escolha do número de clusters

Quando se adota o conceito de clusterização utilizando-se do algoritmo *k-means*, seja na mineração de dados ou em qualquer outro campo de atuação, um problema fundamental, e em grande parte não resolvido é a determinação do número ideal de grupos em um conjunto de dados. A escolha do valor do *k* pode levar a muitas interpretações, pois, de acordo com a forma e a escala dos pontos distribuídos, a leitura será diferente. Ademais, variar o número de grupos de modo indiscriminado até haver um cluster para cada ponto no conjunto de dados, reduzirá a quantidade de erro no agrupamento resultante, de modo que cada ponto seja o próprio grupo. Nem sempre há um conhecimento prévio das propriedades do conjunto de dados que possa levar a escolha de um número verdadeiro de grupos. Assim, algum método deve ser utilizado para realizar a estimativa do valor do *k*. Neste trabalho, discutiremos sobre dois métodos: cotovelo e silhueta.

No método do "cotovelo", inicia-se com um número reduzido de grupos que é va-

riado, aumentando-se paulatinamente. Com o aumento do número de grupos, é verificada a variância em função de cada agrupamento. Se traçarmos um gráfico, a porcentagem de variância apresentada pelos grupos em relação ao número k, os primeiros agrupamentos adicionarão muita informação (apresentam muita variância), mas em algum ponto o ganho marginal cairá, dando um ângulo no gráfico. O número de grupos é escolhido neste ponto, daí o "critério de cotovelo". Este "cotovelo"nem sempre pode ser inequivocamente identificado [Ketchen e Shook 1996].

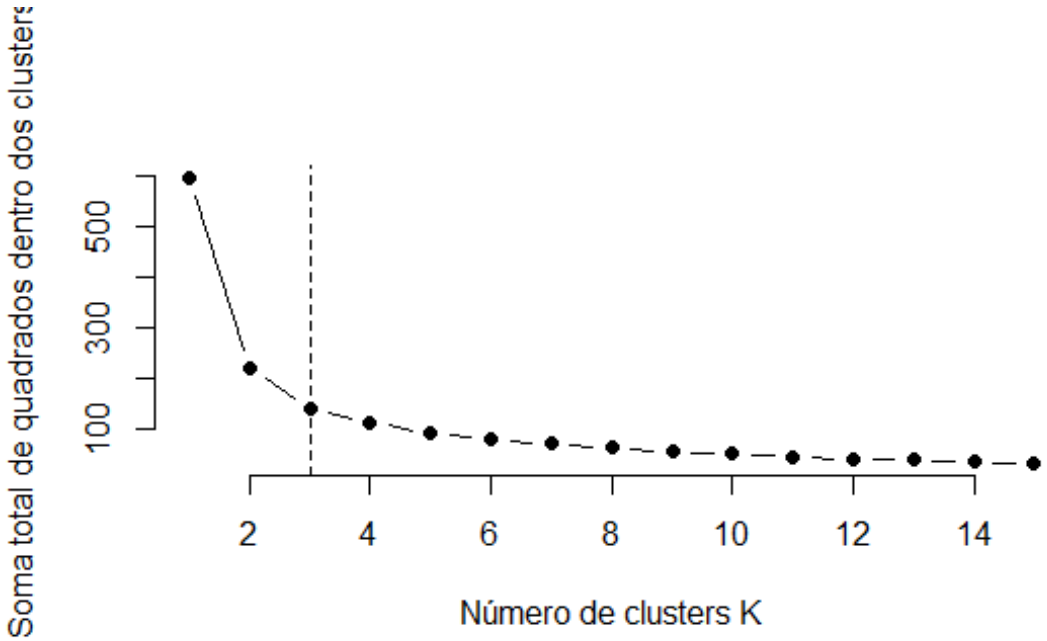


Figura 3.5: Método do cotovelo

O método da silhueta [Kaufman e Rousseeuw 2009], ou silhueta média dos dados é outro método para identificar o número ideal de grupos. Posta no gráfico, a silhueta de um conjunto de dados é uma medida de quão próximo ela está dos pontos de dados dentro de seu grupo e quão vagamente é correspondido aos dados do cluster vizinho, ou seja, o cluster cuja distância média é menor. Assumem-se valores entre -1 e 1. Uma silhueta perto de 1 significa dizer que o dado está em um grupo adequado, enquanto uma silhueta perto de -1 significa que o dado está pertence a um grupo inadequado [Kaufman e Rousseeuw 2009]. As técnicas de otimização, como os algoritmos genéticos, são adequadas na determinação do número de grupos que dá origem à maior silhueta [Lletı et al. 2004]. Também é possível reescalonar os dados de tal forma que a silhueta é mais provável de ser maximizada no número correto de clusters.

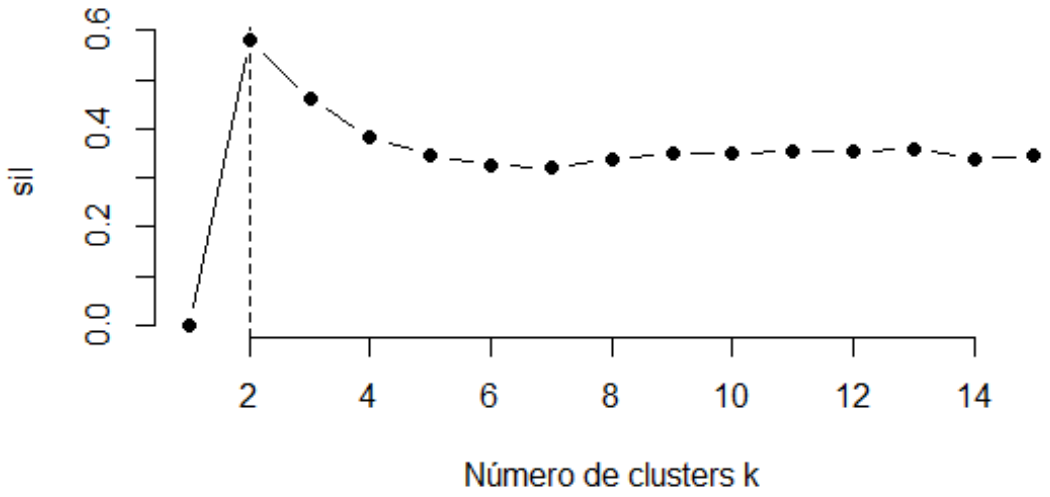


Figura 3.6: Método da silhueta

3.3 A ferramenta WEKA

A ferramenta de software *WEKA (Waikato Environment for Knowledge Analysis)* é um conjunto de algoritmos de aprendizado de máquina para a realização de tarefas em mineração de dados. Foi escolhida para utilização neste trabalho por ser um software livre, de fácil implementação e por já implementar uma versão do algoritmo *k-means*. O projeto do WEKA teve início em 1992, na Universidade de Waikato, Nova Zelândia [Hall et al. 2009]. O *WEKA* tem o código fonte disponível para estudos e modificações de acordo com a *General Public License (GPL)*. Conforme [Hall et al. 2009]:

"O programa visa desenvolver uma ferramenta facilitadora no estado da arte para o desenvolvimento de técnicas de aprendizado de máquina e investigar sua aplicação em áreas-chave da economia da Nova Zelândia. Especificamente, vamos criar uma bancada para a aprendizagem de máquina, determinar os fatores que contribuem para seu sucesso aplicado na indústria agrícola, e desenvolver novas métodos de aprendizado de máquina e formas de avaliar a sua eficácia."

3.3.1 Conceitos básicos do WEKA

Como descrito no manual do *WEKA* [Bouckaert et al. 2016] são apresentados a seguir os conceitos básicos para a compreensão da ferramenta.

- **Dataset**

Dataset é um conjunto de dados, semelhante a uma planilha bidimensional ou a uma tabela de um banco de dados [Silva 2004]. No *WEKA*, os conjuntos de dados são representados externamente pelos arquivos em formato *ARRF (Attribute-Relation File Format)*. Embora seja aceito a entrada de dados em outros padrões, como, por exemplo, o simples *CSV* (*Comma Separeted Values* – Valores separados por vírgulas), o método mais utilizado do *WEKA* para carregar dados é no Formato de Arquivo de Atributo-Relação (ARFF). O formato permite definir os tipos de dados que estão sendo carregados, e então fornecer seus dados propriamente ditos [Damasceno 2010]. É definido no arquivo cada atributo (coluna) e o que o mesmo conterá. Cada linha de dados é fornecida em um formato delimitado por vírgulas [Bouckaert et al. 2016]. A seguir, na figura 3.7, é apresentado um exemplo de um *Dataset* em formato ARFF, dividido em cabeçalho e dados pela declaração "@DATA".

```
@RELATION caso_pi

@ATTRIBUTE fonte_dos_dados    string
@ATTRIBUTE data      DATE "yyyy/MM/dd HH:mm:ss"
@ATTRIBUTE assunto   {PTHC, PEDO, SEXO}
@ATTRIBUTE local     {escola, notebook1, notebook2, celular, escritorio}
@ATTRIBUTE evento    {Tirar_foto,chat_whatsapp,chat_messenger_facebook,download_arquivos}
@ATTRIBUTE nome      {Rodrigo, Carlos, Lucas, Frederico, Sade}

@DATA
Fonte_10,"2014/010/05 22:27:26",SEXO,celular,chat_messenger_facebook,Rodrigo
Fonte_3,"2014/010/010 09:19:55",?,notebook2,chat_messenger_facebook,Rodrigo
Fonte_4,"2014/010/03 10:13:47",PEDO,celular,chat_messenger_facebook,Carlos
Fonte_8,"2014/010/017 18:39:33",?,notebook1,download_arquivos,Carlos
Fonte_7,"2014/010/015 12:39:02",?,notebook2,?,?
Fonte_6,"2014/010/021 20:37:53",PEDO,notebook2,download_arquivos,Rodrigo
Fonte_10,"2014/010/023 10:31:54",PTHC,escritorio,download_arquivos,Carlos
Fonte_5,"2014/010/011 17:15:31",PTHC,escritorio,download_arquivos,Rodrigo
Fonte_10,"2014/010/01 02:08:50",PTHC,notebook1,download_arquivos,Rodrigo
Fonte_6,"2014/010/021 17:17:51",SEXO,escritorio,download_arquivos,Carlos
Fonte_6,"2014/010/017 22:13:56",SEXO,notebook2,chat_messenger_facebook,Rodrigo
Fonte_1,"2014/010/015 03:24:28",PEDO,escritorio,download_arquivos,Rodrigo
Fonte_8,"2014/010/019 12:26:08",PEDO,notebook2,download_arquivos,Carlos
Fonte_2,"2014/010/06 14:32:09",PEDO,escritorio,chat_whatsapp,Rodrigo
Fonte_2,"2014/010/015 07:13:51",PTHC,celular,chat_messenger_facebook,Carlos
Fonte_4,"2014/010/04 07:58:37",PEDO,escritorio,chat_whatsapp,Carlos
Fonte_8,"2014/010/018 03:52:06",PTHC,escritorio,download_arquivos,Carlos
Fonte_5,"2014/010/022 04:55:42",SEXO,notebook2,chat_whatsapp,Rodrigo
```

Figura 3.7: Arquivo do tipo ARFF

- **Classificadores**

No *WEKA*, todos os algoritmos de aprendizado de máquina derivam da classe abstrata "weka.classifiers.AbstractClassifier" [Bouckaert et al. 2016] e, portanto, são considerados classificadores. Um modelo de classificador é um mapeamento arbitrário complexo de todos-exceto-um conjunto de dados de atributos para o atributo de classe.

- **Filtros**

Os filtros são as classes que podem transformar os conjuntos de dados, seja retirando ou adicionando atributos, seja realizando a reamostragem do conjunto de dados, removendo exemplos e assim sucessivamente. Esta funcionalidade oferece suporte útil para pré-processamento de dados, que é um passo importante no aprendizado de máquina.

- **Interface gráfica**

O *WEKA* possui uma interface gráfica como ponto de partida para utilização da ferramenta. Ela é a classe weka.gui.GUIChoosere consiste de 4 (quatro) botões para as principais funções além do menu: *Explorer, Experimenter, KnowledgeFlow e Simple CLI*, conforme podemos visualizar na figura 3.8.

Figura 3.8: Tela de início da interface gráfica do WEKA (WEKA GUI Chooser)

Os botões são utilizados para executar as seguintes funções:

Explorer: Um ambiente para explorar a execução dos algoritmos via interface gráfica, conforme figura 3.9.

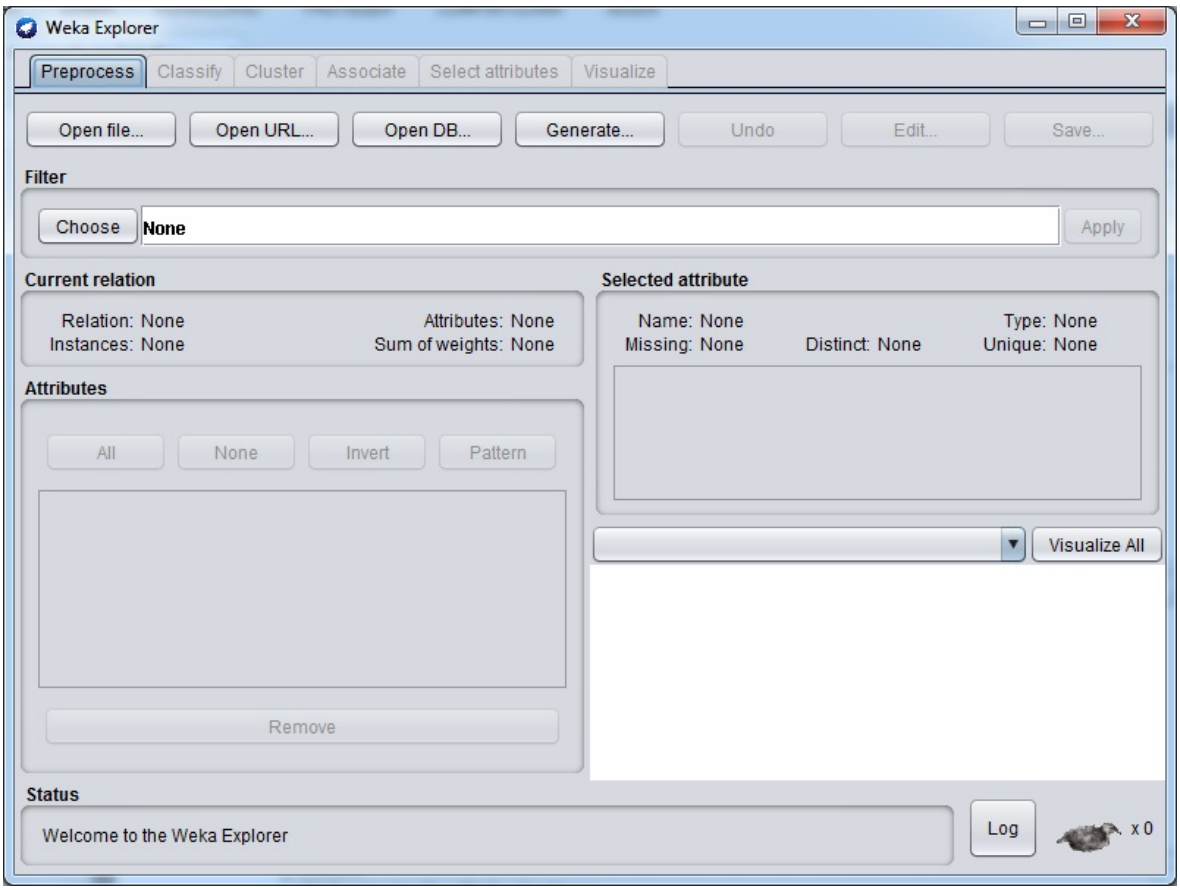


Figura 3.9: Janela que permite a execução dos algoritmos via interface gráfica

No ambiente da tela gráfica *Explorer*, é possível navegar pelas guias *Preprocess, Classify, Cluster, Associate, Select Attributes, Visualize*.

A guia *Cluster* é utilizada para a escolha do algoritmo de clusterização e realização do processo, como ilustra a figura 3.10.

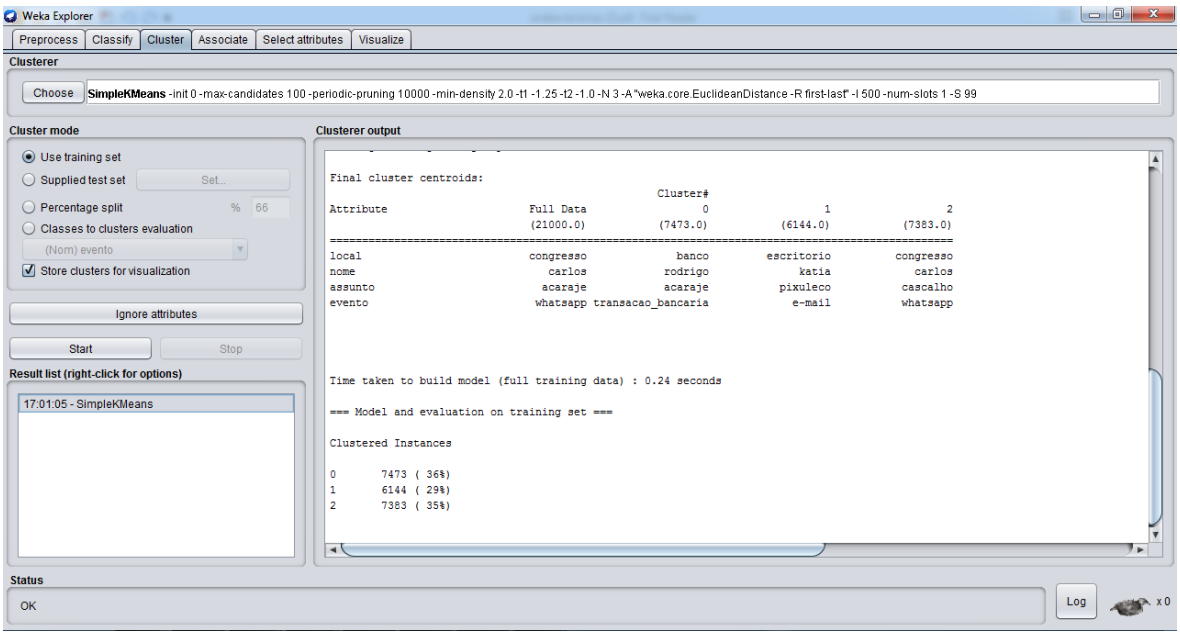


Figura 3.10: Janela que permite a escolha do algoritmo de clusterização e realização do processo

A guia *Visualize* é utilizada para a visualização da saída do processo após a execução do algoritmo, como ilustra a figura 3.11.

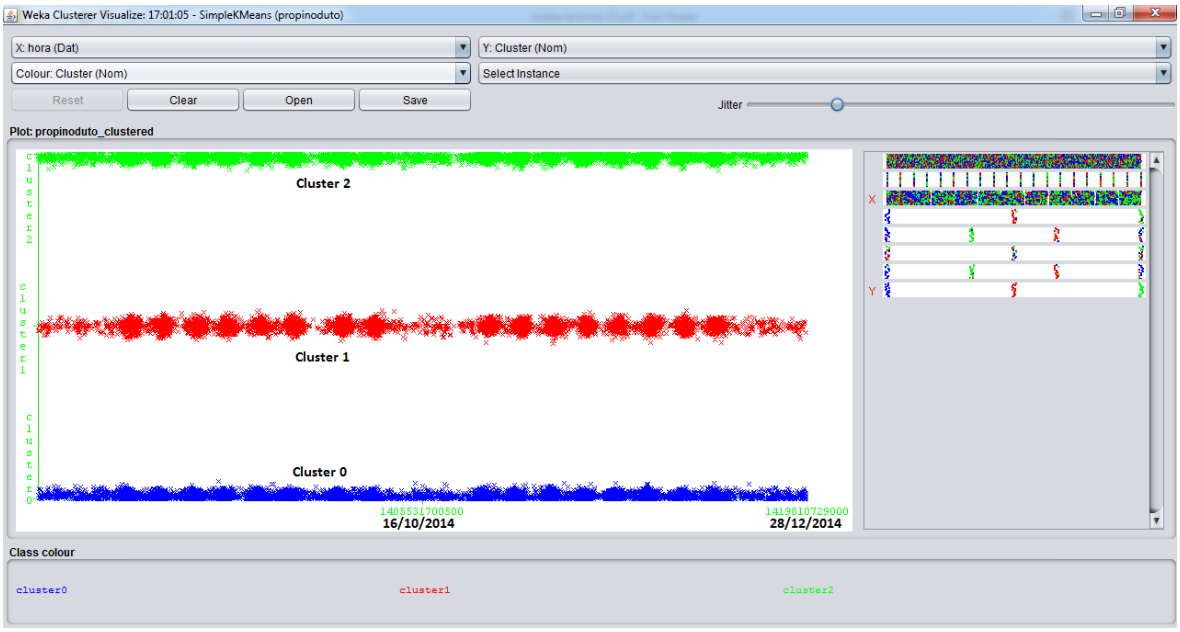


Figura 3.11: Janela de visualização dos clusters

Experimenter: Nesta janela, realiza-se experiências e testes estatísticos entre os sistemas de aprendizagem.

KnowledgeFlow (fluxo de conhecimento): provê um ambiente que suporta essencialmente as mesmas funções que o Explorer, mas com uma interface drag-and-drop (arraste-e-solte). O diferencial é que ele possui a vantagem de suportar aprendizagem incremental.

SimpleCLI: Fornece uma interface de linha de comando simples que permite a execução direta de comandos WEKA para sistemas operacionais que não fornecem sua própria interface de linha de comando.

A interface gráfica, embora útil e de fácil utilização, não contempla todos os recursos disponíveis. Recomenda-se a utilização da interface de linha de comando, o qual contém todas funcionalidades do WEKA.

4 LINHAS TEMPORAIS CONTEXTUAIS

4.1 Modelo proposto

A construção de linhas temporais contextuais está dividida em cinco etapas, conforme ilustra a figura 4.1.

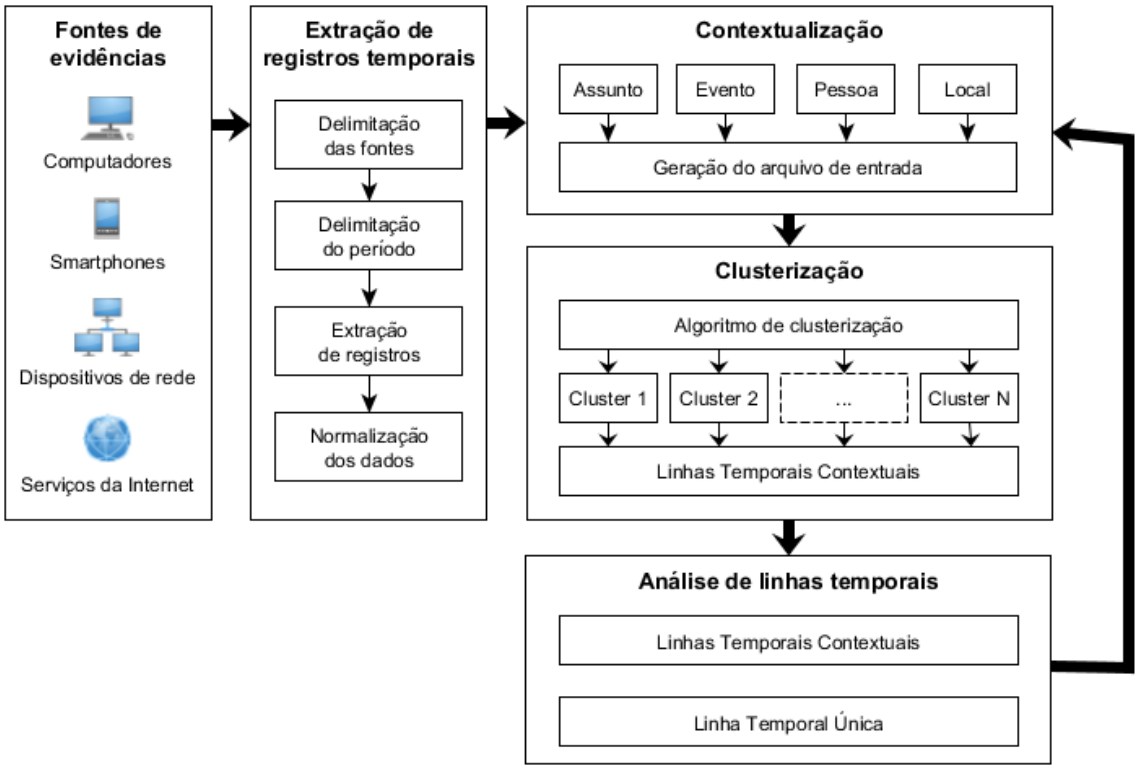


Figura 4.1: Modelo para geração das linhas temporais contextuais

4.2 Fontes de evidências

Esta etapa é a determinação das fontes de evidência das quais serão extraídos os registros temporais. Como discutido anteriormente, registros temporais podem ser provenientes de várias fontes distintas como computadores, smartphones, dispositivos de rede, entre outros. As fontes podem estar associadas a apenas um indivíduo ou a muitas pessoas. Por exemplo, em uma busca e apreensão podem ser coletados diversos dispositivos como notebooks, tablets, celulares, discos rígidos externos, entre outros, de apenas um investigado. Da mesma forma, em uma grande operação, podem ser coletados diversos dispositivos de

pessoas diferentes. Esta etapa é feita manualmente.

4.3 Extração de registros temporais

A etapa de extração de registros temporais tem por objetivo aplicar uma série de métodos a fim de obter as informações (vestígios) dos equipamentos levados à perícia.

4.3.1 Delimitação das fontes

Antes da extração propriamente dita, é necessário dimensionar qual o escopo da análise. Há muitos crimes que os sistemas computacionais são apenas um meio para o cometimento da infração, e não a finalidade (crimes digitais) propriamente dita. Esses crimes são classificados [Hoelz et al. 2014] em próprios (praticados no ambiente cibernético, tendo o recurso de computação como alvo), impróprios (o crime é praticado utilizando-se de equipamentos computacionais mas o alvo está fora no espeço cibernético como, por exemplo, compartilhamento de arquivos de pornografia infantil) e indiretos (equipamentos computacionais não são meios, mas servem de rastros para as condutas criminosas como, por exemplo, planilhas eletrônicas que registram pagamentos em crimes de corrupção).

Assim, pode-se delimitar as fontes, dependendo do incidente investigado. Por exemplo, nas investigações policiais de crimes relacionados a invasões de sistemas governamentais, é interessante a busca por códigos maliciosos. Neste cenário, é muito importante a coleta de registros em logs de eventos do sistema operacional, registros de instalação ou remoção de software, logs de *firewalls* pessoais, de rede, etc.

Em crimes, por exemplo, relacionados a pornografia infantil, a busca será por evidências de imagens armazenadas no dispositivo que contenham imagens ou cenas com conteúdo pornográfico infanto-juvenil, pela presença de softwares de compartilhamento de arquivos, em especial as P2P (peer-to-peer), estes muito utilizados pelos criminosos para o compartilhamento de arquivos contendo imagens de sexo envolvendo crianças e adolescentes. Neste crime, o comportamento dos serviços do sistema operacional pode ser irrelevante para o caso e, portanto, as estampas de tempo para a linha temporal são desnecessárias. No entanto, a visita a sites ou aplicativos de redes sociais, o acesso a contas de e-mail e a conexão a dados armazenados em nuvem são informações muito úteis, pois podem correlacionar o usuário da máquina ou dispositivo (pessoa) aos fatos criminosos (compartilhamento de arquivos com pornografia infantil) que aconteceram no mesmo intervalo de tempo.

Portanto, registros temporais podem não ser interessantes para determinado incidente

como, por exemplo, datas de criação, acesso e modificação de arquivos do sistema operacional.

4.3.2 Delimitação do período

A delimitação do período de interesse faz-se necessário dependendo do caso analisado, das suspeitas, dos vestígios, etc. Em crimes envolvendo organização criminosa em que se reúnem para a prática de crimes tipificadas como corrupção e lavagem de dinheiro, por exemplo, o lapso temporal pode não ser de conhecimento da equipe de investigação. Assim, no exemplo mencionado, não se sabe o ponto temporal inicial das condutas. A critério do perito, o intervalo temporal pode ser delimitado, por exemplo, em meses, dias, horas.

4.3.3 Extração de registros

Esta etapa é a responsável por extrair as informações presentes em aparelhos de telefonia móvel (smartphones), tablets, computadores desktop, notebooks, HDs externos, etc. Em cada equipamento, é possível obter registros de navegação web, conversas de aplicativos de trocas de mensagens instantâneas (Whatsapp, MSN Messenger, Facebook Messenger e outros). Para a extração dessas informações, muitas ferramentas estão disponíveis como, por exemplo, o *log2timeline* [Guðjónsson 2010]. A figura 4.2 apresenta um exemplo da coleta de dados do navegador Chrome da Google. A extração é feita de forma automática.

```
0|[Chrome History] (URL visited) User: john
http://tools.google.com/chrome/intl/en/welcome.html (Get started with
Google Chrome) [count: 1] Host: tools.google.com type: [START_PAGE -
The start page of the browser] (URL not typed directly) (file:
History)|20752894|0|0|0|0|1261044829|1261044829|1261044829|1261044829
```

Figura 4.2: Dados extraídos do navegador Chrome com *log2timeline* [Guðjónsson 2010]

A ferramenta *log2timeline* suporta os seguintes tipos de dados [Plaso-Wiki 2016]:

- Tipos de mídias de armazenamento (*storage media types*): EWF (EWF-E01, EWF-Ex01, EWF-S01), QCOW versões 1, 2, 3, dispositivos de armazenamento, VHD, VMDK;
- Volume de sistemas: *Apple Partition Map* (APM), *BitLocker Disk Encryption* (BDE), *FileVault Disk Encryption* (FVDE) (ou *FileVault* 2), GPT, LVM, MBR, *Volume Shadow Snapshots* (VSS);
- Sistema de arquivo: EXT (versões 2, 3, 4), FAT, HFS, HFS+, HFSX, NTFS versão 3, UFS versão 1, 2.

- Formato de arquivos: *Apple System Log* (ASL), *Android usage-history* (*app usage*), *Basic Security Module* (BSM), *Firefox Cache*, e muitos outros.
- Formato de arquivos Bencode: *Transmission* e *uTorrent*.
- Formato de arquivos de banco de dados ESE: *Internet Explorer WebCache*, *Windows 8 File History*.
- Formato Arquivo Composto OLE: Informações de resumo do documento, informações de resumo (*top-level only*), arquivos *Jump Lists* .automaticDestinations-ms.
- Formato Lista de propriedades (plist): *Airport, Apple Account, Bluetooth, Install History, iPod/iPhone, Mac User, Safari history, Software Update, Spotlight, Spotlight Volume Information, Timemachine*.
- Formato de arquivos SQLite: *Android call logs, Android SMS, Chrome cookies, Chrome browsing and downloads history, Chrome Extension activity, Firefox cookies, Firefox browsing e downloads history, Google Drive, iMessage* (iOS and Mac OS X), Kik (iOS), *Launch services quarantine events*, *MacKeeper cache, Mac OS X document versions, Skype text conversations, Twitter (iOS), Zeitgeist activity database*.
- Registro do Windows (*Windows Registry*): *AppCompatCache, BagMRU (or ShellBags), CCleaner, Explorer ProgramsCache, Less Frequently Used (LFU), MountPoints2, Most Recently Used (MRU) MRUList e MRUListEx (incluido suporte a shell), MSIE Zones, Office MRU, Outlook Search, Run and RunOnce keys, SAM, Services, Shutdown, Task Scheduler Cache (Task Cache), Terminal Server MRU, Timezones, Typed URLS, USB, USBStor, UserAssist, WinRar, Windows version information*.

Mensagens instantâneas são muito relevantes, pois são capazes de revelar, por exemplo, a combinação de um pagamento de propina entre um agente público e um representante de uma empresa privada, não só pela mensagem de texto ali contida, mas também pelas imagens enviadas, e até mesmo conversas em áudio. Assim, as atividades de uma pessoa podem ser analisadas em detalhes, por meio da aquisição desses registros.

Como existem várias fontes de dados, a etapa de extração deve tratar de todos os detalhes técnicos da extração de cada dispositivo físico ou informações em software. Há fontes de dados muito bem estruturadas, cuja a extração e interpretação dos dados, utilizando as ferramentas corretas, são de certa maneira fáceis [Chabot et al. 2014a]. No entanto, dados não estruturados como arquivos de imagem necessitam de algoritmos específicos para a decodificação e disponibilização dos dados.

É possível verificar (em um caso de crime de pornografia infantil, por exemplo) o comportamento de determinado indivíduo examinando o histórico de navegação, as páginas nos favoritos, a presença de softwares de downloads de arquivos peer-to-peer (P2P), etc. Ao examinar e confrontar o histórico de navegação com eventual página nos favoritos do usuário é possível diferenciar o acesso acidental do acesso intencional.

4.3.4 Normalização dos dados

É responsabilidade da etapa de extração de registros temporais realizar a normalização dos dados, uma vez que ocorre a extração de dados de várias fontes diferentes. Assim, problemas relacionados a estampa de tempo dos arquivos, como, por exemplo, desvio nos relógios [Stevens 2004], configuração do fuso-horário (*timezone*) diferente [Kaart e Laraghy 2014], formato da estampa de tempo diferente por se tratar de sistemas de arquivos desiguais (NTFS e EXT3, por exemplo) [Carrier 2005], devem ser solucionados. É realizado de forma manual, utilizando ferramentas para automatizar algumas etapas.

Este trabalho não tem a preocupação de pesquisar sobre os detalhes técnicos da extração dos vestígios, nem do formato das informações devido a existência de diversas fontes de dados. Nessa etapa, assume-se a utilização de ferramentas próprias para extrair os dados, como *log2timeline* [Guðjónsson 2010]. Para a realização do exame, supõe-se que o formato dos dados esteja todo corretamente configurado.

4.4 Contextualização

Após a extração dos registros temporais de interesse, é iniciada a etapa de contextualização. Essa etapa visa vincular um registro temporal a um ou mais contextos. Cada contexto é composto por quatro dimensões: assunto, evento, pessoa e local.

4.4.1 Assunto

A dimensão assunto diz respeito a temas ou palavras-chave relacionadas à investigação. Para isso, devem ser definidos os assuntos segundo o crime investigado, onde cada assunto é composto por um conjunto de palavras-chave. Os registros temporais são associados a um assunto se o conteúdo do arquivo possuir uma ou mais dessas palavras.

A informação sobre os termos que compõe o assunto pode vir por meio de informações da equipe de investigação de forma manual, por meio de declarações de vítimas, por meio de criminosos em busca do abrandamento da pena (delação premiada) ou por meio

de técnicas de processamento de linguagem natural de forma automática, que conseguem identificar assuntos minerando o conteúdo dos textos extraídos das fontes de evidência.

A construção dos termos que irão compor o arquivo de entrada para o processo de clusterização pode ser feita, além do esforço manual, de forma automática por processadores de textos e buscas de ocorrência probabilística das palavras. Os pesquisadores [Steyvers e Griffiths 2007] informam que técnicas estatísticas podem ser usadas para inferir o conjunto de tópicos que foram responsáveis pela geração de uma coleção de documentos.

Cada tópico é interpretável individualmente, fornecendo uma distribuição de probabilidade sobre palavras que seleciona em um conjunto coerente de termos correlacionados. No trabalho realizado por [Steyvers e Griffiths 2007], o modelo proposto é fundamentado em regras de amostragem probabilística simples que descrevem como as palavras em documentos podem ser geradas com base em variáveis latentes (aleatórias). A figura 4.3 ilustra o princípio.

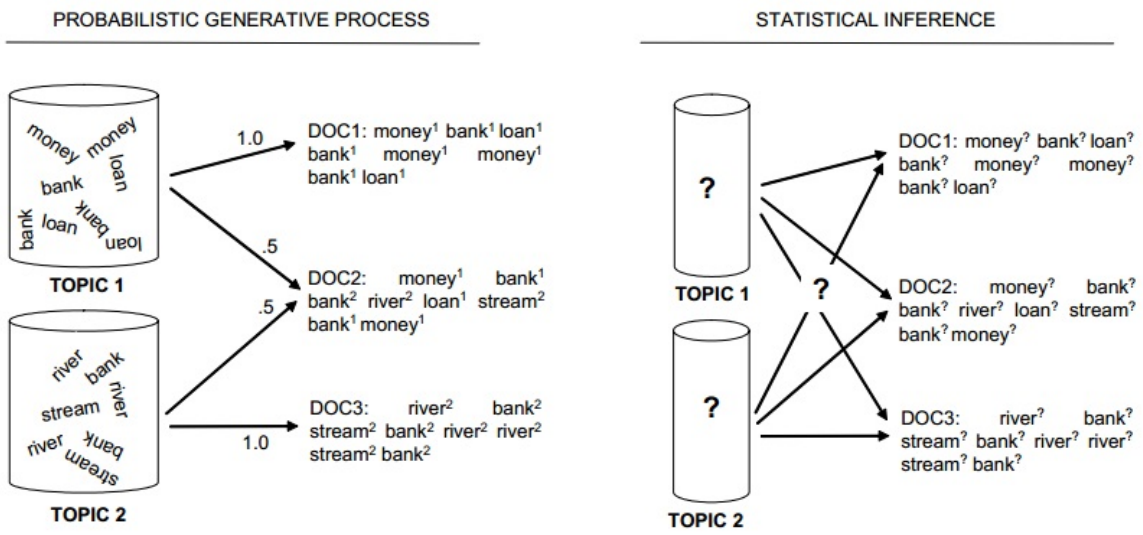


Figura 4.3: Processo de geração probabilística e o problema da inferência estática [Steyvers e Griffiths 2007]

Assim, na figura 4.3 os tópicos 1 e 2 são tematicamente relacionados ao dinheiro e aos rios e são ilustrados como sacolas contendo distribuições diferentes sobre as palavras. Diferentes documentos podem ser produzidos escolhendo palavras de um tópico dependendo do peso dado ao tópico. Por exemplo, os documentos 1 e 3 foram gerados por amostragem somente do tópico 1 e 2, respectivamente, enquanto o documento 2 foi gerado por uma mistura igual dos dois tópicos. Observe que os números sobrescritos associados às palavras em documentos indicam qual tópico foi usado para amostrar a palavra.

Da forma como o modelo é definido, não há nenhuma noção de exclusividade mútua que restringe as palavras para ser parte de um único tópico. Isso permite que os modelos de

tópicos capturem palavras polissêmicas, ou seja, palavras que possuem vários significados. Por exemplo, o dinheiro e o tópico do rio podem dar a probabilidade elevada à palavra "BANK", que é sensível devido à natureza polissêmica da palavra, pois na língua inglesa, a palavra "BANK" pode significar banco ou estar contextualizada com "River bank", que significa margem de rio.

4.4.2 Evento

A dimensão evento diz respeito a eventos relevantes para a investigação, ocorridos digitalmente (ex.: troca de uma mensagem) ou no mundo real (ex.: homicídio) e associada manualmente. Um evento pode ou não ter um momento bem definido no tempo. Nesse caso, os registros temporais podem ser associados ao evento pela proximidade (ex.: logs obtidos após a detecção de uma invasão ao sistema).

Registros temporais (eventos de baixo nível) também podem ser vinculados automaticamente a um evento (de alto nível), como sugerido por [Hargreaves e Patterson 2012]. Por exemplo, a detecção de alteração em vários arquivos do sistema, pode ser vinculado à instalação de um programa malicioso. O perito pode utilizar no contexto evento, os registros das ações do sistema operacional que registram, por exemplo, quando um login não é realizado com sucesso devido a falta de autorização ou permissão. Para a definição deste contexto, os logs de serviços de rede e dos computadores são muito utilizados como logs de *firewalls*, IDSs, etc.

Em etapas mais elaboradas, pode-se utilizar a ferramenta proposta por Yu Jin [Jin 2013], no qual o autor utiliza os mapas auto-organizáveis [Kohonen 1982] para identificar as relações entre os vários serviços em sistemas operacionais Android. Mapas auto-organizáveis, também conhecido pela sigla inglesa SOM (*Self-Organising Map*), é um modelo de agrupamento no qual se baseia nas redes de competição [Silva 1998]. Tem como característica o uso de treinamento não supervisionado, no qual busca-se encontrar similaridades não conhecidas. A arquitetura do modelo é uma rede com entradas, aplica-se pesos nas conexões e tem-se a formação de grupos como saída. Nos mapas auto-organizáveis, a medida que novos dados entram na rede, eles podem ser classificados com o aprendizado gerado anteriormente. Assim, [Jin 2013] cria uma estrutura de dados para interpretar as relações entre os eventos. A estrutura é então disponibilizada para visualização, ajudando o perito a visualizar relações e conexões entre eventos ainda não conhecidos.

4.4.3 Pessoa

Em uma investigação digital, várias pessoas podem estar envolvidas. Logo, registros temporais provenientes de fontes diversas estarão associados a pessoas distintas. Um registro temporal pode ser associado a uma pessoa não só por ser o usuário principal de um determinado dispositivo, mas também pela interação por meio de muitas outras atividades.

Pode-se extrair nomes dos registros das chamadas telefônicas de um aparelho celular, associar eventos temporais a determinados nomes por meio da coleta das contas de e-mail de determinados dispositivos, nomes provindos das contas de usuário das páginas de redes sociais configuradas nos equipamentos apreendidos (Facebook, Instagram, Snapchat, LinkedIn, Twitter, etc), lista de contatos de programas de mensagens instantâneas como Whatsapp, Skype, Facebook messenger, Viber, Hangouts, Telegram e outros identificadores associados às pessoas de interesse e vinculá-los automaticamente. Podem ser associados nomes de suspeitos que as equipes de investigação fornecem, vinculando-os de forma manual.

A vinculação de registros temporais pode ser feita utilizando-se ferramentas capazes de coletar os atributos de arquivos. No caso de arquivos de documentos textos (criados pelo aplicativo Microsoft Office Word, por exemplo), o metadado que armazena o campo "autor" poderia ser extraído e vinculado no contexto pessoa. O pesquisador [Júnior 2012] propõe o uso da técnica de Reconhecimento de Entidades Mencionadas (REM) com o objetivo de identificar e classificar nomes de entidades contidas em um texto não estruturado. Conforme o autor, são mais comumente encontrados nos textos submetidos à análise os termos relacionados a pessoas, organizações, local, tempo e também nomes relacionados a contextos muito especializados como, por exemplo, nomes de proteína e enzimas no campo da Biomedicina. A figura 4.4 ilustra um exemplo da atuação do REM aplicado a um texto.

As **<ORG>**Organizações Pedras Preciosas LTDA**</ORG>** foram vendidas para o **<PES>**Sr. Fulano dos Santos Jr.**</PES>** por **<VAL>**R$200.000,00**</VAL>**, o que levou à mudança da sua sede de **<LOC>**São Paulo**</LOC>** para o **<LOC>**Rio de Janeiro**</LOC>** em **<TPO>**2011**</TPO>**.

Figura 4.4: Exemplo da atuação do REM aplicado a um texto [Júnior 2012]

4.4.4 Local

A dimensão do local associa o registro temporal a um lugar no espaço. Por exemplo, o local de ocorrência de um determinado crime ou locais frequentados pelos suspeitos.

Para vincular um registro temporal a um determinado local, pode-se utilizar metadados de arquivos de imagens de câmera fotográficas e telefones celulares, coordenadas de GPS extraídas de aplicativos, histórico de navegação de sites de mapas, entre outros. O local pode ser vinculado manualmente, por meio da identificação de um lugar conhecido (ex.: suspeito aparece em uma foto em frente a Congresso Nacional) ou de forma automática, utilizando ferramentas de extração de coordenadas GPS.

O termo metadados refere-se a informações sobre algum dado [Carrier 2005]. O pesquisador [Cohen 2007] descreve a técnica para extração de informações em imagens DSC (*Digital Still Camera*). O autor cita a utilização de software forense para a aquisição dos arquivos de imagens, protegendo os blocos contra a sobrescrita.

Muitas informações podem ser coletadas dos metadados das imagens no formato EXIF (Exchangeable image file). Para atender a proposta deste trabalho, nos atemos a possibilidade da coleta das informações de coordenadas do GPS contidas nos arquivos das imagens no formato EXIF. O Diretório de Arquivo de Imagem (IFD) é o local onde os metadados dos arquivos EXIF estão armazenados. O autor revela que o IFD é dividido em três seções: EXIF IFD, GPS IFD, e Interoperability IFD. A seção que interessa a este trabalho é a seção GPS IFD. A figura 4.5 ilustra a estrutura de um GPS IFD.

O pesquisador [Sousa e Gondim 2016] propõe um método de recuperação de mensagens com coordenadas GPS armazenadas na memória RAM de dispositivos móveis Android. O estudo faz uso do protocolo NMEA (*National Marine Eletronics Association*), que é o mecanismo de comunicação padrão entre os receptores GPS e os drivers dos dispositivos receptores na arquitetura Android [Sousa e Gondim 2016]. Para a coleta do posicionamento, os autores citam as mensagens NMEA com maior relevância: GPGGA e GPRMC. A figura 4.6 ilustra as mensagens com as coordenadas sublinhadas.

O local pode revelar a ocorrência de um evento relevante, como, por exemplo, um estupro de vulnerável que ocorre em uma determinada escola. Os vestígios podem indicar que o suspeito morava próximo a escola, visitava as redondezas e trocava mensagens nessa área.

- **Geração do arquivo de entrada**

A qualquer momento no curso da investigação, novos registros podem ser acrescentados e contextualizados. Novas pessoas podem ser inseridas no processo, um local associado a um investigado descoberto posteriormente, novos assuntos e eventos. Caso os novos registros sejam adicionados, todo o processo deve ser realizado novamente. Em estudos futuros, pretende-se examinar o percentual dos registros que foram ou não clusterizados.

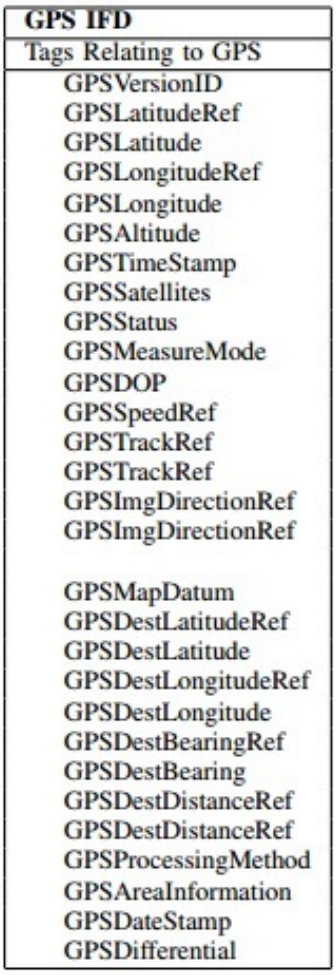


Figura 4.5: Exemplo da estrutura de um GPS IFD [Cohen 2007]

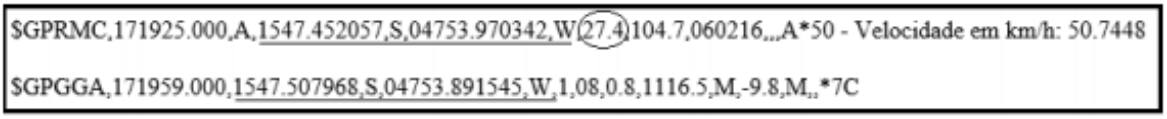


Figura 4.6: Exemplo das mensagens NMEA com as coordenadas sublinhadas [Sousa e Gondim 2016]

Após a definição dos contextos, é realizada a geração do arquivo de entrada que será submetido ao algoritmo de clusterização. Este arquivo contém todas as informações já contextualizadas, formatadas e preparadas para o processamento por meio do algoritmo de agrupamento.

Para a proposta deste trabalho, o arquivo de entrada é configurado para o formato ARFF (https://weka.wikispaces.com/ARFF+%28stable+version%29), formato este utilizado para o software de clusterização *WEKA*, usado para os experimentos neste trabalho.

4.5 Clusterização

A etapa de clusterização é o processo pelo qual os dados são submetidos a um algoritmo de agrupamento. Na aprendizagem de máquina, um cluster é o agrupamento de dados de certo conjunto de dados de entrada (arquivo gerado na última fase da etapa de contextualização). Esses agrupamentos são realizados por meio de características similares (dentro de um mesmo grupo) ou por características não similares (os pontos de um cluster são diferentes dos pontos de outro cluster). Como resultado do algoritmo, cada registro é associado a um cluster. A quantidade de clusters formados varia conforme a quantidade de entidades a preencher em cada um dos contextos. Aumentar o número de contextos fará com que o número de clusters também aumente. Neste trabalho, o algoritmo de clusterização é o *k-means*. O *k-means* utiliza a distância Euclidiana para o cálculo de dissimilaridades.

No WEKA, software utilizado neste trabalho, a distância Euclidiana é calculada pela proximidade dos valores nos atributos. Como exemplo, suponha que seja definido o seguinte atributo "nome" no arquivo de entrada:

```
@ATTRIBUTE nome rodrigo, carlos, katia, joana
```

O WEKA criará um vetor de 04 (quatro) posições, armazenado cada nome em uma posição do vetor. Assim, um vetor de posição v=[0], [1], [2], [3] será criado, onde "v" é o vetor. No exemplo, ao nome "rodrigo" será atribuído o valor numérico "0", ao nome "carlos" será atribuído o valor numérico "1", ao nome "katia" será atribuído o valor numérico "2" e ao nome "joana" será atribuído o valor numérico "3". De posse dos valores numéricos, a distância Euclidiana é calculada.

Após a etapa de contextualização, para a formação do arquivo de entrada, os valores ausentes foram ignorados e não participam da etapa de clusterização. No entanto, os registros podem ser visualizados na linha temporal única que contém todos os registros temporais.

Por fim, para cada cluster gerado, ordenam-se os registros por data e hora, obtendo-se assim uma linha temporal contextualizada. Logo, um número N de clusters gera N linhas temporais. De posse das várias linhas temporais, o perito é capaz de analisar atividades correlacionadas pelos contextos definidos pelas dimensões de nome, evento, local ou assunto. Como estudado, o número de clusters tem relação com o número de contextos e cada cluster gera uma linha temporal. Assim, um número muito grande de contextos pode gerar um número grande de clusters e linhas temporais, inviabilizando a aplicação do modelo. Em estudos futuros, pretende-se estudar o tamanho ideal de contextos para aplicação do modelo. Nos experimentos realizados neste trabalho, cada dimensão é composta de até cinco entidades (nomes). Esse número foi adequado para a aplicação do modelo.

Com as linhas temporais contextualizadas, o especialista pode realizar a análise em conjunto com a linha temporal única. Assim, o investigador pode observar os registros temporais da linha temporal contextualizada na linha temporal única. Essa análise permite identificar registros temporais que possam ser relevantes e que tenham ficado de fora do processo de clusterização ou até mesmo da etapa de contextualização.

4.6 Análise de linhas temporais

Após a etapa de geração das linhas temporais contextualizadas, o perito realiza a análise das linhas temporais contextuais em conjunto com a linha temporal única. Durante esta etapa, o perito examina os registros temporais da linha temporal contextualizada na linha temporal única observando todos os registros que não foram ou não puderam ser clusterizados. Em seguida, caso verifique registros temporais aptos a serem contextualizados presentes na linha temporal única e que não estavam presentes na linha temporal contextualizada, o investigador realiza novamente o processo de contextualização, inserindo os novos dados e executando novamente o processo de clusterização. Consequentemente, novos agrupamentos serão formados e novas linhas temporais contextuais serão criadas. O processo continua enquanto o perito julgar necessário para a elucidação dos fatos em apuração.

5 EXPERIMENTOS

Nos capítulos anteriores, foram discutidas as fundamentações teóricas em que o modelo proposto neste trabalho está alicerçado. Foram abordados conceitos sobre investigação digital, análise de linhas temporais e aprendizado de máquina. Neste capítulo, será abordado a aplicação do modelo em dois estudos de caso e analisados os seus resultados.

5.1 Aplicação em caso fictício

Como visto os conceitos teóricos previamente, será demonstrado a aplicação do modelo proposto em um cenário baseado em um caso fictício. No ensaio proposto, utilizou-se como algoritmo de agrupamento o k-means e como software de processamento o WEKA, embora outros algoritmos possam ser utilizados para a aplicação do modelo. O estudo da eficiência de outros algoritmos é objeto de estudos futuros.

O ensaio trata de uma quadrilha especializada em lavagem de dinheiro e corrupção. Nesse caso, havia suspeita de que dois suspeitos das condutas descritas movimentavam quantias aparentemente de origem duvidosa. Após denúncia e monitoramento dos suspeitos, foram descobertas outras pessoas associadas às condutas. Nesse cenário, os peritos foram acionados para a busca de evidências digitais no material apreendido na residência dos suspeitos e nos escritórios de trabalho, apreendendo-se computadores, celulares, HDs, pendrives e tablets. Para representar essas fontes de evidências, foi utilizado um conjunto de arquivos fictícios, simulando 21.000 registros temporais entre as datas de 01/02/2014 à 28/12/2014. Os formatos de data foram normalizados para um formato único (apresentado na figura 5.1).

5.1.1 Conjunto de dados

Na figura 5.1, é apresentado o Dataset em formato ARFF utilizado no experimento, dividido em cabeçalho e dados pela declaração “@DATA”.

5.1.2 Contextualização

Para a definição do local, podem ser utilizadas informações colhidas de metadados EXIF de imagens fotográficas de telefone celular, a vinculação manual por meio de reco-

```
@RELATION propinoduto

@ATTRIBUTE fonte    string
@ATTRIBUTE hora     DATE "yyyy/MM/dd HH:mm:ss"
@ATTRIBUTE local    {banco, escritorio, congresso}
@ATTRIBUTE nome     {rodrigo, carlos, katia, joana}
@ATTRIBUTE assunto  {pixuleco, acaraje, cascalho}
@ATTRIBUTE evento   {transacao_bancaria,whatsapp,e-mail,sms}

@DATA
Fonte_18,"2014/06/12 09:30:16",escritorio,katia,cascalho,e-mail
Fonte_5,"2014/08/28 09:15:51",escritorio,katia,acaraje,e-mail
Fonte_2,"2014/08/12 14:46:24",banco,rodrigo,cascalho,transacao_bancaria
Fonte_8,"2014/11/18 15:13:55",banco,rodrigo,acaraje,transacao_bancaria
Fonte_3,"2014/09/25 09:49:09",escritorio,katia,acaraje,e-mail
Fonte_5,"2014/03/06 09:43:46",banco,carlos,pixuleco,sms
Fonte_10,"2014/04/24 21:34:49",escritorio,?,cascalho,whatsapp
Fonte_20,"2014/10/09 09:34:16",escritorio,katia,pixuleco,e-mail
Fonte_11,"2014/10/22 11:40:47",congresso,carlos,cascalho,transacao_bancaria
Fonte_9,"2014/08/26 14:06:09",banco,rodrigo,cascalho,transacao_bancaria
Fonte_17,"2014/09/12 12:14:21",escritorio,joana,acaraje,sms
Fonte_1,"2014/06/14 08:02:23",escritorio,rodrigo,pixuleco,?
Fonte_11,"2014/07/25 18:38:16",congresso,?,cascalho,whatsapp
Fonte_17,"2014/07/03 09:14:46",congresso,?,?,transacao_bancaria
Fonte_13,"2014/03/26 15:27:24",congresso,carlos,acaraje,whatsapp
Fonte_16,"2014/08/28 09:24:36",escritorio,katia,pixuleco,e-mail
Fonte_18,"2014/06/24 13:34:14",banco,rodrigo,acaraje,transacao_bancaria
Fonte_2,"2014/11/04 11:16:31",banco,rodrigo,pixuleco,transacao_bancaria
Fonte_14,"2014/03/04 10:57:44",banco,joana,?,?
Fonte_9,"2014/02/26 15:14:14",congresso,katia,pixuleco,sms
Fonte_6,"2014/02/01 12:19:09",?,rodrigo,acaraje,transacao_bancaria
Fonte_15,"2014/06/03 13:10:31",?,rodrigo,cascalho,transacao_bancaria
Fonte_16,"2014/05/22 12:22:45",escritorio,katia,cascalho,e-mail
Fonte_5,"2014/07/25 15:24:06",banco,carlos,?,transacao_bancaria
```

Figura 5.1: Arquivo do tipo ARFF utilizado no experimento fictício

nhecimento visual (ex.: fotos do escritório do suspeito ou da fachada do escritório), dados sobre as anotações de agendas eletrônicas e especialmente informações de localização coletadas por aplicativos. No caso dos aplicativos de dispositivos móveis, os dados podem ser coletados da memória RAM do aparelho [Sousa e Gondim 2016]

Para a dimensão pessoa, os nomes podem ser adquiridos baseados na extração de informações de dispositivos móveis como agenda do aparelho celular, contatos dos aplicativos de rede social, contatos dos aplicativos de mensagens instantâneas como whatsapp, Messenger facebook, etc., contatos ou troca de mensagens de correio eletrônico (e-mail), nomes repassados pela investigação que seriam suspeitos dos crimes, nomes contidos nos metadados de arquivos (ex.: propriedade autor de um documento de texto), etc.

Na definição do evento, primeiramente, é necessário definir os eventos relevantes no crime investigado. Assim, poderia ser utilizado para a geração do contexto, informações sobre a troca de mensagens entre suspeitos. No caso em questão, foi simulado uma grande quantidade de mensagens do aplicativo Whatsapp pelo suspeito Carlos. Na simulação para o experimento, foi verificado também uma grande quantidade de imagens de comprovantes de saque, transferências, pagamentos. Todas essas ações financeiras foram caracterizadas por um só evento chamado “transação_bancaria”.

Outros eventos relevantes, devido a sua quantidade (como mensagens de e-mail e sms) foram configurados. Além disso, para a geração do contexto eventos, os termos a serem extraídos das fontes para formarem o dataset (conjunto de dados que formarão o arquivo de entrada do processo e serão submetidos ao algoritmo de clusterização) podem ser formados pelas atividades geradas pelo sistema operacional, pelos logs de eventos que registram as atividades dos aplicativos, pelos registros de navegação na web etc.

Por fim, na definição do assunto, podem ser utilizadas palavras-chave encontradas em mensagens de chat nos aparelhos celulares, processamento de linguagem natural, palavras repassadas pelos investigadores.

5.1.3 Processo de clusterização

Para a etapa de clusterização, os dados contextualizados foram inseridos na ferramenta WEKA [Bouckaert et al. 2016]. Uma amostra do arquivo de entrada, no formato ARFF, é apresentado na figura 5.1). Os atributos são definidos no início do arquivo. Entre eles, a fonte que originou o registro, o rótulo temporal (data) e as dimensões que definem o contexto (assunto, local, evento e nome). Os dados das dimensões são categóricos e os valores válidos são apresentados entre chaves na figura 5.1).

Os dados foram então submetidos ao algoritmo *Simple K-Means* utilizando a distância euclidiana para formação dos clusters. Registros não contextualizados (cujo valor faltante é representado por um ponto de interrogação) são ignorados. O parâmetro k, do número de clusters a serem gerados, depende da quantidade de valores em cada atributo (nome, assunto, local, evento) e da variância desses dados, o que depende do caso em concreto. Assim, o valor de k foi definido experimentalmente, variando seu valor entre k=2 a k=7. A tabela 5.1 mostra a formação de cluster gerado com k=2.

Tabela 5.1: Clusters formados após a execução do algoritmo *SimpleKMeans* com k=2 no ensaio simulado

		Cluster#	
Atributo	**Todos os dados** (21.000.0)	**0** (14582.0) (69%)	**1** (6418.0) (31%)
local	congresso	congresso	congresso
nome	carlos	carlos	carlos
assunto	acaraje	acaraje	pixuleco
evento	whatsapp	whatsapp	whatsapp

Em seguida, configurou-se o software com o k=3, vindo a obter o resultado exigido na tabela 5.2.

Tabela 5.2: Grupos criados após o processamento variando o valor para k=3 no ensaio simulado

		Cluster#		
Atributo	**Todos os dados** (21.000.0)	**0** (7473.0) (36%)	**1** (6144.0) (29%)	**2** (7383.0) (35%)
local	congresso	banco	escritorio	congresso
nome	carlos	rodrigo	katia	carlos
assunto	acaraje	acaraje	pixuleco	cascalho
evento	whatsapp	transacao_bancaria	e-mail	whatsapp

Percebe-se nos resultados ilustrados pela tabela 5.2 que com um k=3, houve bastante variação na formação do cluster em relação ao processamento com k=2. Observa-se que cada cluster recebe uma dimensão de local, nome, evento diferentes entre si. Posteriormente, o software foi configurado com um valor de k=4, no qual obteve-se os resultados ilustrados pela tabela 5.3.

Tabela 5.3: Agrupamentos formados após a execução do algoritmo *SimpleKMeans* com k=4 no ensaio simulado

		Cluster#			
Atributo	**Todos os dados** (21.000.0)	**0** (5026.0) (24%)	**1** (2586.0) (12%)	**2** (7779.0) (37%)	**3** (5609.0) (27%)
local	congresso	banco	banco	congresso	escritorio
nome	carlos	rodrigo	rodrigo	carlos	katia
assunto	acaraje	acaraje	pixuleco	acaraje	acaraje
evento	whatsapp	transacao_bancaria	transacao_bancaria	whatsapp	e-mail

Observa-se nos resultados ilustrados pela tabela 5.3 que com um k=4 houve a formação quase semelhante dos clusters formados com k=3, exceto pelo cluster 1, que é quase igual ao cluster 0, com pouca variação (apenas do contexto assunto). Percebe-se uma variação mais acentuada na formação dos clusters apenas no contexto assunto.

Posteriormente, o software foi configurado com um valor de k=5, no qual obteve-se os resultados ilustrados pela tabela 5.4.

Tabela 5.4: Dados agrupados variando para um k=5 no ensaio simulado

		Cluster#				
Atributo	**Todos os dados** (21.000.0)	**0** (4631.0) (22%)	**1** (2586.0) (12%)	**2** (7779.0) (37%)	**3** (5609.0) (27%)	**4** (395.0) (2%)
local	congresso	banco	banco	congresso	escritorio	banco
nome	carlos	rodrigo	rodrigo	carlos	katia	rodrigo
assunto	acaraje	acaraje	pixuleco	acaraje	acaraje	acaraje
evento	whatsapp	transacao_bancaria	transacao_bancaria	whatsapp	e-mail	whatsapp

Observa-se nos resultados ilustrados pela tabela 5.4 que com um k=5 houve a formação quase semelhante dos clusters formados com k=3, exceto pelo cluster 1, que é quase igual ao cluster 0, com pouca variação (apenas do contexto assunto) e pelo cluster 4, que

é quase igual ao cluster 0 da tabela 5.2, com pouca variação (apenas do contexto evento). Percebe-se uma variação mais acentuada na formação dos clusters apenas no contexto assunto. Posteriormente, o software foi configurado com um valor de k=6, no qual obteve-se os resultados ilustrados pela tabela 5.5.

Tabela 5.5: Clusters formados escolhendo-se um k=6 no ensaio simulado

		Cluster#					
Atributo	**Todos os dados** (21.000.0)	**0** (2536.0) (12%)	**1** (2586.0) (12%)	**2** (7779.0) (37%)	**3** (5609.0) (27%)	**4** (395.0) (2%)	**5** (2095.0) (10%)
local	congresso	banco	banco	congresso	escritorio	banco	banco
nome	carlos	rodrigo	rodrigo	carlos	katia	rodrigo	rodrigo
assunto	acaraje	acaraje	pixuleco	acaraje	acaraje	acaraje	cascalho
evento	whatsapp	transacao_bancaria	transacao_bancaria	whatsapp	e-mail	whatsapp	transacao_bancaria

Observa-se nos resultados ilustrados pela tabela 5.5 que com um k=6 houve a formação quase semelhante dos clusters formados com k=3, exceto pelo cluster 1, cluster 4 e cluster 5. A formação desses clusters são variações dos dados do cluster 0 mostrado na tabela 5.2. Posteriormente, o software foi configurado com um valor de k=7, no qual obteve-se os resultados ilustrados pela tabela 5.6.

Tabela 5.6: Clusters formados após a execução do algoritmo *SimpleKMeans* com k=7 no ensaio simulado

		Cluster#						
Atributo	**Todos os dados** (21.000.0)	**0** (2536.0) (12%)	**1** (2586.0) (12%)	**2** (4886.0) (23%)	**3** (5751.0) (27%)	**4** (581.0) (3%)	**5** (1835.0) (9%)	**6** (2825.0) (13%)
local	congresso	banco	banco	congresso	escritorio	banco	banco	congresso
nome	carlos	rodrigo	rodrigo	carlos	katia	rodrigo	rodrigo	carlos
assunto	acaraje	acaraje	pixuleco	acaraje	acaraje	acaraje	cascalho	acaraje
evento	whatsapp	transacao_bancaria	transacao_bancaria	whatsapp	e-mail	whatsapp	transacao_bancaria	whatsapp

Observa-se nos resultados ilustrados pela tabela 5.6 que com um k=7 houve a formação quase semelhante dos clusters formados com k=3, exceto pelo cluster 1, cluster 4, cluster 5 e cluster 6. A formação desses clusters são variações dos dados do cluster 0 da tabela 5.2, exceto o cluster 6, variação do cluster 1 da tabela 5.2. Assim, obteve-se um conjunto de clusters satisfatório com k=3, pois para valores de k > 3, a variação era pequena.

Os clusters obtidos são apresentados na tabela 5.2, com a quantidade de registros em cada cluster e o contexto associado. O cluster 0, por exemplo, tem 7473 registros e o seu contexto é definido como [assunto=acaraje, local=banco, evento=Transação_bancaria, nome=Rodrigo].

A figura 5.2 apresenta as linhas temporais contextuais construídas a partir do clusters gerados. O eixo X apresenta a data em milissegundos (com a data correspondente inserida na figura) e cada linha do eixo Y representa um dos clusters. Cada ponto representa um registro temporal.

É possível verificar alguns pontos de maior atividade como nos dias 11/03/2014, 25/03/2014, 08/04/2014, 22/04/2014, 06/05/2014, 20/05/2014, 10/06/2014, 24/06/2014,

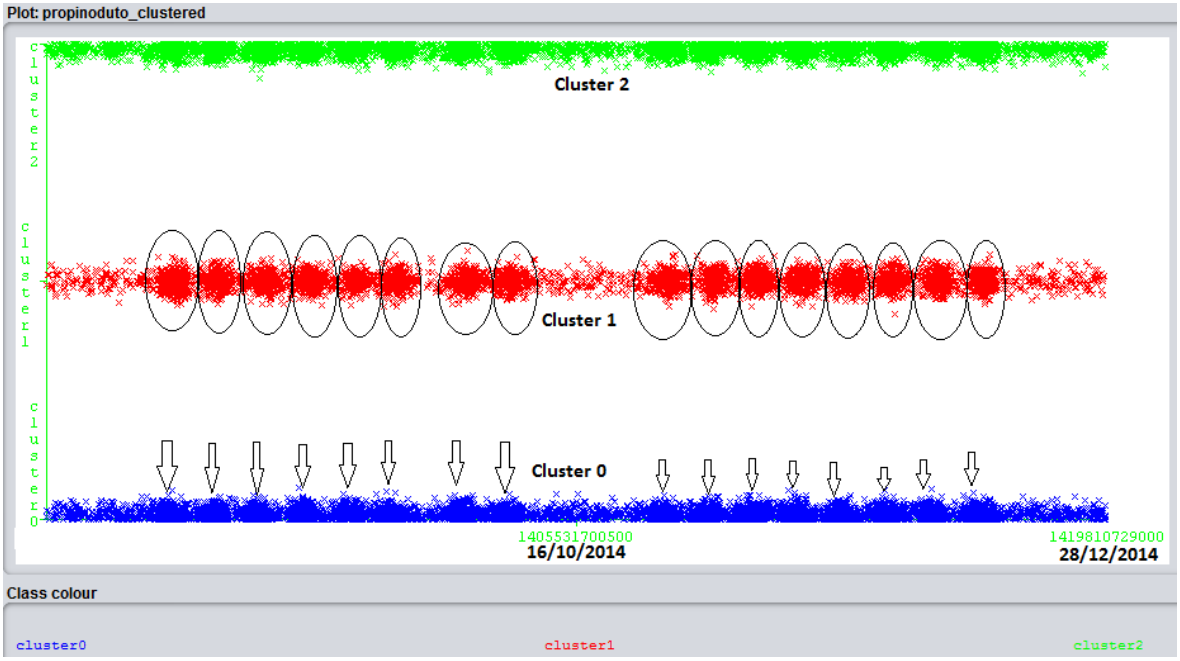


Figura 5.2: Linha temporal contextual no ensaio simulado

12/08/2014, 26/08/2014, 09/09/2014, 23/09/2014, 07/10/2014, 21/10/2014, 04/11/2014, 18/11/2014, ou seja, sempre às terças-feiras e nos dias 13/03/2014, 27/03/2014, 10/04/2014, 24/04/2014, 08/05/2014, 22/05/2014, 12/06/2014, 26/06/2014, 14/08/2014, 28/08/2014, 11/09/2014, 25/09/2014, 09/10/2014, 23/10/2014, 06/11/2014, 20/11/2014, sempre às quartas-feiras. Nos clusters 0 e 1, esses pontos foram destacados com um círculo e uma seta respectivamente.

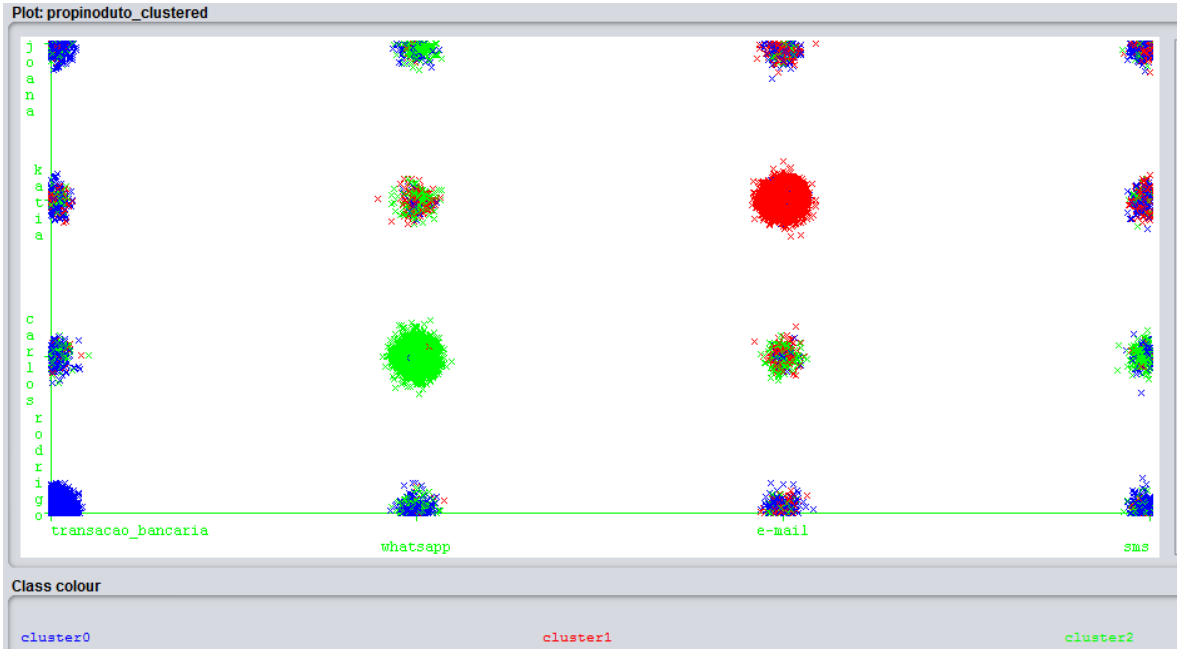


Figura 5.3: Linha temporal contextual na dimensão evento x nome no ensaio simulado

Com a análise das linhas temporais contextuais, verificou-se grande atividade do sus-

peito Rodrigo na prática do evento transacao_bancaria. É possível também verificar muita atividade entre os interlocutores Rodrigo e Carlos, em especial associada ao evento “whatsapp”, sendo intenso a prática desse evento relacionado ao suspeito Carlos.

Observa-se muita atividade entre os interlocutores Rodrigo e Karla, em especial associada ao evento “e-mail”, sendo intenso a prática desse evento relacionado a suspeita Karla.

Assim, a análise das linhas temporais contextualizadas facilita a visualização de evidências associada aos diversos eventos, em especial referente às transações bancárias realizadas por Rodrigo conforme ilustrado na figura 5.3.

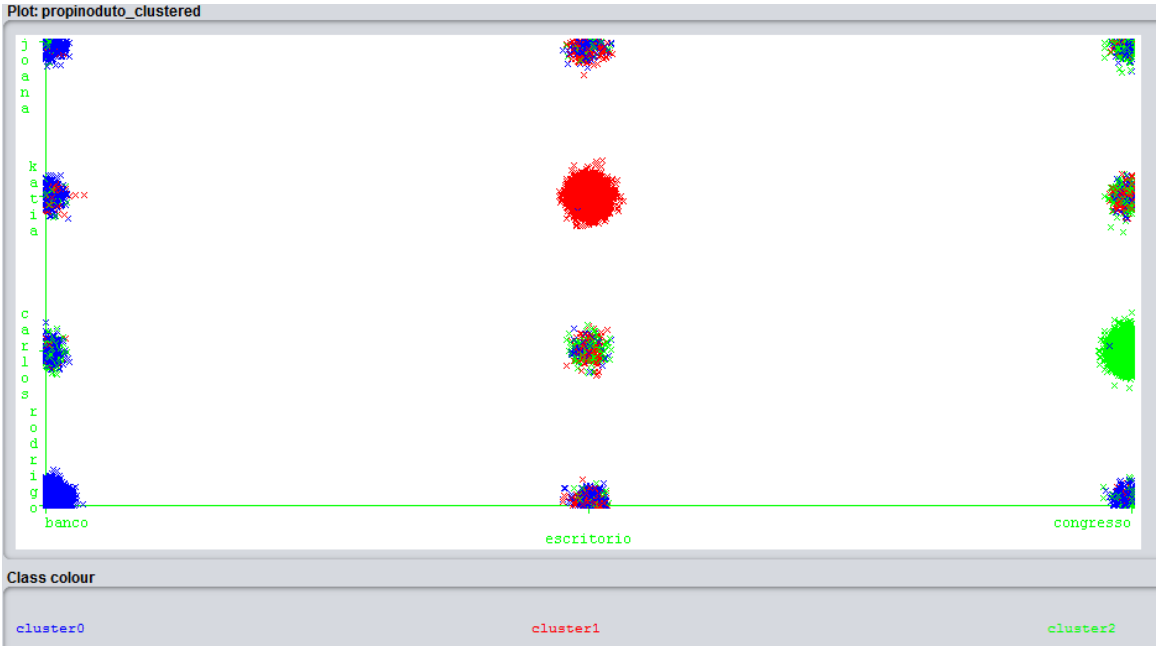


Figura 5.4: Linha temporal contextual na dimensão local x nome no ensaio simulado

Com a análise das linhas temporais contextuais, verificou-se grande atividade do suspeito Rodrigo relacionado ao local banco, da suspeita Karla relacionado ao local escritório e do suspeito Carlos ao local congresso, conforme ilustrado na figura 5.4.

5.2 Aplicação baseada em caso real

O objetivo desta seção é exemplificar a aplicação da proposta em um cenário baseado em um caso real de posse e o compartilhamento de material pornográfico contendo crianças ou adolescentes. Nesse caso, o suspeito da conduta descrita, também dava aulas sempre às segundas-feiras em uma escola, não sendo descartada a hipótese de abuso sexual de alunos. Após denúncia e monitoramento do suspeito, foram descobertas outras pessoas associadas às condutas. Os peritos foram acionados para a busca de evidências digitais no material apreendido na residência dos suspeitos e na escola: computadores, celulares e câmeras di-

gitais. Para representar essas fontes de evidências, foi utilizado um conjunto de arquivos fictícios, simulando 11.500 registros temporais entre as datas de 01/10/2014 à 25/11/2014. Os formatos de data foram normalizados para um formato único (apresentado na figura 5.5).

5.2.1 Contextualização

Para a definição do local, foram utilizadas informações colhidas de metadados EXIF de imagens de câmera fotográfica e telefone celular, vinculação manual por meio de reconhecimento visual (ex.: fotos da casa do suspeito ou da fachada da escola) e informações de localização coletadas por aplicativos. Na ausência de um local mais específico, o registro foi associado ao próprio dispositivo originário (ex.: celular) ou ao local no qual a fonte de evidências se localizava ou ser indeterminada (ex.: uma foto de abuso sem identificação clara do local).

Para a dimensão pessoa, os nomes foram reunidos baseado na extração de informações das trocas de mensagens de aplicativos de chat, nomes repassados pela investigação que seriam suspeitos dos crimes, nomes contidos nos metadados de arquivos (ex.: propriedade autor de um documento de texto).

Na definição do evento, primeiramente, foram definidos eventos relevantes no crime investigado. Por exemplo, a troca de mensagens entre suspeito e vítima. No caso em questão, foi observada uma grande quantidade de fotografias do abuso, downloads de arquivos contendo pornografia infantil e troca de mensagens entre os suspeitos. Por fim, na definição do assunto, foram utilizadas palavras-chave encontradas nos itens de pesquisa para download de arquivos de pornografia infantil.

5.2.2 Clusterização

Para a etapa de clusterização, os dados contextualizados foram inseridos na ferramenta WEKA [Bouckaert et al. 2016]. Uma amostra do arquivo de entrada, no formato ARFF, é apresentado na figura 5.5.

Os atributos são definidos no início do arquivo. Entre eles, a fonte que originou o registro, o rótulo temporal (data) e as dimensões que definem o contexto (assunto, local, evento e nome). Os dados das dimensões são categóricos e os valores válidos são apresentados entre chaves na figura 5.5.

Os dados foram então submetidos ao algoritmo *Simple K-Means* utilizando a distância euclidiana para formação dos clusters. Registros não contextualizados (cujo valor faltante

```
@RELATION caso_pi

@ATTRIBUTE fonte_dos_dados    string
@ATTRIBUTE data     DATE "yyyy/MM/dd HH:mm:ss"
@ATTRIBUTE assunto  {PTHC, PEDO, SEXO}
@ATTRIBUTE local    {escola, notebook1, notebook2, celular, escritorio}
@ATTRIBUTE evento   {Tirar_foto,chat_whatsapp,chat_messenger_facebook,download_arquivos}
@ATTRIBUTE nome     {Rodrigo, Carlos, Lucas, Frederico, Sade}

@DATA
Fonte_10,"2014/010/05 22:27:26",SEXO,celular,chat_messenger_facebook,Rodrigo
Fonte_3,"2014/010/010 09:19:55",?,notebook2,chat_messenger_facebook,Rodrigo
Fonte_4,"2014/010/03 10:13:47",PEDO,celular,chat_messenger_facebook,Carlos
Fonte_8,"2014/010/017 18:39:33",?,notebook1,download_arquivos,Carlos
Fonte_7,"2014/010/015 12:39:02",?,notebook2,?,?
Fonte_6,"2014/010/021 20:37:53",PEDO,notebook2,download_arquivos,Rodrigo
Fonte_10,"2014/010/023 10:31:54",PTHC,escritorio,download_arquivos,Carlos
Fonte_5,"2014/010/011 17:15:31",PTHC,escritorio,download_arquivos,Rodrigo
Fonte_10,"2014/010/01 02:08:50",PTHC,notebook1,download_arquivos,Rodrigo
Fonte_6,"2014/010/021 17:17:51",SEXO,escritorio,download_arquivos,Carlos
Fonte_6,"2014/010/017 22:13:56",SEXO,notebook2,chat_messenger_facebook,Rodrigo
Fonte_1,"2014/010/015 03:24:28",PEDO,escritorio,download_arquivos,Rodrigo
Fonte_8,"2014/010/019 12:26:08",PEDO,notebook2,download_arquivos,Carlos
Fonte_2,"2014/010/06 14:32:09",PEDO,escritorio,chat_whatsapp,Rodrigo
Fonte_2,"2014/010/015 07:13:51",PTHC,celular,chat_messenger_facebook,Carlos
Fonte_4,"2014/010/04 07:58:37",PEDO,escritorio,chat_whatsapp,Carlos
Fonte_8,"2014/010/018 03:52:06",PTHC,escritorio,download_arquivos,Carlos
Fonte_5,"2014/010/022 04:55:42",SEXO,notebook2,chat_whatsapp,Rodrigo
```

Figura 5.5: Arquivo do tipo ARFF utilizado no experimento do estudo de caso real

é representado por um ponto de interrogação) são ignorados. O parâmetro *k*, do número de clusters a serem gerados, depende da quantidade de valores em cada atributo (nome, assunto, local, evento) e da variância desses dados, o que depende do caso em concreto. Assim, o valor de k foi definido experimentalmente, variando seu valor entre k=2 a k=7. A tabela 5.7 mostra a formação de cluster gerado com k=2.

Tabela 5.7: Clusters formados após a execução do algoritmo *SimpleKMeans* com k=2 no estudo de caso real

		Cluster#	
Atributo	**Todos os dados** (11499.0)	**0** (7795.0) (68%)	**1** (3704.0) (32%)
assunto	pedo	sexo	pthc
local	escola	escola	notebook2
evento	tirar_foto	tirar_foto	chat_whatsapp
nome	rodrigo	rodrigo	carlos

Em seguida, configurou-se o software com o k=3, vindo a obter o resultado exigido na tabela 5.8. Percebe-se nos resultados ilustrados pela tabela 5.8 que com um k=3, houve bastante variação na formação do cluster em relação ao processamento com k=2. Observa-se que cada cluster recebe uma dimensão de local, nome, evento e assunto diferentes entre si.

Tabela 5.8: Dados clusterizados com k=3 no estudo de caso real

		Cluster#		
Atributo	**Todos os dados** (11499.0)	**0** (6904.0) (60%)	**1** (3237.0) (28%)	**2** (1358.0) (12%)
assunto	pedo	sexo	pthc	pedo
local	escola	escola	notebook2	escritorio
evento	tirar_foto	tirar_foto	chat_whatsapp	chat_whatsapp
nome	rodrigo	rodrigo	carlos	sade

Posteriormente, o software foi configurado com um valor de k=4, no qual obteve-se os resultados ilustrados pela tabela 5.9. Com um k=4, ainda houve variação na formação do cluster em relação ao processamento com k=3. Observa-se que cada cluster recebe uma dimensão de local, nome, evento e assunto bem diverso em relação a cada cluster.

Tabela 5.9: Resultado do agrupamento com k=4 no estudo de caso real

		Cluster#			
Atributo	**Todos os dados** (11499.0)	**0** (6106.0) (53%)	**1** (2552.0) (22%)	**2** (1324.0) (12%)	**3** (1517.0) (13%)
assunto	pedo	sexo	pthc	pedo	pthc
local	escola	escola	celular	escritorio	notebook2
evento	tirar_foto	tirar_foto	chat_whatsapp	chat_whatsapp	chat_messenger_facebook
nome	rodrigo	rodrigo	carlos	sade	carlos

O software foi então configurado com um valor de k=5, no qual obteve-se os resultados ilustrados pela tabela 5.10. Com um k=5, continua havendo variação na formação do cluster em relação ao processamento com k=4. Observa-se que cada cluster recebe uma dimensão de local, nome, evento e assunto bem diverso em relação a cada cluster.

Tabela 5.10: Clusters formados variando o valor para um k=5 no estudo de caso real

		Cluster#				
Atributo	**Todos os dados** (11499.0)	**0** 5765.0) (50%)	**1** (2163.0) (19%)	**2** (1297.0) (11%)	**3** (1405.0) (12%)	**4** (869.0) (8%)
assunto	pedo	sexo	pthc	pedo	pthc	sexo
local	escola	escola	notebook1	escritorio	notebook2	celular
evento	tirar_foto	tirar_foto	chat_whatsapp	chat_whatsapp	chat_messenger_facebook	chat_whatsapp
nome	rodrigo	rodrigo	carlos	sade	carlos	carlos

Posteriormente, o software foi configurado com um valor de k=6, no qual obteve-se os resultados ilustrados pela tabela 5.11. Enfim, observa-se nos resultados ilustrados pela tabela 5.11 que com um k=6 houve a formação quase semelhante dos clusters formados com k=5, exceto pelo cluster 5. A formação desse cluster é uma variação dos dados do cluster 0 mostrado na tabela 5.10.

Tabela 5.11: Agrupamento gerado com k=6 no estudo de caso real

		Cluster#					
Atributo	**Todos os dados** (11499.0)	**0** 4000.0) (35%)	**1** (2013.0) (18%)	**2** (1328.0) (12%)	**3** (959.0) (8%)	**4** (1404.0) (12%)	**5** (1795.0) (16%)
assunto	pedo	sexo	pthc	pedo	pthc	sexo	pedo
local	escola	escola	notebook1	escritorio	notebook2	celular	escola
evento	tirar_foto	tirar_foto	chat_messenger_facebook	chat_messenger_facebook	chat_messenger_facebook	chat_whatsapp	tirar_foto
nome	rodrigo	rodrigo	carlos	sade	carlos	carlos	rodrigo

Posteriormente, o software foi configurado com um valor de k=7, no qual obteve-se os resultados ilustrados pela tabela 5.12.

Tabela 5.12: Clusters formados após a execução do algoritmo *SimpleKMeans* com k=7 no estudo de caso real

		Cluster#						
Atributo	**Todos os dados** (11499.0)	**0** 2375.0) (21%)	**1** (2013.0) (18%)	**2** (1328.0) (12%)	**3** (959.0) (8%)	**4** (1404.0) (12%)	**5** (1795.0) (16%)	**6** (1625.0) (14%)
assunto	pedo	sexo	pthc	pedo	pthc	sexo	pedo	pthc
local	escola	escola	notebook1	escritorio	notebook2	celular	escola	escola
evento	tirar_foto	tirar_foto	chat_messenger_facebook	chat_messenger_facebook	chat_messenger_facebook	chat_whatsapp	tirar_foto	tirar_foto
nome	rodrigo	rodrigo	carlos	sade	carlos	carlos	rodrigo	rodrigo

Observa-se nos resultados ilustrados pela tabela 5.12 que com um k=7 houve a formação quase semelhante dos clusters formados com k=5, exceto pelo cluster 5 e cluster 6. A formação desses clusters são variações dos dados do cluster 0 da tabela 5.10. Assim, obteve-se um conjunto de clusters satisfatório com k=5, pois para valores de k > 5, a variação era pequena. Logo, o valor de k foi definido experimentalmente, variando seu valor até obter um conjunto de clusters satisfatório com k=5.

Os clusters obtidos são apresentados na tabela 5.10, com a quantidade de registros em cada cluster e o contexto associado. O cluster 0, por exemplo, tem 5765 registros e o seu contexto é definido como [assunto=sexo, local=escola, evento=Tirar_foto, nome=Rodrigo].

A figura 5.6 apresenta as linhas temporais contextuais construídas a partir do clusters gerados. O eixo X apresenta a data em milissegundos (com a data correspondente inserida na figura) e cada linha do eixo Y representa um dos clusters. Cada ponto representa um registro temporal. É possível verificar alguns pontos de maior atividade como nos dias 13/10, 20/10, 27/10, 25/11. No cluster 0, esses pontos foram destacados com um círculo. Observa-se também um tempo de inatividade entre 28/10 e 24/11.

Com a análise das linhas temporais contextuais, verificou-se grande atividade entre os interlocutores Rodrigo e Carlos, em especial associada aos eventos “chat_whatsapp” e “chat_messenger_facebook”. O suspeito Rodrigo também teve maior atividade nos dias citados anteriormente.

Assim, a linha temporal proveniente do primeiro cluster (Cluster 0) facilita a visualização de evidências associadas ao abuso da vítima na escola exatamente nos dias mencionados, associados às fotos tiradas. Observou-se que as demais linhas permitem uma visualização melhor de mensagens utilizando WhatsApp e Facebook Messenger, usadas no

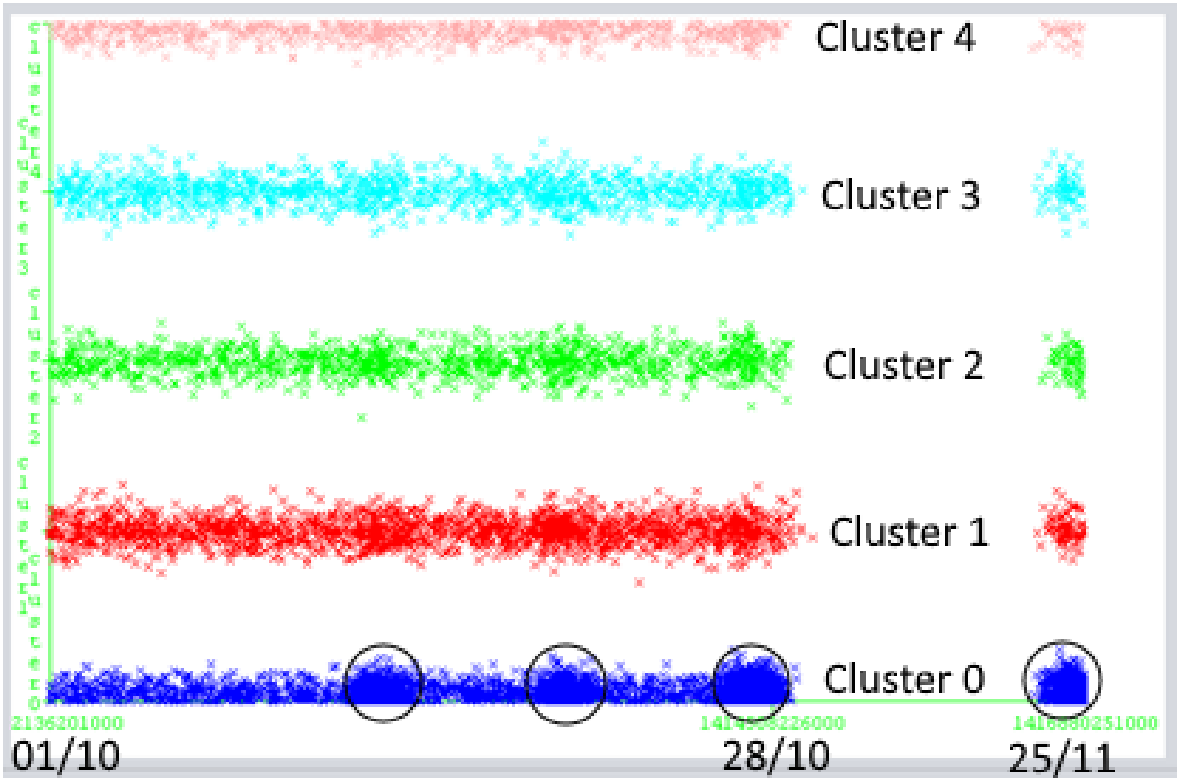


Figura 5.6: Linha temporal contextual no estudo de caso real

compartilhamento de material pornográfico infantil.

6 CONCLUSÃO

A análise de linhas temporais é uma técnica muito empregada em investigações digitais. A grande quantidade de fontes de dados temporais torna essa análise mais complexa, dificultando a adequada visualização das relações entre eventos e a determinação de pontos no tempo que são de interesse da investigação.

O presente trabalho demonstrou que essa análise pode ser simplificada pela implantação do conceito da contextualização e posterior agrupamento (clusterização) dos registros temporais. Como resultado, foi proposto um modelo para a criação de linhas temporais contextuais em que são obtidas linhas temporais cujos registros apresentam maior similaridade contextual entre si, reduzindo a interferência de outros registros não relacionados. Em conjunto com as linhas temporais contextuais, o investigador examina a linha temporal única, no qual contém todos os registros temporais das fontes examinadas. Na análise conjunta das linhas temporais contextuais e linha temporal única, o perito visualiza os registros temporais próximos aos registros que foram clusterizados, podendo participar do processo caso sejam relevantes. Portanto, o perito pode visualizar a linha temporal única tendo como base os registros obtidos nas linhas temporais contextuais.

Nos experimentos propostos, pode-se identificar com mais facilidade os suspeitos com maior interação, os momentos de maior atividade relacionados às condutas investigadas e os locais onde ou de onde as condutas criminosas foram potencialmente praticadas.

Em trabalhos futuros, pretende-se aprimorar o processo de agrupamento com a aplicação de outros algoritmos, a fim de comparar os resultados obtidos para cada um deles. Como proposta de estudos complementares, pretende-se examinar qual a percentagem dos registros clusterizados e não clusterizados. Além disso, pretende-se explorar formas de contextualização baseadas em processamento de linguagem natural, especialmente na dimensão do assunto. É foco de estudos posteriores os limites da aplicabilidade do modelo quanto ao número de contextos e a quantidade de entidades (nomes) presentes em cada um dos contextos. Dependendo do caso investigado, muitas operações de busca e apreensão podem ser realizadas, tendo como alvo os mesmos suspeitos, porém em lugares distintos. Com base nessas situações, pretende-se também estudar formas de classificação (aprendizado supervisionado) dos novos dados, uma vez que já possui a formação de clusters para o caso.

Referências Bibliográficas

[AccessData 2016] AccessData (2016). Accessdata Corp. Forensic Toolit. `http://accessdata.com/solutions/digital-forensics/forensic-toolkit-ftk`. Acessado em julho de 2016.

[Allen 1983] Allen, J. F. (1983). Maintaining Knowledge About Temporal Intervals. *ACM*, 26(11):832–843.

[Araújo 2008] Araújo, D. S. A. d. (2008). Algoritmos de Agrupamento Aplicados a Dados de Expressão Gênica de Câncer: Um Estudo Comparativo. Dissertação de Mestrado, Universidade Federal do Rio Grande do Norte, Brasil.

[Becker 1991] Becker, S. (1991). Unsupervised Learning Procedures for Neural Networks. *International Journal of Neural Systems*, 02(01n02):17–33.

[Bishop 2006] Bishop, C. (2006). *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer.

[Borges 2010] Borges, V. R. P. (2010). Comparação entre as Técnicas de Agrupamento K-Means e Fuzzy C-Means para Segmentação de Imagens Coloridas. *XII Encontro Anual de Computação (EnAComp)*.

[Bouckaert et al. 2016] Bouckaert, R. R., Frank, E., Hall, M., Kirkby, R., Reutemann, P., Seewald, A., e Scuse, D. (2016). Weka Manual for Version 3-8-0.

[Buchholz e Falk 2005] Buchholz, F. P. e Falk, C. (2005). Design and Implementation of Zeitline: a Forensic Timeline Editor. In *DFRWS*.

[Carrier 2005] Carrier, B. (2005). *File System Forensic Analysis*. Addison-Wesley Professional.

[Carrier 2010] Carrier, B. (2010). The Sleuth Kit. TSK. `http://www.sleuthkit.org/sleuthkit`.

[Cassiano 2014] Cassiano, K. M. (2014). *Análise de Séries Temporais Usando Análise Espectral Singular (SSA) e Clusterização de Suas Componentes Baseada em Densidade*. Tese de Doutorado, PUC-Rio, Brasil.

[Chabot et al. 2014a] Chabot, Y., Bertaux, A., Nicolle, C., e Kechadi, M.-T. (2014a). A Complete Formalized Knowledge Representation Model for Advanced Digital Forensics Timeline Analysis. *Digital Investigation*, 11:S95–S105.

[Chabot et al. 2014b] Chabot, Y., Bertaux, A., Nicolle, C., e Kechadi, T. (2014b). Automatic Timeline Construction for Computer Forensics Purposes. In *IEEE Joint Intelligence and Security Informatics Conference (ISI-EISIC 2014), 24-26 September, the Hague, Netherlands*. Institute of Electrical and Electronics Engineers.

[Cohen 2007] Cohen, K. (2007). Digital Still Camera Forensics. *Small Scale Digital Device Forensics Journal*, 1(1):1–8.

[Cruz 2010] Cruz, M. D. (2010). *O Problema de Clusterização Automática*. Tese de Doutorado, COPPE/UFRJ, Brasil.

[Damasceno 2010] Damasceno, M. (2010). Introdução a mineração de dados utilizando o weka. *V Congresso de Pesquisa e Inovação da Rede do Norte Nordeste de Educação Tecnológica*. Disponível: http://connepi.ifal.edu.br/ocs/index.php/connepi/CONNEPI2010/paper/viewFile/258/207. Acessado em abril de 2016.

[Encase 2016] Encase (2016). Guidance Software. Encase Forensic Software. `http://www.guidancesoftware.com/`. Acessado em julho de 2016.

[Fausett 1993] Fausett, L. V. (1993). *Fundamentals of Neural Networks: Architectures, Algorithms And Applications*. Pearson.

[Fonseca e de Castro Reis 2002] Fonseca, B. M. e de Castro Reis, D. (2002). O Fantástico Mundo da Distância de Edição. *Seminário UFMG*.

[Gantz e Reinsel 2012] Gantz, J. e Reinsel, D. (2012). The Digital Universe in 2020: Big Data, Bigger Digital Shadows, and Biggest Growth in the Far East. *IDC iView: IDC Analyze the Future*, 2007:1–16.

[Guðjónsson 2010] Guðjónsson, K. (2010). Mastering the Super Timeline With Log2timeline. *SANS Institute*.

[Hall et al. 2009] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., e Witten, I. H. (2009). The WEKA Data Mining Software: An Update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18.

[Hargreaves e Patterson 2012] Hargreaves, C. e Patterson, J. (2012). An Automated Timeline Reconstruction Approach for Digital Forensic Investigations. *Digital Investigation*, 9:S69–S79.

[Hoelz et al. 2014] Hoelz, B. W., Ruback, M. C., Silva, J. H., Melo, L., e L, K. L. (2014). *Informática Forense*. Brasília: Academia Nacional de Polícia (Caderno Didático), 2014.

[Hruschka e Ebecken 2003] Hruschka, E. R. e Ebecken, N. F. (2003). A Genetic Algorithm for Cluster Analysis. *Intelligent Data Analysis*, 7(1):15–25.

[Huang 1998] Huang, Z. (1998). Extensions to the K-means Algorithm for Clustering Large Data Sets with Categorical Values. *Data Mining and Knowledge Discovery*, 2(3):283–304.

[Jain et al. 1999] Jain, A. K., Murty, M. N., e Flynn, P. J. (1999). Data Clustering: A Review. *ACM Computing Surveys (CSUR)*, 31(3):264–323.

[Jin 2013] Jin, Y. (2013). Timeline Analysis for Android-Based Systems. Dissertação de Mestrado, Technical University of Denmark, Denmark.

[Júnior 2012] Júnior, O. D. (2012). Reconhecimento de Nomes de Pessoas e Organizações em Textos Forenses Usando uma Variação do Modelo Oculto de Markov. Dissertação de Mestrado, Universidade de Brasília, Brasil.

[Kaart e Laraghy 2014] Kaart, M. e Laraghy, S. (2014). Android Forensics: Interpretation of Timestamps. *Digital Investigation*, 11(3):234–248.

[Kaufman e Rousseeuw 2009] Kaufman, L. e Rousseeuw, P. J. (2009). *Finding groups in data: an introduction to cluster analysis*, volume 344. John Wiley & Sons.

[Ketchen e Shook 1996] Ketchen, D. J. e Shook, C. L. (1996). The Application of Cluster Analysis in Strategic Management Research: An Analysis and Critique. *Strategic Management Journal*, 17(6):441–458.

[Kohavi e Provost 1998] Kohavi, R. e Provost, F. (1998). Glossary of Terms: Special Issue on Applications of Machine Learning and the Knowledge Discovery Process. *Machine Learning*, 30(4):271–274.

[Kohonen 1982] Kohonen, T. (1982). Self-Organized Formation of Topologically Correct Feature Maps. *Biological Cybernetics*, 43(1):59–69.

[Lletı et al. 2004] Lletı, R., Ortiz, M. C., Sarabia, L. A., e Sánchez, M. S. (2004). Selecting Variables for K-means Cluster Analysis by Using a Genetic Algorithm That Optimises the Silhouettes. *Analytica Chimica Acta*, 515(1):87–100.

[Macario Filho 2015] Macario Filho, V. (2015). *Algoritmos particionais semissupervisionados com ponderação automática de variáveis*. Tese de Doutorado, Universidade Federal de Pernambuco, Brasil.

[Microsoft 2012] Microsoft (2012). Informações do Registro do Windows para Usuários Avançados. `https://support.microsoft.com/pt-br/kb/256986`. Acessado em julho de 2016.

[Mitchell 1997] Mitchell, T. M. (1997). Machine Learning. *Burr Ridge, IL: McGraw Hill*, 45:37.

[Monard e Baranauskas 2003] Monard, M. C. e Baranauskas, J. A. (2003). Conceitos Sobre Aprendizado de Máquina. *Sistemas Inteligentes-Fundamentos e Aplicações*, 1(1).

[Nassif 2012] Nassif, L. F. d. C. (2012). Técnicas de Agrupamento de Textos Aplicadas à Computação Forense. Dissertação de Mestrado, Universidade de Brasília, Brasil.

[Olsson e Boldt 2009] Olsson, J. e Boldt, M. (2009). Computer Forensic Timeline Visualization Tool. *Digital Investigation*, 6:S78–S87.

[Plaso-Wiki 2016] Plaso-Wiki (2016). Plaso Wiki. `https://github.com/log2timeline/plaso/wiki`. Acessado em julho de 2016.

[Silva 2009] Silva, D. G. e. (2009). Uso de Aprendizado de Máquina para Estimar Esforço de Execução de Testes Funcionais. Dissertação de Mestrado, Universidade Estadual de Campinas, Brasil.

[Silva 1998] Silva, L. N. d. C. (1998). Análise e Síntese de Estratégias de Aprendizado para Redes Neurais Artificiais. Dissertação de Mestrado, Universidade Estadual de Campinas, Brasil.

[Silva 2004] Silva, M. P. d. S. (2004). Mineração de Dados: Conceitos, Aplicações e Experimentos com Weka. Livro da Escola Regional de Informática Rio de Janeiro. *Sociedade Brasileira de Computação*, 1:01–24.

[Sousa e Gondim 2016] Sousa, J. P. C. e Gondim, J. J. C. (2016). Extração e Análise de Memória Volátil em Ambientes Android: Uma Abordagem Voltada à Reconstrução de Trajetórias com Base no Protocolo NMEA 0183. *XVI Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais — SBSeg 2016*, pages 727–739.

[Stacy Jr e Lunsford 2006] Stacy Jr, H. e Lunsford, P. (2006). Computer Forensics For Law Enforcement. ICTN4040 601. *http://www.infosecwriters.com/Papers/HStacy_Forensics.pdf*. Acessado em agosto de 2015.

[Stallings e Vieira 2008] Stallings, W. e Vieira, D. (2008). *Criptografia e Segurança de Redes: Princípios e Práticas*. Pearson Prentice Hall.

[Stevens 2004] Stevens, M. W. (2004). Unification of Relative Time Frames for Digital Forensics. *Digital Investigation*, 1(3):225–239.

[Steyvers e Griffiths 2007] Steyvers, M. e Griffiths, T. (2007). Probabilistic Topic Models. *Handbook of Latent Semantic Analysis*, 427(7):424–440.

[Tanenbaum 2009] Tanenbaum, A. S. (2009). *Sistemas Operacionais Modernos. Terceira Edição*. Editora Pearson Prentice Hall, São Paulo.

[Teknomo 2006] Teknomo, K. (2006). K-means Clustering Tutorial. *Medicine*, 100(4):3.

[Umale e Nilav 2014] Umale, B. e Nilav, M. (2014). Overview of K-means and Expectation Maximization Algorithm for Document Clustering. *International Journal of Computer Applications (0975-8887)*.

[Witten et al. 2011] Witten, I., Frank, E., e Hall, M. (2011). *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 3rd edition.

[Wu et al. 2008] Wu, X., Kumar, V., Quinlan, J. R., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G. J., Ng, A., Liu, B., Philip, S. Y., et al. (2008). Top 10 Algorithms in Data Mining. *Knowledge and Information Systems*, 14(1):1–37.