Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Árvores de Decisão Aplicadas À Detecção de Fraudes Bancárias

José Abílio de Paiva Ramos

Dissertação apresentada como requisito parcial para conclusão do
Mestrado Profissional em Computação Aplicada

Orientador
Prof. Dr. Guilherme Novaes Ramos

Brasília
2014

Ficha catalográfica elaborada pela Biblioteca Central da Universidade de Brasília. Acervo 1017603.

Ramos, José Abílio de Paiva.
R175a Árvores de decisão aplicadas à detecção de fraudes bancárias / José Abílio de Paiva Ramos. -- 2014.
xi, 54 f. : il. ; 30 cm.

Dissertação (mestrado) - Universidade de Brasília, Instituto de Ciências Exatas, Departamento de Ciência da Computação, Pós-Graduação em Computação Aplicada, 2014.
Inclui bibliografia.
Orientação: Guilherme Novaes Ramos.

1. Fraude. 2. Aprendizado do computador. 3. Bancos. 4. Sistemas de segurança. I. Ramos, Guilherme Novaes. II. Título.

CDU 004.056

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Árvores de Decisão Aplicadas à Detecção de Fraudes Bancárias

José Abílio de Paiva Ramos

Dissertação apresentada como requisito parcial para conclusão do
Mestrado Profissional em Computação Aplicada

Prof. Dr. Guilherme Novaes Ramos (Orientador)
CIC/UnB

Prof. Dr. Rommel Novaes Carvalho
CGU e CIC/UnB

Prof. Dr. Hércules Antônio do Prado
Embrapa Sede e UCB

Prof. Dr. Marcelo Ladeira
Coordenador do Programa de Pós-graduação em Computação Aplicada

Brasília, 25 de junho de 2014

Dedicatória

À Paula, Igor e Ana Maria.

Agradecimentos

À Paula, minha esposa, e aos meus filhos Igor e Ana Maria pela compreensão e apoio que me deram, mesmo quando não podia dar-lhes a atenção que necessitavam e mereciam.

À Karla e Gilberto que foram fundamentais para que eu tivesse disponibilidade para tornar este projeto uma realidade.

Ao Prof. Guilherme que sempre buscou orientar-me quanto ao melhor caminho seguir durante a execução desta pesquisa.

Enfim, agradeço a todos que me incentivaram e viabilizaram este trabalho.

Resumo

A oferta de produtos e serviços bancários através de canais virtuais tem aumentado nos últimos anos e, apesar do uso de diversas tecnologias de segurança, ainda existem transações fraudulentas que são concluídas com sucesso. Além disso, frequentemente os atacantes se adaptam a novas tecnologias mais rapidamente que as empresas alvejadas. Como proposta para aprimorar e agilizar as reações a fraudes, este trabalho visa indução automática de árvores de decisão a partir de amostras de dados transacionais para a identificação de transações fraudulentas. Os resultados são superiores aos alcançados pelo sistema vigente na instituição financeira, indicando que sua adoção, acompanhada de medidas reativas, podem reduzir os prejuízos financeiros, aumentar a recuperação de valores e diminuir o risco de dano à imagem da instituição, bem como o desgaste junto aos clientes.

Palavras-chave: árvores de decisão, fraudes bancárias

Abstract

The offer of products and services by banks through virtual channels has increased in recent years and, despite the use of various technologies for security, fraudulent transactions are still being successfully completed. Moreover, hackers often adapt to new technologies more quickly than the targeted companies. In order to improve and expedite responses to frauds, this work aims to identify fraudulent transactions with decision trees induced automatically from samples of transactional data. The results obtained from the proposal are better than those provided by the system currently in use in the financial institution, indicating that the use of decision trees, together with additional reactive actions, can decrease financial loss, increase asset retrieval, and reduce the risk of damage to the client-institution relationship.

Keywords: decision trees, banking frauds

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

As fraudes eletrônicas constituem um grande problema a ser combatido pelos bancos, pois as perdas no sistema financeiro brasileiro superam a cifra de um bilhão de reais anualmente [11]. Com o aumento da oferta de produtos e serviços através de canais virtuais, cada vez mais clientes aderem a estes meios para realização de suas transações financeiras e tornam-se potenciais vítimas para as quadrilhas especializadas em fraudes eletrônicas. Assim, o combate às fraudes ganhou importância nos canais *Internet/Mobile Banking* dado o aumento dos riscos associados.

Os riscos aos quais as organizações estão sujeitas são variados: associados ao usuário (comprometimento de credenciais para obtenção de informações, ataques *zero-day*[1], etc) ou associados ao meio (corrupção de aplicativos para obtenção de informação visando ataque aos clientes, ataques *man-in-the-middle*[2], etc). Estes riscos são potencializados com a crescente popularização dos meios digitais de acesso às informações bancárias.

A redução dos prejuízos decorrentes de fraudes no sistema financeiro depende da tempestiva detecção destes eventos: quanto mais tempo a instituição levar para identificar uma fraude, mais tempo o atacante terá para efetivar transações ilegítimas. Se a fraude for detectada tardiamente, o prejuízo financeiro da instituição está limitado ao saldo disponível do cliente vitimado; se a fraude não for detectada, além do prejuízo financeiro, a conta vitimada pode ser utilizada para recebimento de outros créditos fraudulentos. Este cenário motiva o contínuo aperfeiçoamento das técnicas de detecção e combate às fraudes para fazer frente à dinamicidade e sofisticação das investidas atuais.

Assim, a tarefa de identificar as transações fraudulentas dentre milhões de transações legítimas é um problema de classificação: a meta é associar automaticamente cada tran-

[1]Um ataque *zero-day* explora uma vulnerabilidade previamente desconhecida em uma aplicação computacional, geralmente é desenvolvida no mesmo dia em que é publicamente descoberta e antes que a correção seja disponibilizada aos usuários pelo respectivo fornecedor.

[2]O ataque *man-in-the-middle* consiste na interceptação da comunicação entre cliente e banco, adulteração das mensagens por um fraudador sem que os interlocutores saibam da atuação delituosa.

sação t a uma categoria $c \in \{fraude, não\text{-}fraude\}$ para que as devidas providências sejam tomadas.

Assim, um sistema de detecção de fraudes em transações bancárias deve ser acurado na identificação de transações fraudulentas, possuir boa capacidade de generalização, ser robusto a inconsistências nos dados e ser eficiente para processar grandes quantidades de transações.

1.1 Definição do Problema

Embora existam diversas tecnologias de segurança tais como encriptação dos dados, controle de acesso através de credenciais e pré-cadastramento de dispositivos, ainda existem transações fraudulentas que são concluídas com sucesso nos canais *Internet/Mobile Banking*. Neste ambiente, existem diversas ameaças que caracterizam-se pela complexidade e evolução constante, com vistas a burlar as regras de segurança das empresas. Frequentemente, a evolução tecnológica dos atacantes é mais rápida do que a evolução das empresas, podendo resultar em prejuízos relevantes para as organizações alvejadas [30].

A variabilidade dos ataques impõe constante atualização do modelo de detecção, isto é, uma vez descoberta uma nova estratégia de ataque, quanto maior for o tempo necessário à elaboração, avaliação e implantação de uma nova regra ou modelo que o identifique, maiores serão os prejuízos decorrentes desta investida.

Além disso, o grande volume de operações financeiras transacionadas nos canais exige alta eficiência na tarefa de identificação de transações com indícios de fraude para que um ataque em andamento seja rapidamente identificado e interrompido. Neste cenário, a quantidade de transações fraudulentas costuma ser muito menor que a quantidade de transações legítimas e, assim, as classes de transações legítimas e fraudulentas são muito desbalanceadas exigindo que o modelo trate este problema para não ter seu desempenho, na identificação de transações fraudulentas, prejudicado [25].

Por fim, a acurácia do modelo na identificação de transações fraudulentas é fundamental ao combate efetivo às fraudes bancárias, pois viabiliza a diminuição dos prejuízos quando da ocorrência de fraude e mitiga o desgaste com os clientes quando uma transação legítima é incorretamente classificada como fraude. Portanto, o modelo de identificação de fraudes deve ser acurado para que consiga classificar corretamente as transações fraudulentas, minimizando a quantidade de fraudes apontadas como legítimas (falsos-negativos) e, simultaneamente, deve minimizar a quantidade de transações legítimas apontadas como fraudulentas (falsos-positivos).

Assim, busca-se neste trabalho prover modelos de identificação de fraudes em transações bancárias, especificamente transferências entre contas, que sejam rapidamente adap-

táveis a novos ataques, sejam eficientes para processar grandes volumes de transações e, claro, acurados na identificação de fraudes.

1.2 Justificativa do Tema

A identificação de transações fraudulentas tem como principal objetivo a redução das perdas em determinado produto de uma instituição. As ações fraudulentas podem gerar prejuízos que superam as receitas com este produto, tornando-o inviável, daí a necessidade de equipes que previnam os ataques e reajam a estes através da identificação de transações fraudulentas e adoção de medidas de recuperação de valores.

O combate às fraudes também compõe a estrutura de gerenciamento de risco operacional das instituições financeiras. É uma exigência legal do Banco Central do Brasil que determina, dentre outras coisas, que a estrutura deve ser compatível com a natureza e a complexidade dos produtos, serviços, atividades e sistemas da instituição [3].

Além de ser uma exigência legal, o combate efetivo às fraudes transformou-se em processo cada vez mais complexo e desafiador. Novos ataques surgem a todo momento, e a instituição corre o risco de ficar para trás em sua capacidade de defesa com relação às instituições concorrentes e com relação aos atacantes, comprometendo a sua reputação e suas finanças [30].

Logo, é possível reduzir os prejuízos financeiros através da identificação das transações fraudulentas, aumentar a recuperação de valores através da adoção de medidas reativas e diminuir o risco de dano à imagem da instituição, bem como, o desgaste junto aos clientes.

1.3 Contribuição Tecnológica Esperada

A principal contribuição tecnológica deste trabalho é sistematizar a indução de classificadores para identificação de fraudes permitindo celeridade na atualização, avaliação e implantação do modelo de detecção quando do surgimento de novos ataques.

Para alcançar este objetivo principal, são automatizadas as etapas de pré-processamento dos dados - eliminação de atributos, limpeza e integração dos dados transacionais, tratamento do desbalanceamento e criação de atributos derivados - para que os algoritmos de classificação possam ser aplicados sistematicamente para geração de modelos adaptáveis às novas investidas.

Capítulo 2

Fundamentação Teórica

Em computação, problemas são resolvidos através de algoritmos que especificam, passo a passo, como alcançar uma solução. Entretanto, não é trivial escrever programas que realizam tarefas complexas como:

- prever cotações de moedas e ações;
- reconhecer pessoas pelo rosto ou pela fala;
- classificar textos ou documentos;
- identificar conjuntos de produtos que são frequentemente vendidos em conjunto ou
- detectar fraudes.

Estes problemas e diversos outros são tratados com sucesso através de técnicas de Inteligência Artitificial, em particular de Aprendizado de Máquina, fazendo uso do conhecimento previamente adquirido para a proposição de uma solução para cada uma das tarefas em questão [24].

Nesta seção é apresentado o embasamento teórico necessário para a geração de classificadores através de algoritmos de Aprendizado de Máquina para a identificação de transações bancárias fraudulentas.

2.1 Inteligência Artificial e Aprendizado de Máquina

Inteligência Artificial (IA) pode ser definida como a área da ciência da computação preocupada em como dar aos computadores a sofisticação de agir inteligentemente, isto é, diante de uma dada situação é tomada a melhor decisão possível [27]. Por sua vez, Aprendizado de Máquina (AM) pode ser definido pelo conjunto de métodos computacionais que usam o conhecimento disponível para melhorar o desempenho na realização de determinada tarefa ou para realizar previsões mais acuradas sobre certo problema [40].

Assim, AM é naturalmente associado à IA, embora outras áreas de pesquisa contribuam direta e significantemente no avanço do AM, como Probabilidade e Estatística, Teoria da Computação, Neurociência, etc.

São apresentadas a seguir definições básicas para descrição e avaliação dos algoritmos de AM [40]:

- *Objetos*: itens ou instâncias de dados usados para aprendizado ou avaliação. No problema de diagnóstico por imagens, estes objetos correspondem à coleção de imagens radiológicas que serão usadas para o aprendizado e testes.
- *Atributos*: em geral, cada objeto é descrito por um conjunto de características que o definem. No caso de transações bancárias, alguns atributos relevantes podem incluir o valor e tipo da transação, o terminal utilizado, etc.
- *Rótulos*: categorias atribuídas aos objetos. Em problemas de classificação, os objetos são atribuídos a categorias específicas, por exemplo, às categorias *positiva* ou *negativa* no problema de reconhecimento de pessoas pelo rosto ou fala.
- *Conjunto de hipóteses*: um conjunto de funções ou modelos que mapeam os atributos (vetor de características) ao conjunto de categorias.
- *Amostra de treinamento*: objetos usados por um algoritmo de aprendizado para indução de uma hipótese.
- *Amostra de validação*: objetos usados para ajustar os parâmetros de um algoritmo de aprendizado quando as instâncias de dados são rotuladas.
- *Amostra de teste*: objetos usados para avaliar o desempenho de uma hipótese gerada por um algoritmo de aprendizado. A amostra de teste e as amostras de treinamento e validação devem ser disjuntas.

Os algoritmos de Apredizado de Máquina podem ser classificados em função do cenário de aprendizado [40]:

- *Aprendizado Supervisionado*: o algoritmo recebe um conjunto de objetos rotulados como amostra de treinamento e realiza alocações para os novos objetos. Este é o cenário mais comum em problemas de classificação e regressão.
- *Aprendizado Não Supervisionado*: o algoritmo recebe exclusivamente dados de treinamento não rotulados e descobre semelhanças entre eles. Uma vez que não existem objetos rotulados, pode ser difícil avaliar quantitativamente o desempenho do algoritmo.

Uma vez que o sucesso de uma hipótese ou modelo induzido por um algoritmo de AM depende dos dados disponíveis [40], são utilizadas técnicas para a melhoria da qualidade dos mesmos. Na seção 2.2 são apresentadas algumas destas técnicas.

2.2 Preparação dos Dados

O processo de aprendizado de algoritmos de AM pode ser facilitado e o desempenho desses algoritmos melhorado se técnicas de pré-processamento forem previamente aplicadas aos dados. Isso ocorre porque essas técnicas podem eliminar ou reduzir problemas presentes nos dados, como dados faltantes [7]. Existem diversas técnicas, apresentadas a seguir.

2.2.1 Eliminação Manual de Atributos

Consiste na eliminação de atributos que não contribuem para a categorização de um objeto em uma tarefa de classificação. Uma situação na qual um atributo pode ser eliminado ocorre quando não contém informação que ajude a distinguir os objetos, isto é, o atributo possui o mesmo valor para todos os objetos. No contexto deste trabalho, o atributo *número de conta* não contribui para a estimativa do rótulo, segundo os especialistas do domínio, e pode ser eliminado, por exemplo. Outros atributos que não agregam informação útil são números sequenciais de controle, como identificadores de registro.

2.2.2 Integração de Dados

Quando os dados a serem utilizados estão distribuídos em diferentes conjuntos de dados, estes conjuntos devem ser integrados antes do início do uso da técnica de AM. Alguns aspectos podem dificultar a integração: atributos correspondentes podem ter nomes distintos em diferentes bases ou os dados a serem integrados podem ter sido atualizados em momentos diferentes. Neste trabalho, a informação básica provêm da transação de transferência eletrônica, contudo apenas os dados transacionais são insuficientes para a tarefa de identificação de fraudes. Assim, é necessário a integração com outras fontes de informação para agregar informações úteis nesta tarefa, como perfis transacionais e informações sobre o dispositivo que efetua a transação pelo *Internet/Mobile Banking*, por exemplo.

2.2.3 Amostragem de Dados

Algoritmos de AM podem ter dificuldades no trato de amostra de dados com muitos objetos, em virtude das respectivas complexidades computacionais. Para resolver este problema, usa-se um subconjunto dos dados. Existem basicamente três abordagens para amostragem [40]:

- amostragem aleatória simples;
- amostragem estratificada;
- amostragem progressiva.

A amostragem aleatória simples possui duas variações: amostragem simples sem reposição de objetos e amostragem simples com reposição.

A amostragem estratificada é usada quando as classes apresentam propriedades diferentes, por exemplo, diferentes quantidades de objetos. A abordagem mais simples mantém o número de objetos em cada classe proporcional ao número de objetos da classe no conjunto original.

Por sua vez, a amostragem progressiva começa com uma amostra pequena e aumenta progressivamente o tamanho da amostra extraída enquanto a acurácia do modelo induzido melhorar com a variação da amostra.

2.2.4 Balanceamento de Dados

Em vários conjuntos de dados reais, o número de objetos varia para as diferentes classes e isto pode constituir um problema para vários algoritmos de AM, pois estes algoritmos podem ter dificuldades para aprender o conceito relacionado à classe minoritária [25].

Quando alimentados com dados desbalanceados, esses algoritmos tendem a gerar um modelo que favorece a classificação de novos dados na classe majoritária. Para lidar com o desbalanceamento de dados, as principais técnicas são [44]:

- redefinir o tamanho do conjunto de dados;
- utilizar diferentes custos de classificação para as diferentes classes;
- induzir um modelo para cada classe.

No primeiro caso, pode-se acrescentar objetos à classe minoritária ou eliminar objetos da classe majoritária. Com o acréscimo de novos objetos, aumenta-se o esforço computacional para cálculo do modelo e há o risco de indução de um modelo inadequado para os dados; além disso, pode ocorrer o superajustamento (*overfitting*) do modelo aos dados de treinamento. Quando dados são eliminados da classe majoritária, é possível que dados

de grande importância para a indução do modelo não sejam considerados, levando ao subajustamento do modelo (*underfitting*).

O fator de redefinição do tamanho das classes é determinado, usualmente, de maneira empírica, pois a replicação aleatória de objetos da classe minoritária pode levar a modelos que aparentam ser acurados mas estão superajustados aos dados [8, 25]. Por outro lado, a redução da classe majotitária pode descartar dados potencialmente úteis que poderiam ser importantes ao processo de indução [25].

A utilização de custos de classificação diferentes para as classes majoritária e minoritária tem como dificuldade a definição destes custos. Outro problema dessa abordagem é a dificuldade de incorporar a consideração de diferentes custos em alguns algoritmos de AM.

O último caso inclui as técnicas de classificação com apenas uma classe, em que a classe minoritária ou a classe majoritária são aprendidas separadamente. Nesse caso, pode ser utilizado um algoritmo de classificação para apenas uma das classes [41].

2.2.5 Limpeza de Dados

Conjuntos de dados podem apresentar problemas como: dados inconsistentes, redundantes, incompletos ou ruidosos. Dados inconsistentes, redundantes ou com valores ausentes são fáceis de detectar. A principal dificuldade está na detecção de ruídos.

Os dados com ruídos podem levar a um superajuste do modelo, pois o algoritmo que induz o modelo pode se ater às especificidades relacionadas aos ruídos, em vez da distribuição verdadeira que gerou os dados; por outro lado, a eliminação desses dados pode fazer com que algumas regiões do espaço de atributos não sejam consideradas no processo de indução de hipóteses [7].

Existem diversas técnicas de pré-processamento voltadas à eliminação de ruídos. Em Estatística, esse problema é comumente solucionado por meio de técnicas baseadas em distribuição, em que os ruídos são identificados como observações que diferem de uma distribuição utilizada na modelagem dos dados [7]. O maior problema dessa abordagem está em assumir que a distribuição dos dados é conhecida *a priori*, o que não reflete a verdade em grande parte das aplicações práticas, como na identificação de transações fraudulentas.

Técnicas de *encestamento* são também utilizadas para reduzir o ruído em um atributo: primeiro, os valores encontrados para esse atributo são ordenados; em seguida esses valores são particionados em faixas ou cestas, cada uma com o mesmo número de valores. Os valores em uma mesma cesta são substituídos, por exemplo, pela média ou mediana dos valores presentes na cesta [7].

2.2.6 Redução de Dimensionalidade

Para que objetos com um número elevado de atributos possam ser utilizados em muitos algoritmos de AM, a quantidade de atributos precisa ser reduzida. A redução pode melhorar o desempenho do modelo induzido, reduzir seu custo computacional e tornar os resultados obtidos mais compreensíveis. As técnicas de redução de dimensionalidade podem ser divididas em duas grandes abordagens: agregação e seleção de atributos. Enquanto as técnicas de agregação substituem parte dos atributos originais por novos atributos formados pela combinação de grupos de atributos, as técnicas de seleção mantêm uma parte dos atributos originais e desconsideram os demais.

Análise de Componentes Principais

A Análise de Componentes Principais ou *Principal Component Analysis (PCA)* é um exemplo de técnica de agregação. Este método busca combinações lineares, chamadas de componentes principais, que capturam, de maneira resumida, a maior variabilidade possível dos dados. O primeiro componente principal é definido como uma combinação linear dos atributos que captura a maior variância de todas as possíveis combinações lineares. Os componentes subsequentes capturam a maior variabilidade remanescente e são não-correlacionados com os componentes anteriores [18]. Matematicamente, cada componente principal *pc* pode ser escrito como:

$$pc_j = a_{j1} \cdot atributo_{j1} + ... + a_{jn} \cdot atributo_{jn} \tag{2.1}$$

Onde a quantidade de componentes principais é menor ou igual ao número de atributos e os coeficientes a_{j1},...,a_{jn} são chamados pesos dos componentes e revelam a importância de cada atributo para cada componente principal [18].

A principal vantagem da PCA como método de redução de dimensionalidade encontra-se na construção de componentes não-correlacionados. O uso de atributos correlacionados em alguns algoritmos de AM, como regressão logística ou redes neurais, pode introduzir erro e diminuir o desempenho do algoritmo [18]. Entretanto, a PCA não leva em consideração as classes dos objetos quando do cálculo dos componentes principais e, por isso, é uma técnica não supervisionada. Uma vez que não há relacionamento entre os atributos e os rótulos dos objetos, os componentes principais não provêem um relacionamento apropriado com as categorias dos objetos em tarefas de aprendizado supervisionado [18].

Teste χ^2

Por sua vez, as técnicas de seleção avaliam a relevância do atributo antes da aplicação de algum modelo de AM. Somente os atributos correlacionados com os rótulos dos objetos

serão considerados na aplicação do modelo [18].

Para estimar as associações entre os atributos e as categorias, pode-se utilizar o teste χ^2. Este teste é não-paramétrico, isto é, não é necessário admitir hipóteses sobre a distribuição de probabilidade da qual tenham sido extraídas os dados para análise [5]:

Seja ϵ um experimento aleatório. Sejam E_1,...,E_k, k eventos associados a ϵ. O experimento é realizado n vezes. Considere que Fo_1,...,Fo_k sejam as frequências observadas para cada um dos k eventos considerados e que Fe_1,...,Fe_k sejam as frequências esperadas dos k eventos considerados.

Deseja-se realizar um teste estatístico para verificar se há associação, ou independência, entre duas variáveis. Isto é, se as discrepâncias (Fo_i - Fe_i) são devidas ao acaso, ou se de fato existe diferença significativa entre as frequências.

Considere que a representação das frequências observadas é dada por uma tabela de contingência com L linhas e C colunas. Para efetuar o teste de independência entre as duas variáveis:

- Enuncie as hipóteses nula e alternativa:

  H_0 afirma que as variáveis são independentes.

  H_1 afirma que as variáveis estão associadas.

- Fixe um nível de confiança α e escolha a variável χ^2 com grau de liberdade φ:

$$\varphi = (L-1)(C-1). \tag{2.2}$$

- Com auxílio da tabela de distribuição χ^2, determine χ^2_{tab}.

- Cálculo do valor da variável χ^2_{cal}:

$$\chi^2_{cal} = \sum_{i=1}^{L}\sum_{j=1}^{C}\frac{(Fo_{ij} - Fe_{ij})^2}{Fe_{ij}}. \tag{2.3}$$

  Onde cada Fe_{ij} é determinado pelos elementos a_{ij} da tabela de contingência:

$$Fe_{ij} = \frac{\sum_{i=1}^{L} a_{ij} \cdot \sum_{j=1}^{C} a_{ij}}{\sum_{i=1}^{L}\sum_{j=1}^{C} a_{ij}} \tag{2.4}$$

- Conclusão:

  Se $\chi^2_{cal} \leq \chi^2_{tab}$, não se pode rejeitar H_0.

  Se $\chi^2_{cal} > \chi^2_{tab}$, rejeita-se H_0, concluindo, com risco α, que as variáveis estão associadas.

O grau de associação entre duas variáveis analisadas pelo teste χ^2 pode ser representado pelo *coeficiente de contingência de Pearson*, *CC*, apresentado como:

$$CC = \sqrt{\frac{\chi^2_{cal}}{\chi^2_{cal} + \sum_{i=1}^{L} \sum_{j=1}^{C} a_{ij}}}. \tag{2.5}$$

O coeficiente de contingência é nulo quando não há associação entre as variáveis e quanto maior o valor de *CC*, maior será a associação entre as variáveis [5].

2.2.7 Transformação de Dados

Alguns algoritmos de AM estão limitados à manipulação de valores de determinados tipos, por exemplo, apenas valores numéricos (técnicas como redes neurais e máquinas de vetores de suporte) ou apenas valores simbólicos (árvores de decisão ID3, por exemplo). As técnicas de transformação de dados podem ser utilizadas para converter valores simbólicos em valores numéricos, ou vice-versa. Outro tipo de transformação, notadamente relacionada a atributos com valores numéricos, envolve a mudança de escala ou de intervalo de valores.

Conversão Simbólico-Numérico

Quando o atributo é do tipo nominal e assume apenas dois valores, um dígito binário é suficiente para representá-los, trivialmente. Para um atributo simbólico com mais de dois valores, a técnica utilizada na conversão depende de o atributo ser nominal ou ordinal.

Se não houver relação de ordem entre os atributos simbólicos, esta propriedade deve ser mantida para os valores numéricos gerados. Uma maneira de se conseguir isto é representar cada valor nominal por uma sequência de c bits, em que c é o número de possíveis categorias. Nesta codificação, cada sequência possui apenas um bit com o valor 1 e os demais com valor zero. A diferença entre as sequências é definida pela posição que o valor 1 ocupa nelas. Para definir a diferença entre dois valores, pode ser utilizada a distância de *Hamming*: a distância entre duas sequências binárias com mesmo número de elementos é igual ao número de posições em que as sequências apresentam valores diferentes [7].

Se existe uma relação de ordem entre os valores nominais, a codificação deve manter essa propriedade: basta ordenar os valores categóricos ordinais e codificar cada valor de acordo com sua posição na ordem, utilizando-se números inteiros ou reais.

Conversão Numérico-Simbólico

Se o atributo original for formado por sequências binárias sem uma relação de ordem entre si, cada sequência pode ser substituída por um nome ou categoria. Nos demais casos, fica a cargo do método de discretização a definição de como mapear os valores dos atributos quantitativos para valores qualitativos, o tamanho dos intervalos ou a quantidade de valores nos intervalos. Algumas estratégias são [7]:

- Larguras iguais: o intervalo original de valores é dividido em subintervalos de mesma largura.
- Frequências iguais: atribui o mesmo número de objetos a cada subintervalo.
- Uso de um algoritmo de agrupamento: visa maximizar a pureza dos intervalos.
- Inspeção visual.

Transformação de Atributos Numéricos

Algumas vezes, o valor numérico de um atributo precisa ser transformado em outro valor numérico; por exemplo, quando há uma grande variação de valores, ou ainda, quando vários atributos estão em escalas diferentes.

Uma transformação que é muito utilizada é a normalização de dados: a cada valor do atributo a ser normalizado é adicionada (ou subtraída) uma medida de localização e o valor resultante é em seguida multiplicado (ou dividido) por uma medida de escala [18].

Se as medidas de localização e de escala forem a média (μ) e a variância (σ), respectivamente, os valores de um atributo são convertidos para um novo conjunto de valores com média 0 e variância 1, obtidos através da equação abaixo quando aplicada nos valores originais dos atributos [18]:

$$v_{novo} = \frac{(v_{original} - \mu)}{\sigma} \tag{2.6}$$

2.3 Modelos Preditivos

Um algoritmo de AM preditivo produz, dado um conjunto de objetos rotulados, um modelo capaz de produzir previsões a respeito dos objetos [40]. Os rótulos dos objetos assumem valores em um domínio conhecido. Se o domínio for um conjunto infinito e ordenado de valores, tem-se um problema de regressão e o modelo gerado é um regressor; por outro lado, se esse domínio for um conjunto de valores nominais, tem-se um problema de classificação e o modelo gerado é um classificador [7].

Os classificadores podem ser gerados por diversos algoritmos de AM preditivos, por exemplo:

- k-NN: é um método baseado em distâncias que classifica um novo objeto com base nos exemplos do conjunto que são próximos a ele [7];
- *naive* Bayes: baseia-se no Teorema de Bayes para cálculo das probabilidades das hipóteses e estima a classificação de novos objetos [24];
- máquinas de vetores de suporte: esta técnica envolve a solução de um problema de otimização quadrática, formulado com o objetivo de maximizar a margem de separação entre os objetos de diferentes classes [40];
- redes neurais artificiais (RNA): é outro método baseado em otimização de funções que estimam o erro entre as respostas da rede e os rótulos dos objetos, em uma das suas possíveis implementações [24];
- árvores de decisão: recebem como entrada um objeto, descrito por seu conjunto de atributos, e produzem, através de testes baseados no valor dos atributos, uma decisão que corresponde aos possíveis rótulos [28];

Em particular, uma rede neural artificial é inspirada nos sistemas biológicos de aprendizado e é constituída por um conjunto de unidades simples, versões abstratas dos neurônios, densamente interconectados [24]. Usualmente, estas unidades estão arranjadas em camadas: uma camada de entrada, com as unidades representando os atributos de entrada; uma ou mais camadas ocultas e uma camada de saída representando o atributo alvo. Os dados de entrada são apresentados à primeira camada, estes valores são ponderados e combinados através de uma função matemática e os valores calculados são propagados para os neurônios da camada seguinte até que o resultado seja entregue pela camada de saída [7]. O aprendizado da rede ocorre através da avaliação dos objetos de treinamento: é produzido um resultado para cada registro, caso este resultado seja incorreto quando comparado ao respectivo rótulo, são realizados ajustes nos pesos das interconexões dos neurônios. Este processo é repetido diversas vezes até a rede neural atinja algum critério de parada [7, 24].

Por sua vez, as árvores de decisão são um dos classificadores mais práticos e amplamente usados [14, 24]. Uma árvore de decisão cobre todo o espaço de instâncias ou objetos, isto implica que o modelo gerado por uma árvore de decisão pode classificar qualquer objeto de entrada [7, 24]. Uma vez que todas as decisões para a classificação dos objetos são baseadas nos valores dos atributos dos objetos, a análise e interpretação do classificador podem trazer informações úteis sobre o relacionamento dos objetos e sobre

o cenário de aprendizado [7, 24, 32]. Por aderência ao problema deste trabalho, foram escolhidas árvores de decisão para a identificação de transações fraudulentas.

2.4 Trabalhos Correlatos

O problema de detecção de fraudes tem sido abordado de diversas maneiras na literatura [1, 6, 10, 16], onde o tipo de fraude que se deseja identificar influencia a seleção do método de detecção a ser utilizado. Entretanto, pode-se notar a utilização de diversos algoritmos de aprendizado de máquina para esta finalidade. Estas ferramentas são apropriadas à detecção de fraudes por três razões [19]: são flexíveis e facilmente adaptáveis a novas circunstâncias; não requerem que os projetistas especifiquem todas as condições sob as quais irão trabalhar ou adquirir conhecimento e encontram novas associações entre os padrões de fraude.

A maioria dos trabalhos publicados sobre detecção de fraudes está relacionada ao domínio de cartão de crédito [1, 44], concessão de crédito [23], intrusão de computadores [17, 22] e fraudes em serviços de telecomunicações [36, 43]. Estes artigos examinam o uso de redes neurais artificiais, máquinas de vetores de suporte, algoritmos genéticos, lógica nebulosa, árvores de decisão, dentre outras técnicas para a identificação de fraudes. Entretanto há poucos trabalhos publicados relacionados à identificação de fraudes bancárias, devido ao fato das instituições financeiras, tradicionalmente, não fornecerem detalhes sobre seus sistemas de detecção de fraudes [16].

Kovach [21] propõe uma arquitetura de um sistema de detecção de fraudes bancárias, realizadas através da *Internet*, em tempo real baseada na teoria matemática de evidências de *Dempster-Shafer*. Contudo, a evidência de fraude é baseada no número de acessos efetuados em contas diferentes por um mesmo dispositivo. Uma vez que desconsidera detalhes transacionais dos serviços bancários, utilizados neste trabalho, não é possível comparação entre ambos.

Vadivu et al [29] utilizam árvores de decisão para a identificação de transações de débito fraudulentas. Os autores concluem que as árvores de decisão são úteis na identificação destas fraudes e auxiliam o processo de tomada de decisão ao determinar as características que determinam a fraude, além de manipularem grandes quantidades de dados com complexidade linear; assim, este artigo apresenta maior aderência à tarefa de sistematizar a indução de árvores de decisão para identificação de transferências bancárias fraudulentas, objeto de estudo do presente trabalho, apesar de lidar com outra categoria de transação bancária e desconsiderar importantes características, como perfis transacionais dos clientes, na tarefa de identificação.

Nas seções seguintes serão apresentados maiores detalhes sobre a indução de árvores de decisão.

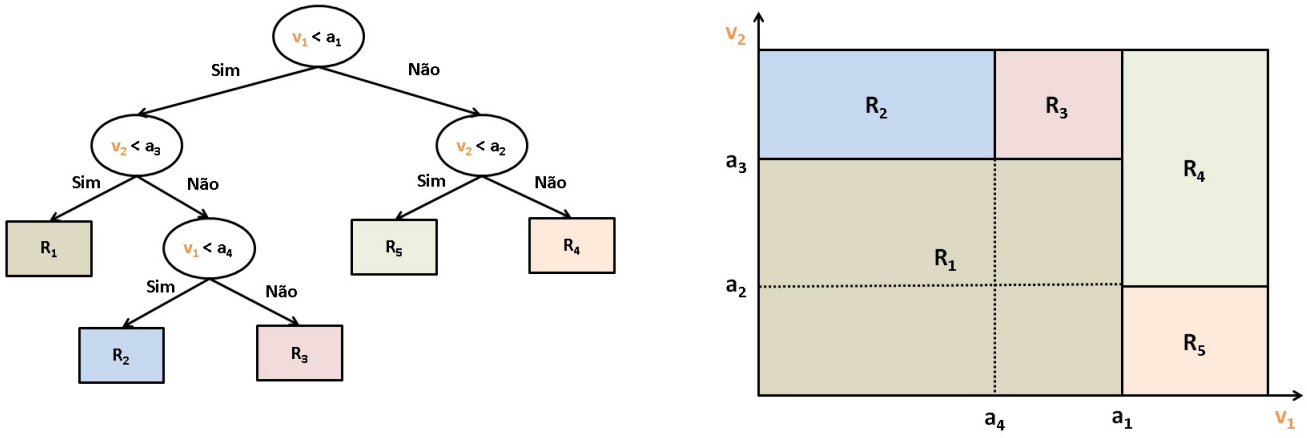
2.5 Árvores de Decisão

Uma árvore de decisão usa a estratégia dividir para conquistar na solução de um problema de decisão: um problema complexo é dividido em problemas mais simples, aos quais recursivamente é aplicada a mesma estratégia; as soluções dos subproblemas podem ser combinadas, na forma de uma árvore, para gerar uma solução do problema complexo [32]. Essa é a idéia básica por trás de algoritmos baseados em árvores de decisão, tais como: *ID3* [31] e *C4.5* [32].

Formalmente, uma árvore de decisão é um grafo acíclico direcionado em que cada nó ou é um *nó de divisão*, com dois ou mais sucessores, ou um *nó folha* [7]:

- um *nó folha* contém os valores dos rótulos;
- um *nó de divisão* contém um teste condicional baseado nos valores dos atributos.

Figura 2.1: Uma árvore de decisão e as regiões de decisão no espaço de objetos



A Figura 2.1 representa uma árvore de decisão e a respectiva divisão no espaço de objetos definida pelos atributos v_1 e v_2. Cada folha da árvore corresponde a uma região neste espaço e a reunião de todas as regiões abrange todo espaço de instâncias. Os testes contidos nos nós de divisão correspondem, no espaço de entrada, a um hiperplano que é ortogonal aos eixos do atributo testado e paralelo a todos os outros eixos.

2.5.1 Indução de Árvores de Decisão

O processo de construção de uma árvore a partir de uma amostra de treinamento é conhecido como indução da árvore. A construção de árvores de decisão binárias[1] mínimas, com relação ao número de nós, para a classificação de um objeto é um problema NP-completo [35] e a indução de uma árvore de decisão mínima, também relativa ao número de nós, é um problema NP-difícil [9]; consequentemente, é inviável procurar algoritmos que construam árvores de decisão mínimas e, por isso, métodos heuristícos são utilizados [9, 35].

A maioria das estratégias de indução de árvores procedem de maneira gulosa, selecionando um teste condicional que melhor discrimina as classes, a partir da raiz da árvore até as folhas [26]. Começando com uma árvore vazia e a amostra de treinamento, o algoritmo abaixo descrito é aplicado até que a condição de parada seja alcançada [26]:

1 - Se todos os objetos de treinamento no nó corrente t pertencem à categoria c, crie um nó folha com a classe c.

2 - Caso contrário, avalie cada um dos possíveis testes condicionais s pertencentes ao conjunto S dos possíveis testes, usando uma função heurística.

3 - Escolha o melhor teste condicional s como teste do nó corrente.

4 - Crie um nó sucessor para cada resultado distinto do teste s e particione os dados de treinamento entre os nós sucessores usando o teste s.

5 - Um nó t é tido como *puro* se todos os objetos de treinamento em t pertencem à mesma classe. Repita os passos anteriores em todos os nós *impuros*.

Um objeto τ é classificado pela árvore através de seu caminho desde o nó raiz até um nó folha. O teste em cada nó de divisão ao longo do caminho é aplicado aos atributos de τ para determinar o próximo caminho que τ deverá seguir. O rótulo no nó folha no qual o objeto τ se encontra ao término do percurso define a sua classificação.

As condições de parada do crescimento da árvore comumente usadas são [7]:

- Todos os objetos de treinamento pertencem a uma mesma categoria.
- A altura máxima da árvore foi alcançada.
- Caso ocorresse a divisão do nó, o número de objetos em um ou mais nós sucessores é menor que um certo limite inferior.

Testes Condicionais para Classificação

A escolha por um teste condicional é guiada por uma função heurística que indica quão bem um dado atributo discrimina as classes [24]. Para cada teste possível, o sis-

[1]Uma árvore de decisão binária é um grafo acíclico direcionado em que cada nó ou é um *nó de divisão*, com no máximo dois sucessores, ou um *nó folha*.

tema hipoteticamente considera os subconjuntos dos dados obtidos e escolhe o teste que maximiza a heurística sobre os conjuntos. Por exemplo, o algoritmo *ID3* utiliza o ganho de informação para a determinação do teste condicional a ser alocado no nó de divisão, enquanto o algoritmo *C4.5* utiliza, além do ganho de informação, a medida razão de ganho, pois essa medida tende produzir árvores de acurácia superior, bem como, menos complexas quando comparadas às geradas unicamente com o ganho de informação [32].

Ganho de Informação

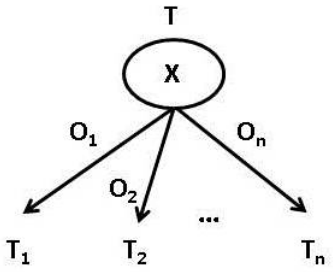
Sejam S um conjunto qualquer de objetos, $|S|$ o número de objetos em S e *freq(C_j,S)* o número de objetos em S que possuem o rótulo C_j. A quantidade média de informação necessária para identificar o rótulo de um objeto em S, medida em bits, é dada por [32]:

$$info(S) = -\sum_{j=1}^{k} \frac{freq(C_j,S)}{|S|} \cdot \log_2 \frac{freq(C_i,S)}{|S|} \tag{2.7}$$

Esta quantidade também é conhecida como entropia do conjunto S.

Assim, dada uma amostra de treinamento T, *info(T)* mede a informação necessária para identificar o rótulo de um objeto nesta amostra. Suponha que exista um teste X com resultados $O_1, ..., O_n$ que particionam a amostra de treinamento T nos subconjuntos $T_1, ..., T_n$.

Figura 2.2: Aplicação do teste X ao conjunto de treinamento T



Após a aplicação do teste X, a informação necessária para a classificação de um objeto nos subconjuntos resultantes $T_1, ..., T_n$ é dada por:

$$info_X(T) = \sum_{i=1}^{n} \frac{|T_i|}{|T|} \cdot info(T_i) \tag{2.8}$$

Logo, o ganho de informação ao particionarmos T de acordo com o teste X é dado por:

$$gain(X) = info(T) - info_X(T) \tag{2.9}$$

Essa heurística seleciona o teste que resulta no máximo ganho de informação [31], que pode ser utilizada para como teste em um nó divisão durante o processo de construção da árvore.

Razão de Ganho

O ganho de informação possui um forte viés em favor de testes que resultam em muitas partições dos dados; por exemplo, gain(X) é maximizado por um teste no qual cada T_i possui um único objeto [32]. Este viés pode ser corrigido por um tipo de normalização no qual o ganho aparente atribuível a um dado teste é corrigido.

Por analogia com a definição de *info(S)*, a informação potencial gerada pela divisão de T em n subconjuntos é definida como:

$$splitinfo(X) = -\sum_{i=1}^{n} \frac{|T_i|}{|T|} \cdot \log_2 \frac{|T_i|}{|T|} \tag{2.10}$$

Por outro lado, o ganho de informação mede a informação relevante à classificação que é obtida da aplicação do mesmo teste X. Então, para expressar a proporção de informação gerada pelo teste X que é útil à tarefa de classificação, é usada a razão de ganho:

$$gainratio(X) = \frac{gain(X)}{splitinfo(X)} \tag{2.11}$$

A razão de ganho é calculada em duas etapas: inicialmente o ganho de informação é calculado para todos os atributos para a obtenção do ganho médio de todos os testes examinados. Na sequência, são selecionados os atributos que tiveram desempenho superior ao ganho médio; aquele atributo que maximizar a razão de ganho será selecionado como teste [32].

2.5.2 Valores Desconhecidos

O algoritmo básico descrito para construção da árvore de decisão supõe que os testes condicionais nos nós de divisão podem ser sempre calculados. Uma vez que o teste é baseado em um único atributo, o resultado de um teste não poderá ser calculado se existirem objetos com o valor deste atributo desconhecido.

Frequentemente em situações reais alguns valores de atributos são desconhecidos ou indeterminados; este problema é relevante em árvores de decisão, pois se não tratado, pode ser impossível determinar o percurso que dado objeto com este problema seguirá.

Uma possível estratégia utilizada para tratar o problema consiste, basicamente, na substituição do valor não conhecido: pelo valor mais frequente no conjunto de treina-

mento, pelo valor determinado pela exploração de inter-relacionamentos entre os valores dos demais atributos ou pela exploração da distribuição de probabilidade dos valores dos atributos [32].

Adaptação dos Testes Condicionais

Para que as árvores construídas com o algoritmo *C4.5*, cujo teste condicional é baseado no ganho de informação e razão de ganho, possam tratar este problema são necessárias modificações nos critérios de escolha do teste. Seja T um conjunto de treinamento e X um teste baseado em algum atributo A para o qual exista uma fração F de objetos em T na qual os valores de A são conhecidos. Considerando apenas os objetos em F, a definição de ganho de informação pode ser ajustada para [32]:

$$gain(X) = F \cdot (info(T) - info_X(T)) \tag{2.12}$$

Similarmente, a definição de *splitinfo(X)* pode ser alterada para levar em consideração os objetos com valores desconhecidos do atributo A como um grupo adicional [32]. Se um teste X possui n resultados, *splitinfo(X)* é calculado como se o teste dividisse os objetos em *n+1* subconjuntos.

Particionamento do Conjunto de Treinamento

Se um teste X com resultados $O_1, ..., O_n$ possui resultado indeterminado para algum objeto de treinamento, o conceito de particionamento do conjunto de treinamento T entre os subconjuntos $T_1, ..., T_n$ deve ser generalizado [32].

É associado a cada objeto, em cada uma das partições T_i, um peso w representando a probabilidade daquele objeto pertencer a cada subconjunto:

$$w = \begin{cases} 1, & \text{se o resultado do teste é conhecido} \\ \frac{|T_i|}{|T - T_0|}, & \text{caso contrário} \end{cases} \tag{2.13}$$

onde T_0 é o subconjunto de objetos em T cujos valores do atributo utilizado pelo teste X é desconhecido [32].

Agora, os objetos com valores desconhecidos serão adicionados a cada uma das partições T_i com peso

$$w_i = w \cdot \frac{|T_i|}{|T - T_0|}. \tag{2.14}$$

Classificação de Novos Objetos

Uma vez que nesta estratégia existem múltiplos caminhos para um objeto com valores faltantes desde a raiz da árvore até as folhas, o rótulo é determinado por uma distribuição de classes ao invés de uma única classe. Desta maneira, a classe com maior probabilidade é atribuída como o rótulo do objeto [32].

2.5.3 Estratégias de Poda

A poda de uma árvore consiste na substituição de nós por folhas e visam à melhoria da capacidade de generalização, pois nós mais profundos refletem mais a amostra de treinamento [7]. Outro benefício da poda é redução das árvores, uma vez que a árvore induzida tende a ser grande e de difícil compreensão.

Os métodos de poda podem ser divididos em dois grupos principais [32]: métodos que param a construção da árvore quando algum critério é satisfeito, conhecidos como *pré-poda*, e métodos que constroem uma árvore completa e a podam posteriormente, conhecidos como *pós-poda*.

A pré-poda conta com regras de parada que previnem a construção de ramos que não parecem melhorar a precisão preditiva da árvore. Três regras de parada comumente usadas são [7]:

- Todos os objetos alcançando um nó pertencem à mesma classe.
- Todos os objetos alcançando um nó possuem o mesmo vetor de características.
- Nenhum teste possível resulta em melhoria na tarefa de classicação, por exemplo.

Na pós-poda, uma árvore completa, superasjustada aos dados de treinamento, é gerada e podada posteriormente.

Poda de Erro Reduzido

Esta estratégia de pós-poda usa a informação da amostra de treinamento para construir e simplificar a árvore: é estimado um erro aparente, para cada subárvore, baseado na proporção dos objetos de treinamento classificados incorretamente nas folhas; uma subárvore é podada e substituída por uma folha quando a taxa de erro para a subárvore não é significantemente menor que o erro da folha [31].

Poda Pessimista

Considere um nó contendo N objetos de um conjunto de treinamento T e seja E o número de erros nesta amostra, suponha que os erros no conjunto de treinamento

T seguem uma distribuição binomial com probabilidade Q em N experimentos; não é possível calcular Q, porém é possível obter um intervalo de confiança [L_c,U_c] relativo à probabilidade de classificação incorreta da folha que, para um nível de confiança C, contém Q [7].

Tendo encontrado o limite superior do intervalo de confiança, U_c, as estimativas de erro para as folhas e subárvores são calculadas assumindo que serão usadas para classificar um conjunto de objetos não conhecidos do mesmo tamanho que o conjunto de treinamento. Assim, a taxa de erro para uma folha será $N \cdot U_c$. Dessa forma, comparando a taxa de erro de um dado nó, como se ele fosse trocado por uma folha, com a subárvore da raiz até esse nó, pode-se decidir pela poda ou manutenção deste nó [7].

2.6 Medidas de Desempenho

Para obter uma ampla perspectiva do desempenho do classificador serão utilizadas as medidas de desempenho: precisão e sensibilidade [7]. Estas medidas são determinadas através da matriz de confusão abaixo, onde *"positivo"* corresponde aos casos de *fraude* e *"negativo"* aos casos de *não-fraude*:

Tabela 2.1: Matriz de confusão

	Previsão Positiva	**Previsão Negativa**
Caso Positivo	Verdadeiro Positivo (VP)	Falso Negativo (FN)
Caso Negativo	Falso Positivo (FP)	Verdadeiro Negativo (VN)

Onde *VP*, *FN*, *FP* e *VN* são assim definidas [7]:

- *VP* corresponde ao número de objetos da classe positiva classificados corretamente;
- *FN* corresponde ao número de objetos pertencentes à classe positiva que foram incorretamente atribuídos à classe negativa;
- *FP* corresponde ao número de objetos cuja classe verdadeira é negativa mas que foram classificados incorretamente como pertencentes à classe positiva;
- *VN* corresponde ao número de objetos da classe negativa classificados corretamente.

Assim, as medidas de desempenho são [7]:

- sensibilidade:

$$S = \frac{VP}{(VP + FN)} \tag{2.15}$$

  mede a acurácia nos casos fraudulentos;

- precisão:

$$P = \frac{VP}{(VP + FP)} \tag{2.16}$$

  mede a acurácia nos casos apontados como fraudulentos.

Precisão e sensibilidade descrevem com exatidão o desempenho de um classificador; entretanto, possuem uma relação de conflito entre si: à medida que a precisão aumenta, há a tendência de diminuição da sensibilidade e vice-versa [45].

Além dessas ponderações, para compararmos dois classificadores distintos é útil a utilização de uma medida que combine os valores daquelas duas métricas: *medida-F*. Esta métrica corresponde à média harmônica da precisão, P, e sensibilidade, S, e é definida como [38]:

$$F_\beta = \frac{(1 + \beta^2)PS}{(\beta^2 P) + S} \tag{2.17}$$

onde β é um parâmetro que confere maior peso à sensibilidade quando $\beta>1$ e se $0<\beta<1$ confere maior peso à precisão.

Um caso particular ocorre quando se confere o mesmo peso, $\beta=1$, à precisão e à sensibilidade. Neste caso, a *medida-F* assumirá o seguinte valor:

$$F_1 = \frac{2PS}{P + S} \tag{2.18}$$

Caso a sensibilidade, S, possua o dobro do peso da precisão, P, quando da comparação de dois classificadores distintos:

$$F_2 = \frac{5PS}{4P + S} \tag{2.19}$$

obtida através da Equação 2.17, com $\beta=2$.

2.7 PMML

Para a implantação do modelo de identificação de fraudes no processo de negócio é utilizado o formato PMML [42]. O PMML, *Predictive Model Markup Language*, desenvolvido pelo *Data Mining Group*[2], é um documento XML, *Extensible Markup Language*, utilizado para representar modelos de mineração de dados e permite a representação de diversos modelos como árvores de decisão, redes neurais ou classificador *naive* Bayes. Além disso, permite representar os atributos dos objetos e as respectivas transfomações de dados necessárias à utilização de um dado modelo. Desta forma, os modelos podem ser facilmente compartilhados entre diferentes aplicações, evitando-se questões proprietárias

[2] `http://www.dmg.org/`

e incompatibilidades entre plataformas [2]. Na Figura 2.3 é exibido um típico documento PMML.

Figura 2.3: Documento PMML

```
<?xml version="1.0"?>
<PMML version="4.2"
  xmlns="http://www.dmg.org/PMML-4_2"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

  <Header copyright="Example.com"/>
  <DataDictionary> ... </DataDictionary>

  ... a model ...

</PMML>
```

Fonte: `http://www.dmg.org/`

A estrutura geral de um documento PMML é composta por um cabeçalho, dicionário de dados, transformações de dados e modelo. No cabeçalho são exibidas informações gerais sobre o documento, como aplicação usada para gerar o modelo e data de geração do modelo. No dicionário de dados são definidos todos os possíveis atributos utilizados pelo modelo e seus respectivos domínios. Em transformações de dados são definidas as funções a serem aplicadas sobre os dados para possibilitar a aplicação do modelo de mineração, como normalização, discretização, etc. Por fim, o elemento modelo contém a definição do modelo de mineração: esquema de mineração, variáveis alvo e especificidades do modelo [2].

Capítulo 3

Material e Método

3.1 Material

O desafio é identificar as transações fraudulentas dentre milhões de transações legítimas, minimizando a quantidade de alertas gerados para não desgastar a relação com o cliente da instituição e maximizando a quantidade de fraudes detectadas para mitigar o prejuízo financeiro. Cumpre destacar que ações decorrentes da identificação da fraude, como estorno de transações , repatriação de valores ou reporte aos órgãos de segurança pública não pertencem ao escopo deste trabalho. Assim, para a construção dos classificadores foi utilizada uma base de dados transacional composta por transferências eletrônicas realizadas entre dezembro/2013 e maio/2014, descaracterizada por questões de sigilo da informação.

Cada objeto nesta base de dados possuem dois atributos com função de rótulo: um atributo para identificar as transações que foram apontadas como fraudulentas pelas regras *ad-hoc* elaboradas pelos especialistas da instituição fornecedora dos dados e outro para indicar quais estão rotuladas como fraudulentas.

O rótulo *não-fraude* é atribuido, por padrão, a todas as transações, com exceção das transações que são contestadas pelos clientes da instituição. Uma vez contestada, a transação é analisada por uma equipe de especialistas que buscam indícios de fraude que comprometeriam a legitimidade da transação. Caso existam os indícios, é atribuído o rótulo *fraude* à transação em questão. Portanto, considera-se que a base de dados rotulada é correta.

3.2 Método

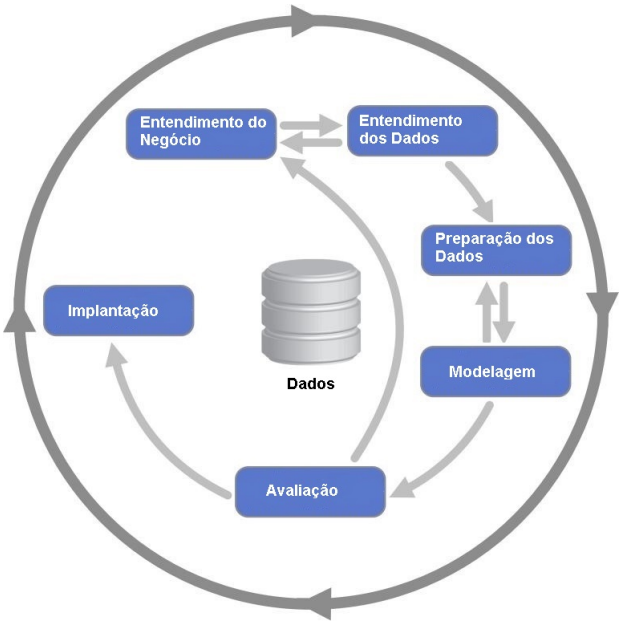
Neste trabalho será utilizada a metodologia de mineração de dados CRISP-DM (*Cross Industry Standard Process for Data Mining*) [4] para condução do projeto de construção

do classificador para identificação de transferências bancárias fraudulentas, pois é uma metodologia madura para guiar projetos de mineração de dados.

O ciclo de vida do projeto de mineração, nesta metodologia, é composto de seis fases: entendimento do negócio, entendimento dos dados, preparação dos dados, modelagem, avaliação e implantação.

O relacionamento entre estas fases pode ser visualizado na figura abaixo:

Figura 3.1: Fases do modelo de referência CRISP-DM



A metodologia CRISP-DM é flexível e permite a criação de um modelo de mineração que se ajusta às necessidades específicas dos projetos. Observa-se que a sequência de execução das fases não é rígida e depende dos resultados alcançados em cada fase.

Nas seções seguintes é realizada a descrição de cada uma das fases do modelo CRISP-DM [4].

3.2.1 Entendimento do negócio

Esta fase inicial está focada no entendimento dos objetivos do projeto a partir de uma perspectiva negocial, visando o entendimento do negócio da organização, os recursos disponíveis, os problemas envolvidos e as definição das metas do projeto. Envolve as seguintes tarefas:

- Determinação dos objetivos negociais com a compreensão das necessidades da organização que devem ser satisfeitas com o projeto;

- Avaliação do cenário atual com o levantamento dos recursos disponíveis, restrições e outros fatores que podem influenciar o atingimento dos objetivos negociais;
- Determinação das metas de mineração com a tradução das metas negociais para uma linguagem técnica, clara e não ambígua;
- Produção do plano de projeto com a especificação dos passos a serem seguidos no restante do projeto, incluindo uma seleção inicial de ferramentas e técnicas.

3.2.2 Entendimento dos dados

Visa a análise detalhada dos dados disponíveis, é uma etapa crítica, pois pode evitar problemas inesperados durante a próxima fase, a preparação dos dados. É composta pelas seguintes atividades:

- Coleta dos dados listados como recursos disponíveis;
- Descrição dos dados adquiridos e elaboração de relatórios;
- Exploração dos dados com a identificação de atributos relevantes para uma tarefa de classificação, por exemplo, bem como, resultados de agregações;
- Verificação da qualidade dos dados com a busca de inconsistências como erros de codificação, valores faltantes e as condições nas quais ocorrem.

3.2.3 Preparação dos dados

Esta fase compreende todas as atividades que visam à construção do conjunto final de dados, a partir de diversas fontes, que será utilizado para a construção dos modelos pelos algoritmos de Aprendizado de Máquina. É considerada uma das etapas mais importantes do CRISP-DM e que mais consome tempo. Usualmente, é composta por atividades de integração e seleção de dados, agregação de objetos, criação de novos atributos, particionamento dos dados em conjuntos de treinamento e testes, bem como balanceamento dos dados que estão alocadas nas seguintes tarefas:

- Seleção dos dados relevantes a serem utilizados no projeto, envolvendo a seleção de atributos e seleção de registros;
- Limpeza dos dados com a seleção de subconjuntos de dados sem inconsistências;
- Construção dos dados com a produção de atributos derivados;
- Integração dos dados com múltiplas tabelas para a criação de novos registros ou valores;

- Formatação dos dados para a utilização de determinados algoritmos de AM.

3.2.4 Modelagem

Nesta etapa são selecionados, em função da aderência ao problema a ser tratado, alguns modelos que são executados com os seus parâmetros padrão e, então, são efetuados ajustes finos dos parâmetros para atingimento dos objetivos do projeto ou retorna-se às etapas anteriores para adequação do projeto a determinado algoritmo de AM. É composta pelas seguintes atividades:

- Seleção da técnica de modelagem a ser utilizada tenha maior aderência aos objetivos negociais;
- Criação de um projeto de teste para mensurar a qualidade e validade de um modelo;
- Construção do modelo a partir do conjunto de dados preparado;
- Avaliação do modelo conforme os critérios técnicos de sucesso do projeto.

3.2.5 Avaliação

Nesta etapa os modelos obtidos na etapa de modelagem que atingiram desempenho satisfatório são reavaliados para verificação de atingimento dos objetivos negociais. Envolve as seguintes tarefas:

- Avaliação dos resultados segundo critérios negociais de sucesso;
- Revisão do processo para determinar se alguma tarefa deve ser revista;
- Definição dos próximos passos em função da avaliação dos resultados e da revisão do processo. O projeto poderá progredir para a etapa de Implantação ou retornar para alguma das etapas anteriores.

3.2.6 Implantação

Esta etapa envolve a utilização das descobertas para melhoria dos processos dentro da organização, pode significar a integração do modelo obtido aos sistemas da corporação ou simplesmente a geração de um relatório expondo o conhecimento obtido para utilização durante tomadas de decisão.

- Plano de implantação leva a uma estratégia para integração do modelo dentro da organização;

- Plano de monitoração visa evitar o uso incorreto dos resultados de mineração pelo uso de modelos desatualizados;
- Produção de relatório final pode ser um relatório ou uma apresentação gerencial dos resultados alcançados;
- Revisão do projeto sumariza as decisões mais relevantes tomadas durante o projeto, o que funcionou, o que deu errado e o que precisa ser melhorado.

Capítulo 4

Experimentos

Aqui são apresentadas todas as tarefas que precederam a geração sistemática dos classificadores, seguindo o modelo de referência CRISP-DM.

4.1 Entendimento do Negócio

O combate às fraudes é um processo dinâmico e transversal, pois envolve diversas unidades empresariais; além disso, requer que a detecção das transações fraudulentas seja tempestiva, pois quanto mais tempo um atacante tiver para realizar transações financeiras, maior será o prejuízo da instituição; requer ações tempestivas que viabilizem a recuperação de valores; bem como, a adoção de tecnologias que impeçam ou dificultem a efetivação da fraude, uma vez que as técnicas de ataque são atualizadas constantemente, as regras e/ou modelos de detecção, soluções de segurança também necessitam de atualização.

Mais especificamente, a tarefa de identificar transações fraudulentas realizadas pelos canais virtuais lida com grande volume de transações que são alocadas nas classes *fraude* e *não-fraude*. Estas classes são altamente desbalanceadas, pois diariamente são realizadas milhões de transações legítimas e apenas algumas dezenas de transações fraudulentas; isto torna a tarefa de identificação mais laboriosa, pois a fraude é um evento raro.

Além disso, o custo financeiro de uma transação fraudulenta não alertada, *falso-negativo*, é superior ao custo de uma transação legítima apontada como fraudulenta, *falso-positivo*; pois quando uma transação legítima é apontada como fraudulenta, o tratamento deste alerta envolve o custo da mão-de-obra necessária para analisá-lo. Por sua vez, se a fraude não for detectada, além do prejuízo financeiro, a conta vitimada pode ser utilizada para recebimento de outros créditos fraudulentos.

E, finalmente, o comportamento dos fraudadores é caracterizado pela dinamicidade, pela realização de transações similares às do cliente vitimado e pela realização de testes em busca de vulnerabilidades ainda desconhecidas em produtos ou serviços.

Neste contexto, a identificação de transações fraudulentas baseada em métodos *ad-hoc*, nos quais as regras de detecção são exclusivamente escritas por especialistas do domínio, não é aceitável, pois, normalmente são muito laboriosas e requerem muito tempo para elaboração e implantação ficando desatualizadas rapidamente.

A adoção de uma metodologia completa e adequada que garanta um modelo de detecção em sintonia com os ataques vigentes, eficiente e de boa acurácia, é recomendada dada a dinamicidade dos ataques e riscos envolvidos.

Assim, a proposta deste trabalho é prover um método sistemático que induza classificadores automaticamente para detecção de fraudes bancárias em transferências de valores realizadas por canais virtuais.

Espera-se identificar pelo menos 80% das fraudes, com uma proporção de uma transação fradulenta a cada três casos de falso-positivo; isto é, alcançar *sensibilidade* do classificador superior a 0,80, bem como *precisão* mínima de 0,25, pois estes são os patamares mínimos estabelecidos pelos especialistas da instituição financeira.

Logo, pretende-se agilizar a atualização do modelo quando necessário, processar grandes volumes de transações eficientemente e garantir o desempenho e capacidade de generalização do modelo de detecção.

4.2 Entendimento dos Dados

A base de dados transacional é composta por cerca de 24 milhões de transferências eletrônicas realizadas entre dezembro/2013 e maio/2014, devidamente descaracterizada para não comprometer os critérios de identificação de transações fraudulentas utilizados e por questões de sigilo da informação da instituição.

Cada objeto nesta base de dados é composto por 10 atributos de tipo numérico discreto, 7 atributos de tipo numérico contínuo e 5 atributos categóricos. Além destes 22 atributos, foram incluídos outros dois: um atributo para identificar as transações que foram alertadas pelas regras *ad-hoc* elaboradas pelos especialistas da instituição e outro para indicar quais estão rotuladas como fraudulentas.

Para se evitar perda de informação relevante para a construção do modelo não foi omitido qualquer objeto da base de dados no período considerado, tampouco existem atributos com valores desconhecidos.

Neste universo de 24 milhões de transações, existem 6.228 que foram rotuladas como fraudulentas; isto é, apenas 0,026% compõem a classe *fraude*. Daí a necessidade de técnicas de pré-processamento de dados para corrigir o desbalanceamento e tentar garantir a acurácia preditiva na classe minoritária, pois o problema de desbalanceamento de classes pode ser um obstáculo à indução de bons classificadores por algoritmos de AM [25].

Para minimizar o problema das classes extremamente desbalanceadas foram aplicadas duas regras definidas pelos especialistas do domínio que excluem transações que raramente são fraudulentas, estas regras especialistas baseiam-se em perfis transacionais e na agregação de atributos para a extração de informação latente nos dados. A aplicação destas duas regras permitiu reduzir o universo de transações de 24 milhões para cerca de 4,8 milhões. Este subconjunto contém 6.157 transações fraudulentas e as 71 fraudes do conjunto original, que não pertencem a este novo universo, são transações realizadas pelos fraudadores visando despistar uma eventual análise humana, pois pertencem ao perfil do cliente. Assim, foi possível aumentar a proporção de objetos fraudulentos em cerca de 5 vezes e a classe *fraude* passou a representar 0,127% da amostra.

Passou-se, então, à análise qualitativa dos dados antes da etapa de modelagem. Os dados raramente são perfeitos e podem conter erros de codificação, atributos com valores desconhecidos ou outros tipos de inconsistências que comprometem o desempenho de um classificador. Desta análise foram identificados alguns erros de codificação e providenciada o devido ajuste na origem da transação; não há objetos com valores faltantes ou com valores diferentes de seu domínio.

Daí, pode-se afirmar que a base de dados transacional é de boa qualidade para utilização na indução do classificador. Contudo, os 24 atributos transacionais são insuficientes para a tarefa de classificação, segundo os especialistas do domínio, pois não contêm informações referentes ao comportamento histórico dos clientes, por exemplo, e muitos atributos possuem papel de controle, como identificadores de registro. Daí, a necessidade de integração da base de dados original com outras fontes para prover importantes características que auxiliam na discriminação das classes, como dados cadastrais e comportamentais.

4.3 Preparação dos Dados

Uma vez que os dados estão distribuídos em diferentes fontes, é necessária a integração das bases de dados para a classificação das transações. Visando alcançar este objetivo, foram desenvolvidas aplicações em Java para automatizar este processo de integração: as aplicações consultam um data warehouse e em outras fontes de dados históricos e de perfis para integração de outros onze atributos à base de dados transacional. Além disso, são calculados outros oito atributos derivados que, segundo os especialistas do domínio, incluem novas informações relevantes às transações.

Como resultado, a nova base de dados passou a conter objetos com 43 atributos: 24 atributos de tipo numérico discreto, 10 atributos de tipo numérico contínuo e 9 atributos categóricos.

Estes atributos foram nomeados como *v01*, *v02*, ..., *v041*, *v_regra*, *alvo*; onde *v_regra* indica os alertas das regras *ad-hoc* e *alvo* corresponde ao rótulo das transações.

Porém, existem 25 atributos que não contribuem para a categorização das transações, pois ou possuem o mesmo valor para todas as transações ou não agregam nenhuma informação relevante e, por isso, não serão consideradas nas próximas etapas.

Dando continuidade à preparação dos dados e visando reduzir a dimensionalidade da base, mensurou-se a importância dos 18 atributos remanescentes - 16 categóricos/numéricos discretos e 2 numéricos contínuos - na classificação das transações. Para isso foi utilizado o teste χ^2 para estimar o grau de associação entre os atributos nominais/numéricos discretos e o rótulo *fraude* ou *não-fraude*, representado pela variável *alvo*. Além disso, foi verificado também se estes atributos são correlacionados entre si, pois a manutenção de atributos correlacionados pode potencializar o erro do modelo preditivo e diminuir seu desempenho [18].

O grau de associação entre as variáveis pode ser mensurado pelo *coeficiente de contigência de Pearson*, definido pela Equação 2.5, e os resultados obtidos da aplicação do teste χ^2 a 4.835.391 objetos válidos são apresentados na Tabela 4.1.

Tabela 4.1: Coeficiente de contingência entre atributos transacionais

	v02	**v03**	**v05**	**v06**	**v07**	**v08**	**v11**	**v12**	**v13**	**v14**	**v15**	**v17**	**v18**	**v41**	**v42**	**alvo**
v02	-	0,15	0,97	0,10	0,15	0,13	0,02	0,02	0,15	0,15	0,02	0,02	0,09	0,05	0,03	**0,00**
v03	0,15	-	0,17	0,14	0,17	0,19	0,04	0,42	0,26	0,54	0,15	0,63	0,19	0,09	0,04	0,02
v05	0,97	0,17	-	**0,93**	**0,91**	**0,93**	0,04	0,04	0,04	0,05	0,06	0,06	0,06	0,45	0,34	0,03
v06	0,10	0,14	**0,93**	-	**0,93**	**0,92**	0,01	0,01	0,04	0,06	0,05	0,06	0,04	0,40	0,32	**0,00**
v07	0,15	0,17	**0,91**	**0,93**	-	**0,97**	0,01	0,01	0,06	0,07	0,09	0,10	0,05	0,47	0,39	0,01
v08	0,13	0,19	**0,93**	**0,92**	**0,97**	-	0,05	0,06	0,05	0,07	0,08	0,10	0,06	0,57	0,44	0,06
v11	0,02	0,04	0,04	0,01	0,01	0,05	-	0,66	0,01	0,01	0,02	0,05	0,01	0,04	0,03	0,10
v12	0,02	0,42	0,04	0,01	0,01	0,06	0,66	-	0,01	0,01	0,02	0,05	0,02	0,04	0,03	0,08
v13	0,15	0,26	0,04	0,04	0,06	0,05	0,01	0,01	-	0,51	0,01	0,01	0,01	0,04	0,01	0,04
v14	0,15	0,54	0,05	0,06	0,07	0,07	0,01	0,01	0,51	-	0,01	0,01	0,01	0,05	0,01	0,05
v15	0,02	0,15	0,06	0,05	0,09	0,08	0,02	0,02	0,01	0,01	-	0,24	0,03	0,47	0,06	0,02
v17	0,02	0,63	0,06	0,06	0,10	0,10	0,05	0,05	0,01	0,01	0,24	-	0,04	0,12	0,02	0,07
v18	0,09	0,19	0,06	0,04	0,05	0,06	0,01	0,02	0,01	0,01	0,03	0,04	-	0,02	0,01	0,03
v41	0,05	0,09	0,45	0,40	0,47	0,57	0,04	0,04	0,04	0,05	0,47	0,12	0,02	-	0,36	0,44
v42	0,03	0,04	0,34	0,32	0,39	0,44	0,03	0,03	0,01	0,01	0,06	0,02	0,01	0,36	-	0,06
alvo	**0,00**	0,02	0,03	**0,00**	0,01	0,06	0,10	0,08	0,04	0,05	0,02	0,07	0,03	0,44	0,06	-

Em virtude do baixo grau de correlação entre a variável *v02* e o rótulo *alvo*, bem como, entre a variável *v06* e o rótulo *alvo*, expressos pelo coeficiente de contigência, estes atributos não serão considerados durante a etapa de modelagem.

Ainda desta análise de correlação, foram encontrados 4 atributos, com grau de associação relevante: os atributos *v05*, *v06*, *v07* e *v08* são correlacionados, conforme a Tabela 4.1, e será mantido apenas o atributo *v08* por apresentar maior associação com o rótulo *alvo*.

Portanto, serão utilizados doze atributos categóricos/numéricos discretos - *v03*, *v08*, *v11*, *v12*, *v13*, *v14*, *v15*, *v17*, *v18*, *v41*, *v42* e o rótulo *alvo* - e dois atributos numéricos contínuos - *v04*, *v09* - para indução do classificador.

Uma vez que os atributos relevantes à tarefa de classificação já foram selecionados, agora é necessário particionar o conjunto de dados entre as amostras de treinamento, validação e teste.

Para obter um bom classificador é desejável utilizar o máximo de objetos disponíveis para o treinamento e validação; por outro lado, para obter uma boa medida do desempenho do classificador também é desejável utilizar o máximo de dados para a tarefa de teste; além deste dilema, existe outra questão que requer atenção: as amostras usadas podem não ser representativas produzindo um classificador de baixo desempenho [15].

Com relação ao particionamento, em termos práticos, é difícil estabelecer uma regra geral para a alocação dos objetos em cada uma das amostras; um particionamento típico é alocar 50% dos objetos para treinamento e 25% para cada uma das amostras de validação e teste [13]. Com relação à representatividade das amostras, geralmente não é possível afirmar se uma amostra é representativa ou não, para mitigar este problema é utilizado o processo de estratificação: cada classe no conjunto de dados original deve ser representada na mesma proporção nas amostras de teste, validação e treinamento [15].

Logo, é utilizada amostragem aleatória estratificada para particionar os dados em três amostras: 2,4 milhões de transações contendo 3.076 objetos fraudulentos para a amostra de treinamento; 1,2 milhão de objetos contendo 1.571 transações fraudulentas para a amostra de validação e 1,2 milhão de objetos restantes com 1.510 transações fraudulentas para a amostra de teste.

4.4 Modelagem

Neste trabalho, busca-se a construção automática de um classificador para a identificação de transações fraudulentas. Especificamente em aplicações que manipulam grandes volumes de dados, onde o tempo é um fator crítico, é vantajoso adotar algoritmos que exigem menos tempo de treinamento para a indução de novos classificadores, como árvores de decisão [39].

Além disso, a opção pelas árvores de decisão para este trabalho deve-se às seguintes vantagens:

- Árvores de decisão não assumem nenhuma distribuição para os dados [26]. Elas são métodos não-paramétricos. O espaço de objetos é dividido em subespaços e cada subespaço é ajustado com diferentes modelos, fornecendo uma cobertura exaustiva do espaço de objetos.

- Possuem relativa robustez a valores atípicos e ruídos, uma vez que árvores univariáveis são invariantes a transformações estritamente monotônicas dos atributos [12]. Por exemplo, usar x_j ou $\log x_j$ como j-ésimo atributo de entrada produz árvores com a mesma estrutura.

- O algoritmo de construção de uma árvore possui embutido o processo de seleção de atributos relevantes. Esta seleção produz modelos que tendem a ser bastante robustos contra a adição de atributos irrelevantes e redundantes [7].

- Possuem boa interpretabilidade: todas as decisões são baseadas nos valores dos atributos usados para descrever o problema e, portanto, são mais fáceis de interpretar que os pesos numéricos das conexões entre os nós de uma rede neural [20], por exemplo.

- São algoritmos altamente eficientes, sua complexidade de tempo é linear em função do número de objetos e atributos [31].

Para indução do classificador utilizou-se o algoritmo *C5.0* [34], um sistema inicialmente comercial da RuleQuest Research[1] e agora também distribuído sob a licença GNU GPL[2]. Este algoritmo é o sucessor de *C4.5* [32], a estratégia de indução da árvore de decisão no *C5.0* é similar a do *C4.5*: todas as funcionalidades de seu antecessor são mantidas e introduzidas novas tecnologias que resultam na indução de árvores de decisão menores e de acurácia superior, por exemplo [33].

Por sua vez, a avaliação dos classificadores gerados que atingirem os patamares mínimos de sensibilidade e precisão será feita pela medida F_2, definida pela Equação 2.19, onde a sensibilidade possui o dobro do peso da precisão, pois é mais interessante maximizar a sensibilidade com a identificação de mais transações fraudulentas que possuir um classificador de alta precisão que identifica poucas fraudes.

Este viés pela sensibilidade deve-se a peculiaridades do processo de combate às fraudes, no qual as transações apontadas como fraudulentas pelo classificador, *VP+FP*, são disponibilizadas para análise, que será executada por um grupo de especialistas. Esta atividade de análise envolve diversos custos: é desejável maximizar *VP*, isto é, aumentar a quantidade de fraudes identificadas e minimizar *FP*, pois as transações legítimas

[1] `http://www.rulequest.com/`
[2] `http://www.gnu.org/copyleft/gpl.html`

apontadas como fraudulentas geram impacto sobre os clientes envolvidos e oneram os especialistas com trabalho desnecessário.

Além disso, na identificação acurada, o custo financeiro da análise, geralmente, é inferior aos prejuízos decorrentes dos ataques. Por outro lado, quando uma fraude não é detectada, *FN*, os prejuízos decorrentes desta investida são maiores que os decorrentes de fraudes detectadas, *VP*, pois nenhuma ação reativa foi tomada para minimizar os prejuízos.

Por fim, segundo os especialistas do domínio, os limiares mínimos para precisão e sensibilidade são 0,8 e 0,25, respectivamente. Assim, busca-se identificar 80% das transações fraudulentas com uma proporção de uma fraude a cada quatro transações classificadas como fraudulentas.

4.4.1 Árvores de Decisão

Nesta etapa são exibidas as diversas árvores construídas com o algoritmo *C5.0* para a identificação de transações fraudulentas. O algoritmo *C5.0* constrói a árvore através da divisão da amostra de treinamento com base no teste que resulta na maior razão de ganho. Cada subconjunto obtido da primeira divisão é novamente divido pela aplicação de um novo teste e este processo é repetido até que nenhuma outra divisão seja possível. Por fim, a simplificação da árvore com a poda dos nós que não contribuem para a tarefa de classificação é realizada através da poda pessimista embutida no *C5.0*.

A indução da primeira árvore de decisão com o algoritmo *C5.0* foi realizada com os seguintes parâmetros: sem aplicação de custos aos erros de classicação, *FN* e *FP*; nível de confiança da poda pessimista em 25% e número mínimo de objetos a serem alocados em um nó folha configurado em 2.

A Tabela 4.3 e Tabela 4.2 exibem o resultado da avaliação de cada transação pelo classificador obtido nas amostras de treinamento e validação. Os casos positivos representam transações fraudulentas enquanto que os casos negativos representam transações legítimas; uma previsão positiva representa uma possível transação fraudulenta, enquanto uma previsão negativa representa uma possível transação legítima.

Tabela 4.2: Treinamento sem balanceamento para a árvore de decisão

	Previsão Positiva	**Previsão Negativa**
Caso Positivo	998	2.078
Caso Negativo	56	2.414.997

Tabela 4.3: Validação sem balanceamento para a árvore de decisão

	Previsão Positiva	**Previsão Negativa**
Caso Positivo	513	1.058
Caso Negativo	26	1.206.239

Da Tabela 4.3, obtém-se precisão P=95,17%, sensibilidade S=32,65% e F_2=0,3759. Isto é, um classificador de alta precisão e baixa sensibilidade. Este problema deve-se ao fato das classes *fraude* e *não-fraude* serem muito desbalanceadas, daí a necessidade do balanceamento das classes para a identificação dos eventos raros de fraude [8, 25].

Balanceamento

Muitos algoritmos de classificação têm sua acurácia prejudicada quando as classes possuem quantidades de objetos muito diferentes. Nesta situação a classe majoritária pode ser reduzida, a classe minoritária pode ser inflada ou uma combinação de ambas as técnicas [44]. Ao inflar a classe minoritária aumenta-se o custo computacional para a indução do modelo e este pode estar superajustado aos dados [8, 25]; por outro lado, ao diminuir a classe majoritária pode-se excluir objetos relevantes ao modelo e resultar no subajustamento do modelo aos dados [25]. Para evitar a perda de transações relevantes ao modelo optou-se pelo aumento da classe minoritária. Se a classe *fraude* é balanceada com uma quantidade de objetos próxima à da classe *não-fraude*, o algoritmo de classificação possui melhores condições de encontrar os padrões que distinguem as classes [25].

Assim, o balanceamento foi efetuado na amostra de treinamento através da replicação de objetos da classe *fraude* e pela manutenção dos objetos da classe *não-fraude*. Uma vez que a determinação do fator de balanceamento é empírica [8], a classe *fraude* foi aumentada cerca de 785 vezes para a quantidade de objetos fosse igualada à da classe *não-fraude*.

A Tabela 4.4 apresenta a matriz de confusão obtida pela aplicação do modelo gerado com fator de balanceamento 785 na classe minoritária *fraude* da amostra de treinamento, enquanto a e Tabela 4.5 exibe a matriz de confusão obtida pela aplicação da árvore de decisão à amostra de validação.

Tabela 4.4: Treinamento fator balanceamento 785 para a árvore de decisão

	Previsão Positiva	**Previsão Negativa**
Caso Positivo	2.410.416	1.068
Caso Negativo	3.999	2.411.054

Tabela 4.5: Validação fator balanceamento 785 para a árvore de decisão

	Previsão Positiva	**Previsão Negativa**
Caso Positivo	1.170	401
Caso Negativo	3.213	1.203.052

Observa-se da Tabela 4.5 que a precisão na amostra de validação diminuiu para P=26,69%, enquanto a sensibilidade S=74,47% e F_2=0,5484 melhoraram quando comparadas ao desempenho da árvore obtida sem balanceamento dos dados de treinamento. Ao analisarmos com mais atenção a Tabela 4.4 e Tabela 4.5, observa-se que quase todos os objetos fraudulentos, na amostra de treinamento, foram corretamente identificados pela árvore de decisão gerada com uma pequena quantidade de falsos-positivos. Contudo, esse desempenho não é observado na amostra de validação levando a crer que o modelo gerado com fator de balanceamento 785 está sobreajustado aos dados.

A partir de então, o fator de balanceamento, inicialmente em 785, foi reduzido pela metade enquanto houvesse melhoria da métrica F_2 visando a determinação de um máximo local dessa métrica.

Tabela 4.6: Fatores de balanceamento para a árvore de decisão

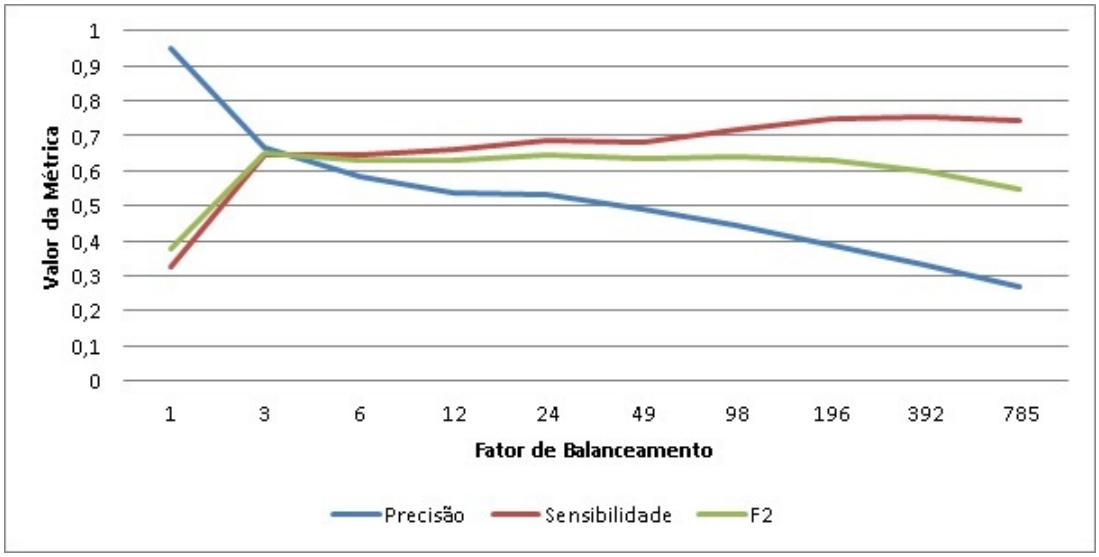
	Amostra de Treinamento			**Amostra de Validação**		
Fator	**Precisão**	**Sensibilidade**	F_2	**Precisão**	**Sensibilidade**	F_2
1	0,947	0,324	0,3735	0,952	0,327	0,376
3	0,931	0,796	0,820	0,667	0,645	**0,649**
6	0,963	0,894	0,907	0,585	0,644	0,631
12	0,980	0,965	0,968	0,537	0,659	0,630
24	0,989	0,979	0,981	0,530	0,686	0,648
49	0,993	0,996	0,995	0,490	0,684	0,634
98	0,994	0,997	0,997	0,446	0,717	0,639
196	0,996	0,998	0,998	0,385	0,749	0,630
392	0,997	0,999	0,999	0,329	0,752	0,598
785	0,997	1,000	0,999	0,267	0,745	0,548

Para cada árvore obtida são calculadas a *precisão* e *sensibilidade* nas amostras de validação e treinamento e a medida F_2 na amostra de validação para avaliação dos classificadores gerados, uma vez que a replicação dos objetos da classe minoritária podem levar ao sobreajustamento do modelo gerado aos dados de treinamento.

Como pode ser observado da Tabela 4.6, o melhor valor para a medida F_2 foi obtido com fator de balanceamento igual a 3, usado daqui em diante.

Da Figura 4.1 pode-se observar a variação das métricas *precisão* e *sensibilidade* e F_2 na amostra de validação obtidas com a aplicação das árvores geradas em função da variação dos fatores de balanceamento.

Figura 4.1: Fatores de balanceamento árvores de decisão



É notório o sobreajustamento do modelo induzido aos dados à medida que o fator de balanceamento tende a 785, pois a precisão do modelo antige valores cada vez menores.

A Tabela 4.7 e Tabela 4.8 exibem as matrizes de confusão para a árvore gerada com fator de balanceamento 3.

Tabela 4.7: Treinamento fator balanceamento 3 para a árvore de decisão

	Previsão Positiva	**Previsão Negativa**
Caso Positivo	7.344	1.884
Caso Negativo	541	2.414.512

Tabela 4.8: Validação fator balanceamento 3 para a árvore de decisão

	Previsão Positiva	**Previsão Negativa**
Caso Positivo	1.013	558
Caso Negativo	506	1.205.759

Com este fator de balanceamento os valores obtidos para as métricas de avaliação do classificador são: P=66,68%, S=64,48% e F_2=0,6491. Apesar da *Precisão* ter superado o valor mínimo esperado, 25%, a *Sensibilidade* ainda está aquém de piso de 80% sendo necessário o ajuste dos demais parâmetros para a obtenção de um classificador que possa alcançar os objetivos estabelecidos pelos especialistas do domínio.

Poda

A extensão da poda da árvore é determinada pelo parâmetro *nível de confiança* que possui valor padrão de 25%. A diminuição deste valor resulta em árvores menores e mais concisas, privilegiando a capacidade de generalização do modelo; por outro lado, o aumento do nível de confiança é usado para a obtenção de árvores de maior acurácia, devido ao maior ajustamento aos dados de treinamento [32].

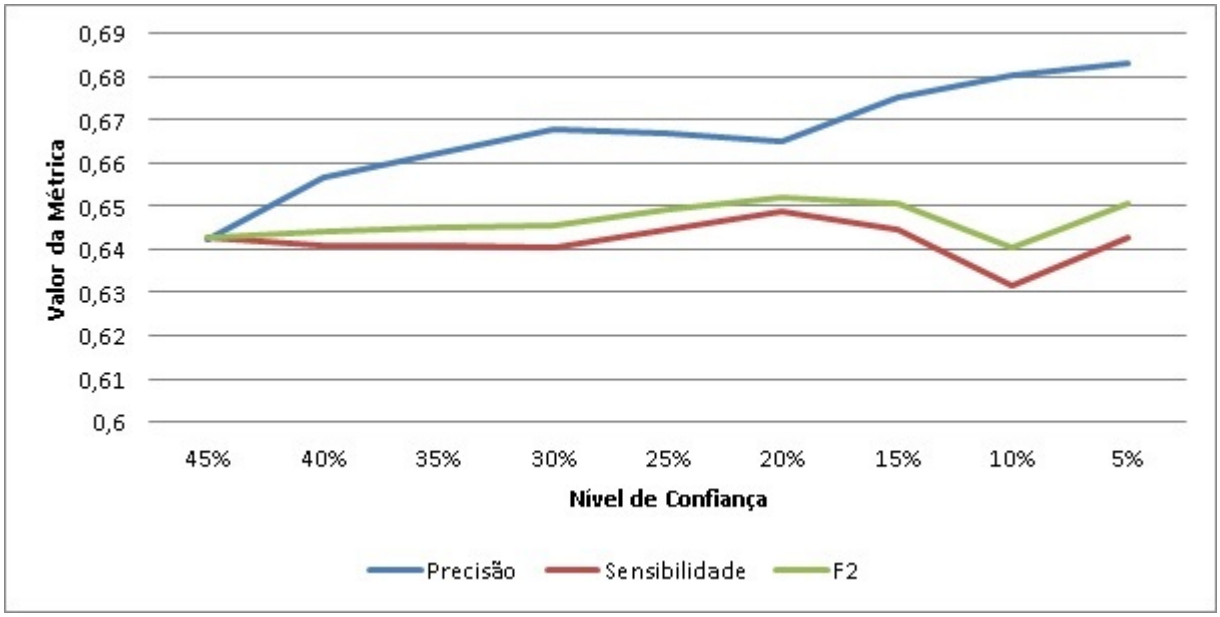
Foram geradas, então, diversas árvores com a alteração do nível de confiança para a determinação do nível de confiança que otimiza a métrica F_2. Observa-se da Tabela 4.9 que o parâmetro *nível de confiança* em 20% resulta no melhor valor para a métrica F_2 dentre os valores testados.

Tabela 4.9: Nível de confiança.

	Amostra de Treinamento			**Amostra de Validação**		
Nível de Confiança	**Precisão**	**Sensibilidade**	F_2	**Precisão**	**Sensibilidade**	F_2
45%	0,943	0,843	0,848	0,642	0,643	0,643
40%	0,944	0,827	0,848	0,657	0,641	0,644
35%	0,941	0,816	0,838	0,662	0,641	0,645
30%	0,938	0,807	0,830	0,668	0,640	0,646
25%	0,931	0,796	0,820	0,667	0,645	0,649
20%	0,924	0,783	0,808	0,665	0,649	**0,652**
15%	0,923	0,762	0,790	0,675	0,645	0,651
10%	0,922	0,728	0,760	0,680	0,631	0,641
5%	0,904	0,704	0,737	0,683	0,643	0,651

Da Figura 4.3 visualiza-se que à medida que o nível de confiança diminui, aumenta-se a capacidade de generalização do modelo e, consequentemente, há o aumento da *sensibilidade* com diminuição da *precisão*.

Figura 4.2: Nível de confiança



Da Tabela 4.10 e Tabela 4.11 obtém-se *Sensibilidade* S=92,36% e *Precisão* P=78,31% na amostra de treinamento e *Sensibilidade* S=64,86% e *Precisão* P=66,51% na amostra de validação.

Tabela 4.10: Treinamento nível de confiança 20%

	Previsão Positiva	**Previsão Negativa**
Caso Positivo	7.227	2.001
Caso Negativo	597	2.414.456

Tabela 4.11: Validação nível de confiança 20%

	Previsão Positiva	**Previsão Negativa**
Caso Positivo	1.019	552
Caso Negativo	513	1.205.752

Ainda é possível verificar se a alteração do critério de parada *default* da árvore resulta em melhoria da acurácia do modelo.

Critério de parada

A árvore gerada pelo algoritmo *C5.0* cresce até que todos os objetos em um nó folha pertençam à mesma classe ou o número de objetos nos nós resultantes da aplicação de um

dado teste condicional não sejam inferiores a um dado limiar. Este limiar atua como um limite no número de testes condicionais aplicados e previne o treinamento desnecessário quando da existência de ruído nos dados [32].

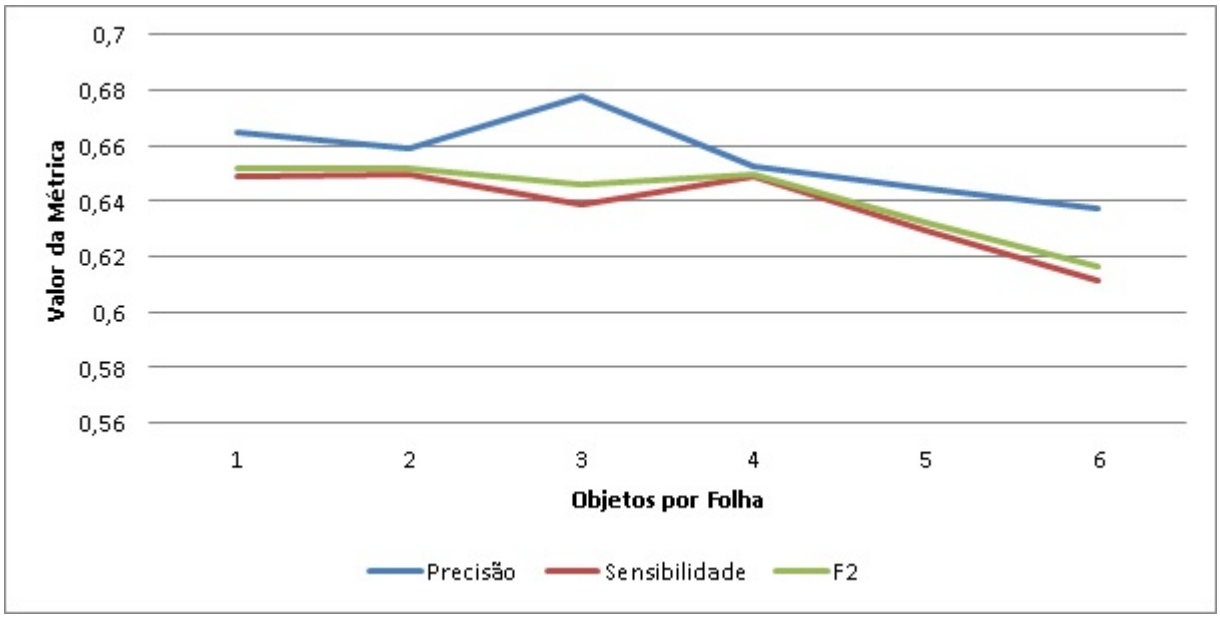
Observa-se da Tabela 4.12 que as métricas *Precisão*, *Sensibilidade* e F_2 variaram minimamente com o aumento do critério de parada e o melhor resultado foi obtido com o valor de 2 objetos por nó folha.

Tabela 4.12: Mínimo de objetos por folha.

	Amostra de Treinamento			**Amostra de Validação**		
Mínimo de objetos por folha	**Precisão**	**Sensibilidade**	F_2	**Precisão**	**Sensibilidade**	F_2
2	0,924	0,783	0,808	0,665	0,649	**0,652**
4	0,905	0,755	0,781	0,659	0,650	0,650
8	0,896	0,712	0,742	0,678	0,638	0,646
16	0,870	0,690	0,719	0,653	0,649	0,650
32	0,861	0,663	0,695	0,645	0,630	0,633
64	0,851	0,637	0,671	0,638	0,612	0,617

Uma vez que a alteração dos critérios de parada não resultou em melhoria significativa da métrica F_2, conforme pode ser observado através da Figura 4.3, foi mantido o valor padrão de no mínimo duas transações por nó folha.

Figura 4.3: Mínimo de objetos por folha



Ainda não foi obtido um classificador que atinja *Precisão* superior a 25% e *Sensibilidade* mínima de 80%, sendo necessário o ajuste de mais um parâmetro para visando ao atingimento dos critérios de sucesso do classificador.

Aplicação de Custos

O algoritmo *C5.0* permite a utilização de custos, quando da ocorrência de classificações incorretas, visando ao aumento da acurácia da árvore de decisão gerada. Dessa forma, a partir da árvore obtida anteriorme com critério de parada igual a 2, foram aplicados diversos custos aos objetos fraudulentos classificados como legítimos, isto é, ao subconjunto de objetos classificados como falsos-negativos.

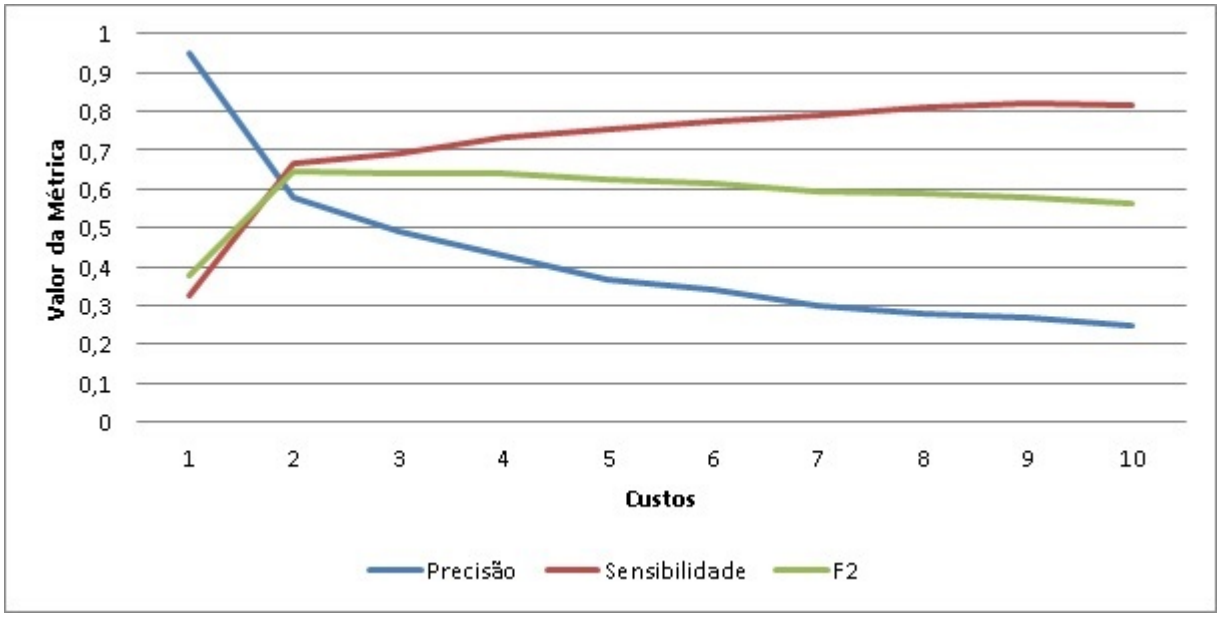
Os resultados obtidos com a aplicação dos modelos gerados às amostras de treinamento e validação podem ser analisados na Tabela 4.13. Destas árvores, a que obteve o melhor valor para a métrica F_2 foi obtida com custo 8 para objetos fraudulentos classificados incorretamente.

Tabela 4.13: Aplicação de custos.

	Amostra de Treinamento			**Amostra de Validação**		
Custos	**Precisão**	**Sensibilidade**	F_2	**Precisão**	**Sensibilidade**	F_2
1	0,924	0,783	0,374	0,665	0,649	0,652
2	0,876	0,823	0,833	0,580	0,666	0,647
3	0,815	0,862	0,852	0,493	0,694	0,642
4	0,763	0,897	0,866	0,430	0,731	0,641
5	0,710	0,923	0,871	0,367	0,754	0,623
6	0,666	0,925	0,858	0,339	0,773	0,616
7	0,618	0,928	0,844	0,301	0,789	0,596
8	0,590	0,942	0,841	0,280	0,810	**0,587**
9	0,565	0,937	0,828	0,267	0,820	0,579
10	0,544	0,943	0,823	0,250	0,817	0,562
11	0,534	0,758	0,699	0,281	0,770	0,571
12	0,492	0,947	0,799	0,217	0,835	0,532

Nota-se através da Figura 4.4 que os modelos gerados com custos 8, 9 e 10 apresentaram *Sensibilidade* superior a 80% e *Precisão* superior a 25% que são os critérios de sucesso estabelecidos pelos especialistas do domínio da instituição financeira; uma vez que a árvore gerada com custo 8 apresentou o melhor valor para a métrica F_2 esta foi selecionada para a identificação das transações fraudulentas.

Figura 4.4: Aplicação de custos



Para a confirmação do desempenho, as transações contidas na amostra de teste foram avaliadas pela árvore selecionada. A partir das Tabelas 4.14, 4.15 e 4.16 pode-se avaliar o desempenho do classificador obtido em cada uma das amostras.

Tabela 4.14: Treinamento aplicação de custos

	Previsão Positiva	**Previsão Negativa**
Caso Positivo	8.691	537
Caso Negativo	6.041	2.409.012

Tabela 4.15: Validação aplicação de custos

	Previsão Positiva	**Previsão Negativa**
Caso Positivo	1.272	299
Caso Negativo	3.270	1.202.995

Tabela 4.16: Teste aplicação de custos

	Previsão Positiva	**Previsão Negativa**
Caso Positivo	1.228	282
Caso Negativo	3.123	1.204.793

Uma vez que a *Sensibilidade* obtida na amostra de teste foi S=81,32% e a *Precisão* foi P=28,22% demonstrou-se que o classificador obtido atingiu os critérios de sucesso da instituição financeira para a identificação de transferências eletrõnicas fraudulentas.

4.4.2 Redes Neurais Artificiais

Para comparação com o modelo gerado através do algoritmo *C5.0*, foram construídas redes neurais artificais perceptron multicamadas, pois este tipo de rede permite a representação de relacionamentos complexos entre os dados [24].

Os modelos gerados através do algoritmo rede neural artifical perceptron multicamadas foram obtidos com o cálculo automático do número de camadas ocultas e de neurônios em cada camada, com tempo máximo de treinamento por unidade de 15 minutos. O primeiro modelo gerado usando a mesma amostra de treinamento e validação utilizada para a criação das árvores de decisão também apresentou precisão superior à sensibilidade: P=68,62% e S=49,71%, conforme avaliação da Tabela 4.17 e Tabela 4.18.

Tabela 4.17: Treinamento sem balanceamento rede neural artificial

	Previsão Positiva	**Previsão Negativa**
Caso Positivo	1.550	1.526
Caso Negativo	641	2.414.412

Tabela 4.18: Validação sem balanceamento rede neural artificial

	Previsão Positiva	**Previsão Negativa**
Caso Positivo	781	790
Caso Negativo	357	1.205.908

Daí, a necessidade de balanceamento das classes *fraude* e *não-fraude* para que o classificador gerado possa melhorar sua sensibilidade.

Balanceamento

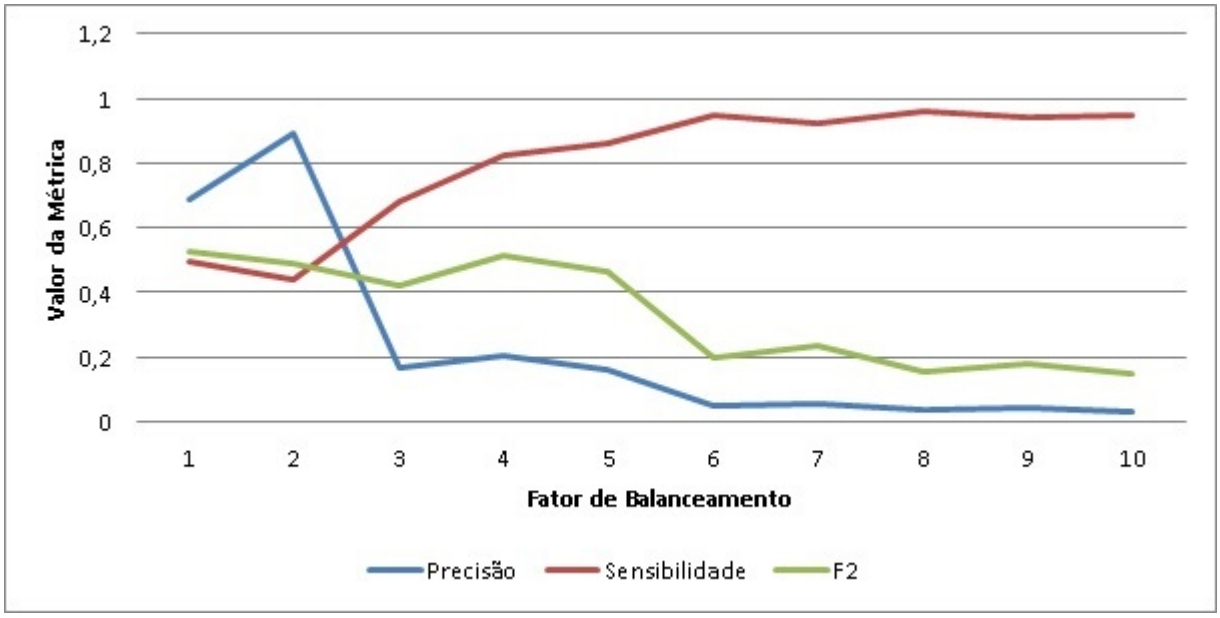
Para a determinação do fator de balanceamento para a rede neural artifical procedeu-se de maneira similar à utilizada para a criação das árvores de decisão: a classe *fraude* na amostra de treinamento foi aumentada 785 vezes para que contivesse a mesma quantidade de transações da classe *não-fraude*; a partir de então, o fator de balanceamento foi ajustado em busca da melhoria da métrica F_2, conforme Tabela 4.19.

Tabela 4.19: Fatores de balanceamento classe minoritária para as redes neurais

	Amostra de Treinamento			**Amostra de Validação**		
Fator de Balanceamento	**Precisão**	**Sensibilidade**	F_2	**Precisão**	**Sensibilidade**	F_2
1	0,707	0,504	0,535	0,686	0,497	**0,526**
3	0,891	0,445	0,494	0,894	0,442	0,492
6	0,165	0,702	0,426	0,166	0,681	0,420
12	0,207	0,848	0,523	0,207	0,827	0,517
24	0,164	0,877	0,469	0,164	0,859	0,465
49	0,047	0,955	0,197	0,048	0,947	0,200
98	0,058	0,947	0,233	0,058	0,926	0,233
196	0,035	0,968	0,152	0,035	0,959	0,154
392	0,042	0,961	0,178	0,042	0,944	0,179
785	0,033	0,957	0,146	0,034	0,946	0,148

Observa-se que o classificador gerado com fator de balanceamento 785 apresentou elevada sensibilidade e baixíssima precisão na amostra de validação, evidenciando o sobreajustamento do modelo aos dados de treinamento. Após a criação de diversos modelos, o melhor valor obtido para a métrica F_2 dentre os modelos gerados foi F_2=0,526 sem aumento da classe *fraude*, conforme Figura 4.5.

Figura 4.5: Fatores de balanceamento classe minoritária para as redes neurais



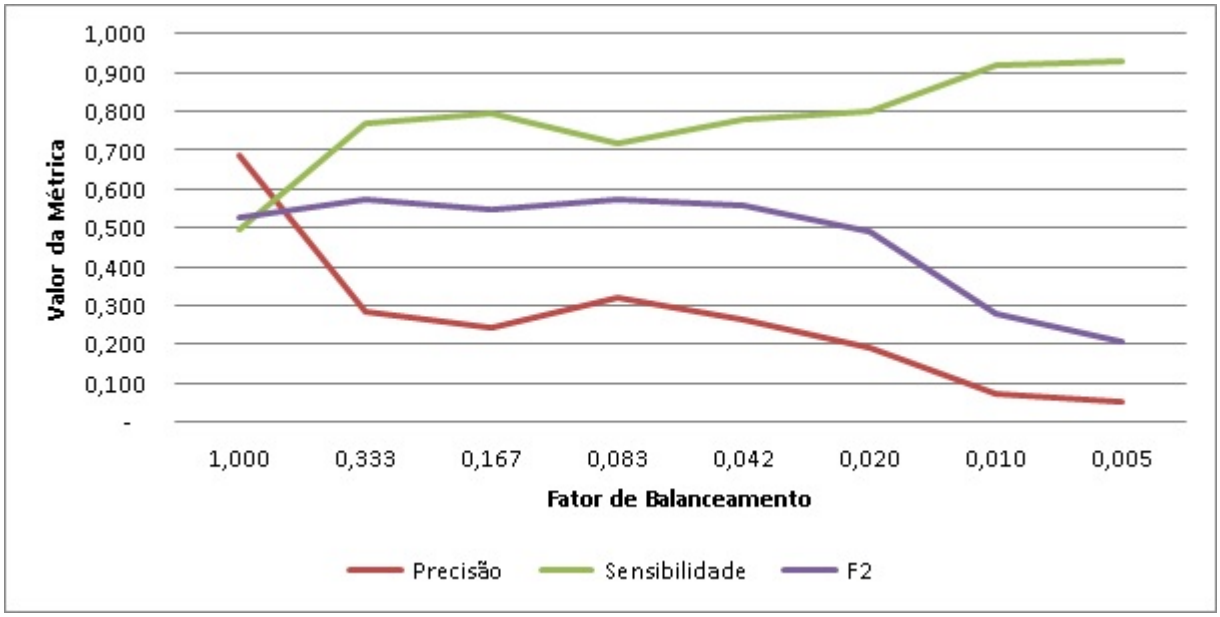
Uma vez que não foi possível a obtenção de um classificador que atendesse aos valores mínimos para a *Sensibilidade* e *Precisão*, procedeu-se com o tratamento do desbalanceamento dos dados com a redução da classe majoritária, conforme Tabela 4.20.

Tabela 4.20: Fatores de balanceamento classe majoritária para as redes neurais

	Amostra de Treinamento			**Amostra de Validação**		
Fator de Balanceamento	**Precisão**	**Sensibilidade**	F_2	**Precisão**	**Sensibilidade**	F_2
1,000	0,707	0,504	0,535	0,686	0,497	0,526
0,333	0,284	0,789	0,582	0,283	0,771	0,573
0,167	0,238	0,813	0,548	0,242	0,797	0,546
0,083	0,319	0,723	0,577	0,320	0,716	**0,574**
0,042	0,262	0,800	0,567	0,261	0,781	0,559
0,020	0,192	0,822	0,497	0,191	0,801	0,488
0,010	0,073	0,933	0,279	0,074	0,917	0,280
0,005	0,050	0,941	0,208	0,051	0,932	0,209

Da Figura 4.6 nota-se que os resultados obtidos para as métricas *Precisão* e *Sensibilidade* com a avaliação da amostra de validação também ficaram abaixo dos limiares estabelecidos como mínimos para aceitação de um classificador.

Figura 4.6: Fatores de balanceamento classe majoritária para as redes neurais



4.5 Avaliação

Os patamares mínimos estabelecidos pelos especialistas da instituição financeira foram: identificação de pelo menos 80% das fraudes com precisão de 25%. Para isso, os classificadores foram contruídos com uma base de dados contendo cerca de 4,8 milhões de transações com 6.157 transações fraudulentas que foram particionadas em três amostras: 50% dos dados para a amostra de treinamento e 25% para cada uma das amostras de validação e teste.

Na amostra de teste, as regras *ad-hoc* concebidas pelos especialistas de domínio atingiram sensibilidade de 66,79% e precisão de 11,31%. Na Tabela 4.21 temos as métricas de desempenho das regras *ad-hoc*, bem como, da árvore de decisão selecionada nas etapas anteriores deste trabalho.

Tabela 4.21: Avaliação dos modelos

Modelos	**Precisão**	**Sensibilidade**	F_2
Regras *ad-hoc*	10,23%	65,21%	0,3142
Árvore de decisão	28,22%	81,32%	**0,5908**

Nota-se que a árvore de decisão obteve desempenho superior à regras *ad-hoc*; além disso, a precisão e a sensibilidade superaram os limiares mínimos estabelecidos pelos especialistas do domínio. Portanto, foi possível melhorar a precisão na identificação de transferências fraudulentas em cerca 175%, minimizando a quantidade de falsos-positivos gerados e, por consequência, podendo reduzir o impacto sobre os clientes e sobre a equipe de tratamento de alertas. Além disso, melhorou a sensibilidade em cerca de 25% viabilizando a redução do prejuízo financeiro com fraudes na modalidade de transação monitorada.

Cumpre citar que a interpretabilidade da árvore de decisão provê maior entendimento dos relacionamentos do atributo alvo, *fraude* ou *não-fraude*, e os atributos transacionais: possibilitando a extração de insumos importantes para o aprimoramento das regras de negócio da instituição, como, por exemplo, a diminuição dos limites transacionais para determinado público.

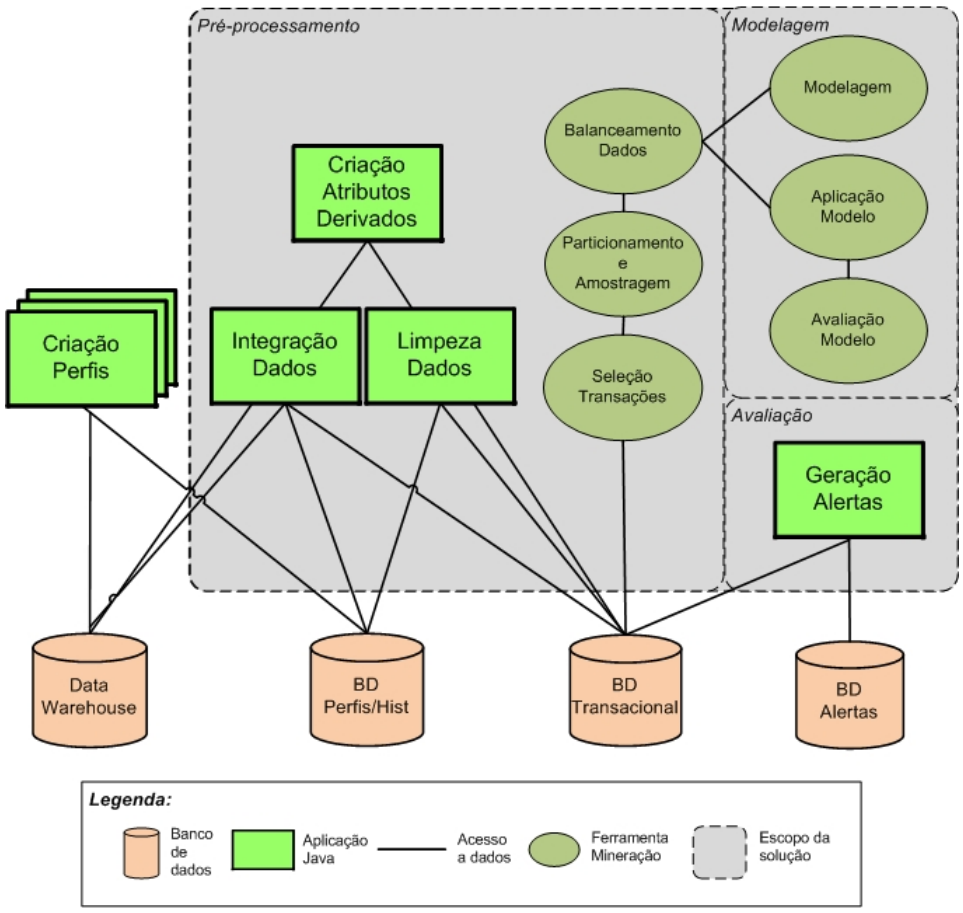
4.6 Implantação

Nesta etapa foi realizada a incorporação do classificador obtido, a árvore de decisão, no processo de negócio. Para sua implantação foi utilizado o formato PMML [42] que permite o compartilhamento dos modelos de AM entre diferentes plataformas. A árvore de decisão no formato PMML foi interpretada por uma aplicação construída com base

na API Java JPMML-Evaluator [37], *Java Evaluator API for Predictive Model Markup Language (PMML)*, após a inclusão de classes para comunicação com os bancos de dados utilizados. Esta aplicação buscou as transferências bancárias de uma base de dados, submeteu as mesmas à árvore de decisão para geração do escore e classicação de cada transação como *fraude* ou *não-fraude*. Este resultado foi finalmente armazenado em banco de dados pela aplicação.

Na Figura 4.7 é exibida a arquitetura da solução concebida para a sistematização da indução dos classificadores - composta pelas camadas de pré-processamento, modelagem e avaliação - em tempo de execução.

Figura 4.7: Visão em tempo de execução



A camada de pré-processamento realiza a automação das etapas de limpeza dos dados, integração entre as diferentes fontes de dados - data warehouse, banco de dados históricos e de perfis transacionais e o banco de dados transacional - e criação de atributos derivados; além disso, realiza as etapas de seleção de transações através da aplicação de regras especi-

alistas, particionamento e amostragem aleatória estratificada, bem como, balanceamento dos dados para a atualização do classificador.

A camada de modelagem é responsável pela atualização do modelo de identificação de transações fraudulentas utilizando as transações preparadas de forma automática pela camada de pré-processamento. A atualização é realizada sob demanda quando do surgimento de uma nova modalidade de ataque ou quando observa-se redução nas métricas precisão e sensibilidade. A cada atualização é gerada uma nova árvore de decisão que é exportada no formato PMML para utilização na camada de avaliação de transações.

A integração destas duas camadas permitiu imprimir celeridade na atualização dos modelos de identificação de fraudes, pois cada nova árvore é gerada, em média, em 15 minutos. Isto permite rápidas reações a novos ataques em comparação à atualização das regras *ad-hoc* concebidas pelos especialistas do domínio que levam algumas horas ou dias para atualização e análise dos resultados.

Por fim, na camada de avaliação, o classificador atualizado é utilizado para a avaliação das transações selecionadas como críticas e o escore gerado é armazenado em banco de dados para uma eventual análise humana. Uma vez que o modelo no formato PMML é um parâmetro da aplicação Java que realiza a avaliação das transações, a implantação de um novo modelo no processo de negócio é imediata.

Dessa forma, com a automação das etapas de pré-processamento dos dados foi possível a sistematização da indução de classificadores para identificação de fraudes, obtendo-se celeridade na atualização, avaliação e implantação do modelo de detecção quando do surgimento de novos ataques. Além disso, com a utilização do avaliador, a árvore de decisão pode ser integrada ao processo de negócio da instituição para a identificação das transações com indícios de fraude. Logo, a contribuição tecnológica esperada para este trabalho foi plenamente alcançada.

Capítulo 5

Conclusões e Trabalhos Futuros

5.1 Conclusões

Buscou-se neste trabalho prover modelos de identificação de fraudes em tranferências bancárias que fossem rapidamente adaptáveis a novos ataques, fossem eficientes para processar grandes volumes de transações e, claro, acurados na identificação de fraudes. Para tanto, foram automatizadas as etapas de preparação dos dados, as informações geradas foram disponibilizadas para a criação de árvores de decisão para a identificação de fraudes e os modelos criados puderam ser integrados ao processo de negócio.

A etapa de preparação dos dados foi a que mais consumiu tempo para automação, pois as informações necessárias à identificação das transações fraudulentas estavam dispersas em diversas fontes de dados. Passou-se então à etapa de modelagem e verificou-se que o desempenho dos classificadores estava aquém dos limiares estabelecidos de *precisão* e *sensibilidade*, sendo necessário o retorno à etapa de preparação dos dados para a introdução de novas variáveis e adequada seleção de atributos relevantes.

Na etapa de modelagem, cuidado especial foi dado ao problema do desbalanceamento dos dados para que o classificador pudesse capturar as características que distinguem as classes *fraude* e *não-fraude* e, ao mesmo tempo, não ficasse sobreajustado aos dados de treinamento.

Com a exportação do modelo no formato PMML e utilização da API JAVA JPMML-Evaluator foi possível implementar a avaliação das transações e armazenar o escore gerado pela árvore de decisão.

Assim, foi possível criar sistematicamente, através do sistema desenvolvido, árvores de decisão para identificação de transações bancárias fraudulentas. Isto possibilitou celeridade na atualização do modelo quando do surgimento de novas modalidades de fraude, diminuindo a dependência de intervenção humana e dependência de especialistas.

Uma vez que a precisão do classificador foi superior à precisão das regras *ad-hoc* da instituição é possível diminuir os custos de tratamento de alertas e com o aumento da sensibilidade reduzir os prejuízos com as fraudes em nível nacional, pois todas as transferências bancárias podem ser monitoradas.

Portanto, concluímos que a aplicação de técnicas de Aprendizado de Máquina, árvores de decisão, para a identificação de transações fraudulentas apresentou resultados superiores aos alcançados pelo sistema vigente na instituição financeira e indica que sua adoção, acompanhada de medidas reativas, pode reduzir os prejuízos financeiros, aumentar a recuperação de valores e diminuir o risco de dano à imagem da instituição, bem como, o desgaste junto aos clientes.

5.2 Trabalhos Futuros

Dada a dinamicidade dos ataques realizados pelos canais virtuais, é necessário a criação de uma plano de monitoração e manutenção do sistema de identificação de transações fraudulentas.

Assim, o modelo em produção deve ser avaliado periodicamente para garantir sua efetividade e, eventualmente, para introdução de melhorias.

Dentre as tarefas a serem implementadas estão:

- mensurar e monitorar a validade e acurácia do modelo em produção, através do cruzamento de informação entre os apontamentos do classificador e a análise das contestações dos clientes;
- através da análise de novos ataques, identificar e introduzir novos atributos no modelo visando à melhoria do desempenho;
- incoporar as descobertas obtidas da interpretação da árvore no processo de negócio;
- efetuar mais experimentos com outros algoritmos de AM.

Referências

[1] R. Wheeler; S. Aitken. Multiple algorithms for fraud detection. *Knowledge-Based Systems*, 1:1–12, 2000. 14

[2] Wen-Ching Lin Graham Williams Alex Guazzelli, Michael Zeller. PMML: An Open Standard for Sharing Models. *The R Journal*, 1:60–65, 2009. 23

[3] BACEN. Resolução nº 003380, de 29 de junho de 2006 - Dispõe sobre a implementação de estrutura de gerenciamento do risco operacional. Technical report, Banco Central do Brasil, 2006. 3

[4] CRISP-DM Consortium. CRISP-DM 1.0, 1999. 24, 25

[5] G. de A. Martins. *Estatística geral e aplicada*. Atlas, 2010. 10, 11

[6] Steve Donoho. Early detection of insider trading in option markets. *ACM Sigkdd Explorations*, pages 22–25, 2004. 14

[7] K. Facelli; et al. *Inteligência Artificial: Uma Abordagem de Aprendizagem de Máquina*. LTC, 2011. 6, 8, 11, 12, 13, 14, 15, 16, 20, 21, 34

[8] N. V. Chawla; et al. Special issue on learning from imbalanced data. *ACM Sigkdd Explorations*, 6:1–6, 2004. 8, 36

[9] T. Hancock; et al. Lower bounds on learning decision lists and trees. *Information and Computation*, 126:114–122, 1996. 16

[10] Y. Kou; et al. Survey of fraud detection techniques. *Proceedings of the 2004 IEEE International Conference on Networking, Sensing & Control*, 1:749–754, 2004. 14

[11] FEBRABAN. A sociedade conectada: Setor bancário em números, tendências tecnológicas e agenda atual. In *CIAB Febraban*, 2012. 1

[12] J. Friedman. Greedy function aproximation: a gradient boosting machine. Technical report, Statistics Departament, Stanford University, 1999. 34

[13] Trevor Hastie; Robert Tibshirani; Jerome Friedman. *The Elements of Statistical Learning Data Mining, Inference and Prediction*. Springer-Verlag London, 2008. 33

[14] João Gama. *Knowledge Discovery from Data Streams*. CRC Press, 2010. 13

[15] Ian H. Witten; Frank Eibe; Mark A. Hall. *Data Mining - Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Publishers, 2011. 33

[16] R. J. Bolton; D. J. Hand. Statistical fraud detection: A review. *Statistical Science*, 17:235–255, 2002. 14

[17] Ali A. Ghorbani John Zhong Lei. Improved competitive learning neural networks for network intrusion and fraud detection. *Neurocomputing*, 75:135–145, 2012. 14

[18] M. Kuhn; K. Johnson. *Applied Predictive Modeling*. Springer-Verlag London, 2013. 9, 10, 12, 32

[19] E. W. S. Khin. Detection artificial intelligence to minimize internet fraud. *International Journal of Cyber Society and Eduacation*, 2:61–72, 2009. 14

[20] S. B. Kotsiants. Decision trees: a recent overview. *Springer Science*, 39:261–283, 2011. 34

[21] Stephan Kovach. *Detecção de Fraudes em Transações Financeiras em Tempo Real*. PhD thesis, USP, 2011. 14

[22] S.; et al. Kumar. A pattern matching model for misuse intrusion detection. *Proceedings of the National Computer Security Conference*, pages 11–21, 1994. 14

[23] Pang Su lin; Gong Ji-zhang. C5.0 Classification Algorithm and Appplication on Individual Credit Evaluation of Banks. *Systems Engineering - Theory & Practice*, 29:94–104, 2009. 14

[24] T. M. Mitchell. *Machine Learning*. McGraw-Hill, 1997. 4, 13, 14, 16, 44

[25] G. E. A. P. A. Batista; R. C. Prati; M. C. Monard. A study of the behavior of several methods for balancing machine learning training data. *ACM Sigkdd Explorations*, 6:20–29, 2004. 2, 7, 8, 30, 36

[26] S. K. Murthy. Construction of decision trees from data: A multi-disciplinary survey. *Data Mining and Knowledge Discovery*, 2:345–389, 1998. 16, 34

[27] N. J. Nilsson. *Principles of Artificial Intelligence*. Palo Alto, CA: Tioga Publishing Company, 1980. 4

[28] S. J. Russel; P. Norvig. *Artificial Intelligence A Modern Approach*. Prentice Hall, 1995. 13

[29] V.Malathi P.Senthil Vadivu. Decision Trees for handling Uncertain Data to identify bank Frauds. *International Journal of Wireless Communications and Networking Technologies*, 1:38–41, 2012. 14

[30] PwC. Pesquisa global de segurança da informação. Technical report, PwC, http://www.pwc.com.br/giss2013, 2013. 2, 3

[31] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1:81–106, 1986. 15, 18, 20, 34

[32] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., 1993. 14, 15, 17, 18, 19, 20, 34, 39, 41

[33] Rulequest Research. Is See5/C5.0 Better Than C4.5? http://www.rulequest.com/see5-comparison.html., 2012. 34

[34] Rulequest Research. Data mining tools See5 and C5.0. http://www.rulequest.com/see5-info.html, 2013. 34

[35] L. Hyafil; R. L. Rivest. Constructing optimal binary decision trees is np-complete. *Information Processing Letters*, 5:15–17, 1976. 16

[36] U.; Neumann E.; Idan Y.; Pinkas G. Rosset, S.; Murad. Discovery od fraud rules for telecommunications challenges and solutions. *ACM Sigkdd*, pages 409–413, 1999. 14

[37] Villu Ruusmann. JPMML-Evaluator. https://github.com/jpmml/jpmml-evaluator, 2013. 48

[38] Y. Sasaki. The truth of the f-measure. *University of Manchester*, 2007. 22

[39] L. Tjen sien; L. Wei-Yin; S. Yu-Shan. A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms. *Machine Learning*, 40:203–229, 2000. 33

[40] M. Mohri; A. Rostamizadeh; A. Talwakar. *Foundations of machine learning*. MIT Press, 2012. 4, 5, 6, 7, 12, 13

[41] D. M. J. Tax. *One-class classification: concept-learning in the absence of counter-examples*. PhD thesis, University of Delft, 2001. 8

[42] The Data Mining Group Consortium ("DMG"). PMML Standard. http://www.dmg.org/v4-2-1/GeneralStructure.html, 2013. 22, 47

[43] Foster Provost Tom Fawcett. Adaptive fraud detection. *Data Mining and Knowledge Discovery*, 1:291–316, 1997. 14

[44] S. Bhattacharyya; S. Jha; K. Tharakunnel; C. Westland. Data mining for credit card fraud: A comparative study. *Decision Suport Systems*, 50:602–613, 2011. 7, 14, 36

[45] Allen Kent; James G. Williams. *Encyclopedia of Microcomputers: Volume 28 (Supplement 7)*. CRCPress, 2002. 22