# ESTRATÉGIAS DE SELEÇÃO DE ATRIBUTOS PARA DETECÇÃO DE ANOMALIAS EM TRANSAÇÕES ELETRÔNICAS

RAFAEL ALEXANDRE FRANÇA DE LIMA

# ESTRATÉGIAS DE SELEÇÃO DE ATRIBUTOS PARA DETECÇÃO DE ANOMALIAS EM TRANSAÇÕES ELETRÔNICAS

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais - Departamento de Ciência da Computação
como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação.

Orientador: Adriano César Machado Pereira

Belo Horizonte

Junho de 2016

Lima, Rafael Alexandre França de

B333o Estratégias de seleção de atributos para detecção de anomalias em transações eletrônicas / Rafael Alexandre França de Lima. — Belo Horizonte, 2016
xvi, 94 f. : il. ; 29cm

Dissertação (mestrado) — Universidade Federal de Minas Gerais - Departamento de Ciência da Computação

Orientador: Adriano César Machado Pereira

1. Computação-Teses. 2. Mineração de Dados-Teses. 3. Detecção de Anomalias. 4. Fraude na Internet-Teses. 5. Seleção de Atributos. I. Orientador. II. Título.

CDU 519.6*73.(043)

UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

# FOLHA DE APROVAÇÃO

Estratégias de seleção de atributos para detecção de anomalias em transações eletrônicas

**RAFAEL ALEXANDRE FRANÇA DE LIMA**

Dissertação defendida e aprovada pela banca examinadora constituída pelos Senhores:

PROF. ADRIANO CÉSAR MACHADO PEREIRA - Orientador
Departamento de Ciência da Computação - UFMG

PROF. ANÍSIO MENDES LACERDA
Departamento de Computação - CEFET

PROFA. GISELE LOBO PAPPA
Departamento de Ciência da Computação - UFMG

PROF. WAGNER MEIRA JÚNIOR
Departamento de Ciência da Computação - UFMG

Belo Horizonte, 15 de junho de 2016.

# Agradecimentos

Primeiro, agradeço a Deus, por ter me dado saúde, força e sabedoria para cumprir essa meta tão especial. Agradeço também a ele por ter colocado no meu caminho pessoas que fizeram essa caminhada ser mais fácil e agradável. Existem muitas pessoas que gostaria de agradecer por terem contribuído para esse objetivo, mas por limitação do espaço separei algumas delas que preciso dizer um muito obrigado.

A minha mãe Maria José França Lima e meu pai Arcanjo Jacinto de Lima, pois se hoje tenho minhas mãos macias é devido as mãos calejadas desses dois, que me deram a educação, apoio e estrutura necessária para vencer as batalhas da vida. Aos meus irmãos Giselle Lima, Leonardo Lima e Giovanni Lima que sempre foram um espelho para mim, o reflexo de quem eu queria ser quando crescesse. A minha vó Silvia Hallais pelo carinho e brilhos nos olhos de sempre. A minha namorada, futura noiva e mãe dos meus filhos Thaís Felix (Potchaca), pelo carinho, compreensão e incentivo de sempre. Por há quase 10 anos estar sempre me incentivando, mesmo ouvindo um hoje não posso, tenho que estudar.

A minha sogra Romeria Felix, meu sogro Emir Felix, minhas cunhadas Adriana Santos, Lara Felix, Priscila Felix e meu cunhado Wallace Loures por todo o incentivo e apoio prestado nessa caminhada. A minha sobrinha Larissa por fazer o "titio El" querer ser um exemplo para ela.

Aos meus amigos do tempo de Cefet que tanto contribuíram para fazer os 5 anos de graduação serem mais agradáveis. Em especial Manoel, Débora, Léo, Lucas, Boninho, Nenas, Luiz, Henrique, Barbara e Marreta. Aos amigos adquiridos no mestrado que me proporcionaram inúmeras boas conversas sobre algoritmos, problemas, provas, bandejão, futebol, política e outras asneiras mais. Em especial André (o moço da falta de sorte), Artur (o forte), Clebson (o cara do coração bom), Leandro (o companheiro), Luis Pedraza (o colombiano mais mineiro) e Thiago (o Zagueiro Gordinho). Aos meus companheiros do laboratório *Speed* Alex, Camila, Denise, Elverton, Filipe, Fernandinho, Júlio, Osvaldo, Renno, Samuel, Vinícius e Walter pelos diversos aprendizados que me proporcionaram. Em especial ao amigo Paulo Bicalho, por ter me feito sentir tão

em casa nesse laboratório.

Aos integrantes da melhor carona do Brasil, por terem feito com que as minhas idas e vindas para UFMG fosse muito mais alegre. Em especial Jean Freire, Lucas Castro, Paulo Bicalho e Thássia Almeida. Aos demais amigos Luzienses que reencontrei na UFMG. Em especial agradeço aos ensinamentos e sempre bom bate papo com Ruhan Bidart e Fabricio Rodrigues. Agradeço aos parceiros dos times, Estudiantes de La Farra e Meia Boca Juniors, do campeonato de futebol do DCC. Aos meus amigos "Touros", por ter me apoiado nas diversas vezes que disse estar "agarrado" e não poder comparecer aos eventos.

Ao meu amigo e orientador Adriano C.M Pereira, por ter despertado em mim a paixão pela Pesquisa. Por ter me recebido como um aluno de Iniciação Científica em 2011 e me tornando um mestre 5 anos depois. Por ter sido sempre tão atencioso, paciente e companheiro, por me orientar não apenas academicamente, mais me oferecer ensinamentos para toda a vida.

Enfim, muito obrigado a todos aqueles que contribuíram direto ou indiretamente para a concretização desse trabalho.

*"It is not the strongest of the species that survives, nor the most intelligent that survives. It is the one that is most adaptable to change."*

(Charles Darwin)

# Resumo

Detecção de anomalias é o problema de encontrar padrões que não se comportam de acordo com o esperado. Um dos cenários clássicos é a detecção de fraudes, que consiste em, a partir de um conjunto de observações, aprender um comportamento fraudulento. Em transações eletrônicas existe um volume muito grande de informações que podem ser utilizadas para detectar fraudes. Assim, filtrar esse conjunto de informações e escolher as melhores é uma tarefa crucial, conhecida como seleção de atributos. Os melhores métodos de seleção de atributos baseiam-se em informações da classe, ou variável resposta. Contudo, uma característica marcante dos problemas de detecção de anomalias é o alto desbalanceamento entre as classes, o qual gera um novo desafio para as técnicas de seleção de atributos, as quais tendem a selecionar os atributos a favor da classe dominante. Neste trabalho foram analisadas estratégias de seleção de atributos para detecção de anomalias em transações eletrônicas, categorizadas em duas abordagens principais. A primeira abordagem consiste na aplicação de 7 métodos de *resampling*, incluindo um criado neste trabalho, para reduzir o desbalanceamento entre as classes antes da seleção de atributos realizadas por 3 técnicas tradicionais. A segunda abordagem consiste na avaliação de 8 métodos de seleção de atributos considerados insensíveis ao desbalanceamento entre as classes, além da criação de um método que utiliza o conceito de Fronteira de Pareto para combinação das métricas. A validação sobre a eficácia dos métodos foi realizada construindo modelos de detecção de fraude, formados por 3 diferentes técnicas de classificação sobre os atributos selecionados pelas distintas abordagens. Para validação desses modelos, realizamos estudos de casos com dados reais, para detecção de fraudes em 2 sistemas de pagamentos eletrônico. Através dos experimentos realizados, verificamos nossas hipóteses de pesquisa, observamos comportamentos interessantes sobre a seleção de atributos para detecção de anomalias e construímos modelos mais efetivos para detectar fraudes. Os melhores modelos apresentaram ganhos econômicos de até 57% sobre o cenário real.

**Palavras-chave:** Detecção de Anomalias, Seleção de Atributos, Mineração de Dados.

# Abstract

Anomaly detection refers to the problem of finding patterns in data that deviates from the expected average behavior. One of the classic scenarios in this area is fraud detection, which consist in learn a fraudulent behavior from a set of observations. In electronic transactions, there is a large amount of information that could be used to detect fraud. Thus, filter this information and choose the most representative of it is a crucial task, known as Feature Selection. The best Feature Selection methods uses the class information to perform this task. However, an important characteristic in fraud detection problems is the high imbalance between the classes. This behavior generates a new challenge to Feature Selection techniques, which tend to select features in favor of the dominant class. Therefore, in this work we analyzed feature selection strategies to anomaly detection in electronic transactions. These strategies were divided in two distinct approaches. In the first approach we applied 7 resampling methods, including one created in this work, to reduce the imbalance between classes before feature selection step. In the second approach we evaluated 8 feature feature selection methods, considered insensitive to imbalance between the classes and we also create a method that uses the concept of Pareto Frontier to combine metrics. The validation of the effectiveness of the methods was performed building fraud detection models. This was performed applying 3 different classification techniques on the attributes selected by different approaches. To validate these models we performed case studies to fraud detection in 2 real dataset from electronic payment systems. We evaluate these models by 3 different metrics. Trough this experiments, we validate our research hypothesis, providing contributions to feature selection area in order to detect fraud. The best models achieved economic gains of up to 57% compared to the actual scenario of the company.

**Keywords:** Anomaly Detection, Feature Selection, Data Mining, Fraud Detection.

# Lista de Figuras

# Lista de Tabelas

# Sumário

# Capítulo 1

# Introdução

Detecção de anomalias é o problema de encontrar padrões que não se comportam de acordo com o esperado. Esses padrões são muitas vezes denominados *outliers*, anomalias, aberrações, peculiaridade e exceções [Chandola et al., 2009]. As anomalias diferem-se dos ruídos, que são normalmente encontrados nos dados. Em [Enderlein, 1987], ruídos são definidos como fenômenos que não são interessantes para análise e necessitam ser removidos para uma correta mineração sobre uma base de dados, enquanto anomalias trazem informações e comportamentos importantes, necessitando, assim, serem detectadas e interpretadas corretamente.

Existem diversos cenários nos quais Detecção de Anomalia é indicada. Em [Chandola et al., 2009] são apresentadas algumas aplicações, tais como:

- Detecção de intrusos, caracterizado por ataques em sistemas de computadores que apresentam comportamento diferente de um comportamento normal;
- Pertubações no ecossistema, tais como: furacões, secas, enchentes e ondas de calor;
- Na indústria, onde um comportamento anômalo de uma máquina pode ser utilizado para detecção de falhas;
- Na área médica um sintoma anômalo pode indicar uma doença;
- Detecção de fraude, um dos cenários clássicos de detecção de anomalia, que se baseia na hipótese de que usuários fraudulentos apresentam comportamentos distintos de usuários não fraudulentos.

Um cenário onde detectar anomalias se torna fundamental são as transações eletrônicas, já que, devido à popularidade dos serviços Web e o grande volume de in-

formações gerado, existem diversos comportamentos que precisam ser investigados para garantir a credibilidade, segurança e melhorar a interação entre os usuários [Tseng & Fogg, 1999]. Dentre esses, destaca-se a detecção de fraudes em transações de comércio eletrônico (*e-commerce*), definido como qualquer transação realizada entre um comprador e um vendedor por meio de um equipamento eletrônico [Cameron, 1997].

Os trabalhos que lidam com este tópico procuram encontrar padrões que podem ser considerados como um comportamento anômalo. Para isso, utilizam-se técnicas estatísticas, aprendizado de máquina e mineração de dados para aprender os padrões [Richhariya & Singh, 2012, Bhattacharyya et al., 2011, Zhang et al., 2013, Kim et al., 2013, Ngai et al., 2011b, Almendra, 2013, Lima & Pereira, 2012], desenvolvendo, assim, modelos que sejam capazes de identificar *outliers*, que são comportamentos diferentes do padrão esperado, por exemplo, fraudes em transações eletrônicas.

Embora possamos encontrar diversos trabalhos neste tópico de pesquisa, ainda existem alguns pontos de melhorias para detecção de anomalias em transações eletrônicas. Um desses pontos consiste em filtrar o grande volume de dados gerados nas transações eletrônicas e transforma-los em informações úteis para identificar um comportamento anômalo. A principal tarefa nesse sentido é selecionar um conjunto de atributos que sejam capazes de aumentar a eficácia e reduzir a complexidade dos modelos para detecção de anomalias, tarefa conhecida como seleção de atributos (*Feature Selection*) [Ravisankar et al., 2011].

As técnicas de seleção de atributos são baseadas em um método que avalia a importância de cada atributo, geralmente usando uma medida estatística e define, de acordo com uma regra de decisão, se o atributo é relevante para predição [Liu & Motoda, 1998a]. Assim, a partir de um conjunto $D$ com $n$ atributos, pretende-se encontrar um subconjunto $D'$ com $m|m \leq n$ atributos, que possam descrever os dados e contenham as melhores características para predição [Piramuthu, 2004].

Em geral, as técnicas de seleção de atributos baseiam-se nos valores das classes. Contudo, uma característica marcante dos problemas de detecção de anomalias é o alto desbalanceamento entre as classes, o qual gera um novo desafio para as técnicas de seleção de atributos, que tendem a selecionar os atributos a favor da classe dominante [Kamal et al., 2010].

Assim, neste trabalho é investigado como o desbalanceamento entre as classes afeta as técnicas de seleção de atributos e consequentemente, os modelos para detectar anomalias em transações eletrônicas. Com isso em mente, foram definidas as seguintes hipóteses para esta pesquisa:

1. O alto desbalanceamento entre as classes reduz a eficácia da seleção de atributos

para detectar anomalias em transações eletrônicas.

2. Os métodos tradicionais de seleção de atributos não são adequados para serem utilizados em cenários onde se deseja detectar anomalias.

3. O uso de estratégias de *resampling* antes da seleção de atributos pode melhorar a eficácia da seleção de atributos.

4. Abordagens para seleção de atributos, considerando o alto desbalanceamento entre as classes, aumentam a eficácia dos modelos de detecção de anomalias.

Na Seção 1.1 são traçados os objetivos a serem seguidos para a investigação e validação dessas hipóteses.

## 1.1 Objetivos

Este trabalho visa estudar, caracterizar e propor estratégias eficazes para seleção de atributos na detecção de anomalias em transações eletrônicas. Para isso, podemos descrever os seguintes objetivos específicos:

- Investigar se o desbalanceamento entre as classes diminui a acurácia da seleção de atributos para detecção de fraudes em transações eletrônicas. Para atender este objetivo, foram utilizadas técnicas tradicionais de seleção de atributos, ou seja, que não fazem nenhuma adequação para lidar com o desbalanceamento entre as classes. Essas técnicas foram aplicadas sobre duas diferentes distribuições da mesma base de dados. A primeira mantendo a proporção real entre as classes e a segunda utilizando uma estratégia de *resampling*, medindo, assim, como o desbalanceamento entre as classes afeta a seleção de atributos.

- Avaliar as principais técnicas de seleção de atributos para dados desbalanceados existentes e verificar se essas são eficazes para selecionar atributos para detectar anomalias.

- Construir modelos de detecção de anomalias utilizando estratégias adequadas de seleção de atributos. Esse objetivo foi proposto para confirmar a hipótese 3 desta pesquisa. Para execução desse objetivo, foram utilizados técnicas de seleção de atributos com diferentes estratégias de *resampling* ou técnicas de seleção de atributos com métricas insensíveis ao desbalanceamento entre as classes. Após a seleção de atributos, foram utilizados técnicas de classificação para construir modelos de detecção de anomalias.

- Aplicação dos modelos gerados em bases de dados reais, que apresentam anomalias cuja descoberta represente uma contribuição para garantir segurança, credibilidade e confiabilidade das transações eletrônicas.

## 1.2 Principais Contribuições

Destaca-se como principais contribuições deste trabalho:

1. Um sólido estudo e referencial bibliográfico sobre anomalias em transações eletrônicas;
2. A investigação da eficácia de métodos tradicionais de seleção de atributos na detecção de anomalias;
3. A análise e comparação de diferentes estratégias de *resampling* aplicadas em bases de dados anômalas;
4. A consolidação de métodos de seleção de atributos e de classificação adequados para detectar anomalias;
5. A construção de modelos para detectar anomalias, utilizando técnicas de seleção de atributos adequadas ao desbalanceamento entre as classes e técnicas de classificação;
6. Estudo de casos com dados reais, onde a detecção de anomalias é de extrema importância, como, por exemplo, em sistemas de pagamento eletrônico de grandes corporações de nosso continente.
7. Publicações de artigos em veículos científicos, que serão citados no Capítulo 8.

## 1.3 Organização do Documento

O restante deste documento está organizado da seguinte forma: No Capítulo 2, são descritos alguns trabalhos relacionados aos temas de pesquisa deste projeto e que servirão de base de conhecimento para elaboração deste trabalho. No Capítulo 3, serão descritos os fundamentos conceituais mais importantes para a análise de seleção de atributos em cenários anômalos descrevendo os conceitos fundamentais mais importantes, assim como as técnicas que foram base dos modelos de detecção de anomalias usados neste trabalho. No Capítulo 6, será descrita a metodologia utilizada neste trabalho e que foi aplicada sobre dois distintos cenários reais descritos no Capítulo 7, juntamente com

os resultados experimentais obtidos nesses estudos de casos. O Capítulo 8 descreve as principais conclusões e trabalhos futuros.

# Capítulo 2

# Trabalhos Relacionados

Nesse capítulo são apresentados os trabalhos relacionados com a seleção de atributos para detecção de anomalias em transações eletrônicas. A Seção 2.1 apresenta os trabalhos mais relevantes de detecção de anomalias em transações eletrônicas. Já a Seção 2.2 descreve trabalhos que discutem sobre seleção de atributos para base de dados desbalanceadas, os quais de alguma forma influenciaram a metodologia aqui desenvolvida.

## 2.1 Anomalias em transações eletrônicas

Métodos para detecção de anomalias são bastante discutidos na literatura, com diversas pesquisas nesta área. Em [Chandola et al., 2009] é realizado um estudo bibliográfico sobre os tipos de técnicas e as principais aplicações de detecção de anomalias, discutindo os trabalhos de destaques em cada área. A única forma de detecção de anomalia em transações eletrônicas abordada foi a detecção de fraude em transações de cartão de crédito. Nesse contexto são apresentados trabalhos utilizando Redes Neurais [Aleskerov et al., 1997, Ghosh & Reilly, 1994], *clustering* [Bolton et al., 2001] e baseando-se em regras [Brause et al., 1999].

Inúmeras técnicas são utilizadas em mineração de dados para detectar anomalias em *e-commerce* [Zhang et al., 2013, Kim et al., 2013]. Em [Bhattacharyya et al., 2011] utiliza-se as técnicas de *Support Vector Machines*, *Random Forests* e Regressão Logística na detecção de fraudes em cartões de crédito em um cenário real. Já em [Chiu et al., 2011], foram utilizados métodos de descoberta em redes e mineração de dados na detecção de usuários falsos em um serviço de contas de usuários na *Web*. Em [Almendra, 2013] são utilizadas técnicas de reconhecimento de padrão na detecção de produtos não entregues realizadas por falsos vendedores no sistema de comércio

eletrônico Mercado Livre [1].

Apesar dos trabalhos acimas descritos focarem em diferentes técnicas para detectar anomalias em transações eletrônicas, reforçando a necessidade de estudo nessa área, nenhum desses trabalhos apresentaram foco na seleção de atributos para detecção de anomalias. Os trabalhos descritos até então não utilizaram abordagens de seleção de atributos para se adequar a um cenário anômalo.

Para confirmar essa carência, realizamos uma revisão sistemática da literatura sobre os trabalhos de detecção de fraude, seguindo a metodologia descrita em [Keele, 2007]. Nessa revisão avaliamos os 30 trabalhos mais citados e os 20 trabalhos mais relevantes a partir do ano de 2011, e não identificamos a utilização de estratégias de seleção de atributos adequada para dados desbalanceados. Observamos que apenas 53% dos trabalhos avaliados descrevem a utilização de alguma técnica de seleção de atributos, entretanto, sem tratamento para dados desbalanceados.

Entre os trabalhos analisados, destacamos alguns interessantes sobre detecção de anomalias em transação eletrônicas, mas sem o uso de nenhuma abordagem semelhante com a proposta nesse trabalho. Em [Ahmed et al., 2016] é realizado um survey sobre detecção de fraudes em sistemas financeiros. Os autores focam em algoritmo de agrupamento, mencionando seleção de atributos como uma etapa de pré-processamento importante, mas sem citar a sua relação com dados desbalanceados e estratégias para lidar com esse problema.

O trabalho [Yang & King, 2009] apresenta uma abordagem híbrida, que combina diversas técnicas de classificação para detectar anomalias em uma base de dados de transações eletrônicas providas de um *e-commerce*. Apesar de ser uma base de transações eletrônicas, os autores continham poucos atributos e não utilizaram nenhuma técnica de seleção de atributos. Para validar a metodologia os autores utilizaram as técnicas de classificação convencionais SVM, Redes Neurais e Regressão Logística e propuseram um modelo híbrido composto por combinações dessas.

Os autores em [Arif et al., 2015] fazem uma revisão sobre as técnicas de mineração de dados mais utilizadas para detecção de fraudes em diferentes contextos, citando a seleção de atributos como uma etapa importante para a eficácia dessas técnicas, mas não detalha nenhum método para selecionar atributos para detectar anomalias.

Já em [Pawar et al., 2014], são apresentadas estratégias de detecção de anomalias voltadas para detecção de fraude em cartão de crédito. Os autores realizaram experimento com métodos não supervisionadas para detecção de fraude e PCA para seleção de atributos, mantendo também uma abordagem não supervisionada nessa etapa.

[1] http://www.mercadolivre.com.br

No trabalho de [Ngai et al., 2011a] é apresentada uma revisão literária dos principais algoritmos de mineração de dados para detecção de fraude financeira, conceituando diversos tipos de fraudes. Entretanto, não se cita a seleção de atributos como uma etapa do processo.

Por fim, em [Dal Pozzolo et al., 2014], os autores apresentam o estado da arte em detecção de fraude e dicas práticas importantes para detecção de fraudes em cartão de crédito. Os autores avaliam, sobre um cenário real, diferentes técnicas e estratégias para detecção de fraude, avaliando o impacto de estratégias para dados desbalanceados como *undersampling* na fase de classificação. Entretanto, os autores focam na construção de *features* para detecção de fraude como um processo semântico. No trabalho não é abordado nenhuma técnica de seleção de atributos.

Embora foram apresentadas diversas pesquisas abordando detecção de anomalias em transações eletrônicas, é possível notar que ainda há um grande poder de crescimento nas abordagens até então utilizadas. Os trabalhos aqui descritos mostram que pouco esforço foi feito na avaliação, estudo e utilização de estratégias de seleção de atributos adequadas para detecção de anomalias.

Existem alguns trabalhos que realizaram estudos de seleção de atributos em base de dados desbalanceadas, entretanto nenhum desses trabalhos foca na detecção de anomalias em transações eletrônicas, tema central dessa pesquisa. Alguns poucos trabalhos que propõem ou avaliam estratégias para base de dados desbalanceadas serão apresentados na Seção 2.2.

## 2.2 Seleção de atributos para dados desbalanceados

Embora classificação em base de dados desbalanceadas seja um problema recorrente, com aplicações em diversas áreas, foram encontrados poucos trabalhos que discutem sobre seleção de atributos nesse tipo de cenário. Nesta seção são destacados os mais interessantes encontrados.

Em [Kamal et al., 2010] é realizado um trabalho de seleção de atributos para classes desbalanceadas, mas em um cenário de genes biológicos. Os autores mostram que o problema de selecionar atributos para base de dados desbalanceadas é diferente do problema padrão. No artigo são propostas 3 novas técnicas de seleção de atributos baseadas em técnicas existentes, mudando os pesos dos atributos e a distribuição entre as classes através de *oversampling*. Os autores ao realizem testes sobre a base de dados

de genes biológicos mostraram que as 3 técnicas superaram a técnica de seleção de atributos tradicional *Relief*.

No trabalho desenvolvido em [Chen & Wasikowski, 2008] é apresentado um algoritmo para seleção de atributos para base de dados desbalanceados, denominado *FAST*, que se baseia em uma classificação linear para cada *feature*. Os melhores atributos são escolhidos de acordo com a métrica de avaliação *AUC*, que mede a área abaixo da curva *ROC*. Os autores mostram ganhos do método proposto sobre os métodos de seleção de atributos *Relief* e *CFS*, principalmente ao selecionar poucos atributos.

Já em [Van Hulse et al., 2012], também é realizada uma classificação linear, mas são exploradas diferentes métricas para avaliar um atributo como *F1*, razão de probabilidade, área abaixo da curva de precisão e revocação, média geométrica, informação mutua, etc. Em ambas as técnicas é realizada um deslizamento do limiar para classificação, já que em problemas com desbalanceamento entre as classes, a escolha do limiar padrão, pode não ser uma boa escolha.

Em [Cuaya et al., 2011] é criado um método denominado *FSMC* (Feature Selection to Minority Class), que mede a diferença entre o valor esperado para os atributos da classe majoritária e para os atributos da classe minoritária. O método escolhe os atributos cuja a média dos valores na classe minoritária seja mais distante que a média somada a duas vezes o desvio padrão para os valores das classes majoritária. Para avaliação foram utilizados 5 base de dados médicas provenientes do repositório *UCI ML* [2] e comparado a performance do *FSMC* com 7 técnicas de seleção de atributos tradicionais.

Os autores de [Alibeigi et al., 2012] adotam uma heurística denominada DBFS (*Density Based Feature Selection*), que se baseia na função de densidade de probabilidade. No *DBFS* assume-se que um bom atributo é aquela que tem mínima interseção com as outras classes, ou seja, as instâncias de uma classe não estão espalhadas em outras classes. O método *DBFS* foi comparado a 5 métodos de seleção de atributos baseados em ranking. Para avaliação foram utilizados os classificadores *Naive Bayes*, *1-NN* e *Linear SVM* na classificação dos conjuntos de atributos selecionados. Os autores relatam que os métodos *DBFS* e *FAST* obtiveram os melhores resultados.

O único *survey* encontrado nesse assunto foi [Pant & Srivasta, 2015], que foi publicado como um artigo curto e por isso mostrou-se pouco detalhes sobre o assunto. Nesse artigo, não foi apresentada nenhuma comparação entre os resultados das técnicas. Os autores relatam a existência de apenas 5 métodos para seleção de atributos em base de dados desbalanceadas. No trabalho são apenas replicados os resultados

[2] http://archive.ics.uci.edu/ml/

apresentados nos artigos citados, não é feita nenhuma comparação entre os métodos ou aplicação em outras bases.

Através das pesquisas apresentadas nesta seção é possível perceber a falta de trabalhos que consolidem os métodos apresentados, comparando-os e aplicando em cenários de detecção de anomalias. A maioria dos trabalhos aqui descritos testaram suas abordagens em bases de dados com desbalanceamento entre as classes, mas não em anomalias com alto desbalanceamento, como, por exemplo fraudes em transações eletrônicas. Não foi encontrado nenhum trabalho que realizou análises comparativas entre os diferentes métodos propostos, para que torne algum desses métodos estado-da-arte em seleção de atributos para detecção de anomalias.

# Capítulo 3

# Fundamentos Conceituais

Neste capítulo são introduzidos os conceitos técnicos fundamentais para o desenvolvimento deste trabalho. Na Seção 3.1 é apresentada uma rápida descrição sobre as abordagens de seleção de atributos existentes. Na Seção 3.2 serão apresentadas os principais conceitos sobre *resampling* em dados desbalanceados.

Cabe ressaltar que este trabalho trata de um problema de seleção de atributos para detecção de anomalias, que apresenta como característica marcante o alto desbalanceamento entre as classes, ou seja, em um *dataset* encontra-se um volume muito maior de instâncias de uma determinada classe.

A classe com maior volume de dados será chamada neste trabalho de classe majoritária ou negativa, consequentemente a classe minoritária ou positiva é aquela que apresenta a anomalia, apresentando um volume menor de dados. O primeiro conceito essencial para desenvolvimento deste trabalho é o de seleção de atributos, que é descrito na Seção 3.1

## 3.1 Seleção de Atributos

Seleção de atributos ou *Feature Selection* é uma etapa crucial para descoberta de conhecimento em uma base de dados [Guyon & Elisseeff, 2003]. Intuitivamente, pode se pensar que quanto maior o número de atributos em um conjunto de dados, maior o poder discriminatório e consequentemente maior a acurácia do classificador. Entretanto na prática esse comportamento não é verdadeiro [Koller & Sahami, 1996].

Diversas pesquisas nessa área indicam que um grande número de atributos irrelevantes podem introduzir ruídos nos dados, confundindo o algoritmo de aprendizagem e ocasionando erros na classificação. Além disso, um grande número de atributos desnecessários podem fazer com que os algoritmos de aprendizagem tenham dificuldade em

extrair informações que sejam realmente significantes e relevantes para classificação. [Liu & Motoda, 1998b].

O segundo motivo para a realização da seleção de atributos é a redução da dimensionalidade, já que muitos algoritmos de reconhecimento de padrões sofrem da maldição de dimensionalidade. Essa diz que o número de elementos de treinamento requeridos para que um classificador tenha um bom desempenho é uma função exponencial da dimensão do espaço de atributos [Bittner, 1962].

Visto isso, para a redução do tempo de treinamento e melhora da acurácia em um modelo de classificação, torna-se fundamental a seleção de atributos. Entretanto, essa seleção é um problema de combinação exponencial pelo número de atributos originais em uma base de dados, tornando a missão de encontrar um subconjunto de atributos ótimo um problema NP-difícil [Guyon et al., 2006].

Tipicamente, dado um conjunto de $N$ atributos, a complexidade de busca por um subconjunto ótimo é $O(2^N)$. Assim, surge a necessidade de técnicas, baseadas em métricas heurísticas, para realizar a tarefa de seleção de atributos.

As técnicas de seleção de atributos são baseadas em um método que avalia a importância de cada atributo, geralmente usando uma medida estatística e define, de acordo com uma regra de decisão, se o atributo será selecionado para predição [Liu & Motoda, 1998a]. Assim, a partir de um conjunto $D$ com $n$ atributos, pretende-se encontrar um subconjunto $D'$ com $m|m \leq n$ atributos, que possam descrever os dados e contenham as melhores características para predição [Piramuthu, 2004].

Em geral, um atributo pode ser relevante por dois motivos:

1. Estar fortemente correlacionado com a classe para a qual deseja-se realizar a classificação.

2. Formar um subconjunto com outros atributos que estarão fortemente correlacionados com a classe.

As técnicas de seleção de atributos podem ser divididas em três principais categorias: *wrapper*, *embeded* e *filtro*.

A abordagem ***wrapper*** utiliza algoritmos de classificação no processo de seleção de atributos. Em linhas gerais, nesse tipo de abordagem, aplica-se um processo de treinamento e classificação sobre todos os subconjuntos de atributos disponíveis em uma base de dados ou sobre um subconjunto determinado por alguma heurística.

As principais desvantagens apresentadas pelos métodos *wrapper* são:

- A dependência de um algoritmo de reconhecimento de padrões. Como a abordagem *wrapper* utiliza um algoritmo de treinamento como critério para seleção de

atributos, a solução encontrada por esses métodos não é generalizada. Assim, um subconjunto $s'$ pode ser o melhor para o classificador $A$, mas não ser o melhor para o classificador $B$.

- A complexidade da solução. A abordagem *wrapper* tende a aumentar o tempo de execução do algoritmo de seleção de atributos, pois, além do tempo de avaliação de cada subconjunto, esse tipo de abordagem requer um tempo de treinamento do algoritmo de reconhecimento de padrões que está sendo utilizado.

Devido ao alto tempo computacional necessário, os métodos *wrapper* são geralmente utilizados quando se tem uma quantidade pequena de atributos na base de dados e combinados com técnicas de aprendizado lineares, ou que apresentam baixa complexidade para sua solução.

A abordagem embutida, ou ***embeded***, tem esse nome pois neste tipo de abordagem a etapa de seleção de atributos é embutida no algoritmo de reconhecimento de padrões. Ou seja, nesse tipo de abordagem a etapa de seleção de atributos deixa de ser uma fase de pré processamento e é realizada dinamicamente no processo de indução do classificador.

A abordagem por ***filtro*** não utiliza nenhum algoritmo de classificação para a tarefa de seleção de atributos. Essa abordagem utiliza uma métrica de avaliação do atributo independente do algoritmo de aprendizado, tais como entropia, correlação, informação mútua, chi-quadrado, etc. Assim, são analisadas características gerais dos dados para selecionar um subconjunto de atributos.

A seleção de atributos baseada em filtragem é a abordagem mais utilizada tanto em pesquisas acadêmicas, quanto na indústria [Goswami & Chakrabarti, 2014]. Essa popularidade deve-se principalmente à generalidade, simplicidade e a rapidez da solução, quando comparados com métodos do tipo *wrapper* e *embedded*. Portanto, essa foi a abordagem escolhida neste trabalho.

Para qualquer uma das 3 abordagens, os métodos de seleção de atributos podem trabalhar de duas formas, descritas a seguir:

- **Baseadas em ranking individual**: Nesse tipo utiliza-se um *score*, ou uma métrica de mérito para cada atributo individualmente, construindo um *ranking*. Para selecionar os atributos são escolhidos os top $N$ atributos do *ranking*, podendo $N$ ser encontrado por um número fixo de atributos, alguma porcentagem referente ao total ou sobre alguma métrica de poda.

- **Baseadas em subconjunto**: Nessa categoria, pode-se formular o problema de seleção de atributos como um problema de busca e selecionar o melhor sub-

conjunto de atributos, baseando-se em alguma métrica. Tipicamente, dado um conjunto de $N$ atributos, a complexidade de busca por um subconjunto ótimo é $O(2^N)$, tornando a tarefa de avaliar todos os subconjuntos computacionalmente exaustiva. Assim, geralmente são utilizadas uma escolha gulosa ou alguma heurística de busca para encontrar o melhor subconjunto. A seleção de atributos baseadas em subconjunto, podem ainda ser dos seguintes tipos:

- *Forward*: começa com um subconjunto com apenas um atributo e os atributos são acrescentados incrementalmente, até que se atinja uma pontuação.
- *Backward*: começa com um subconjunto completo e os os atributos vão sendo retirados incrementalmente, até que se atinja um critério de parada.
- *Bidirectional*: nessa abordagem é realizada uma busca adicionando e retirando atributos de um subconjunto.

Da mesma forma que os métodos de aprendizagem, os métodos de seleção de atributos podem ser categorizados como supervisionados, semi supervisionados e não supervisionados.

Os métodos **supervisionados** utilizam informação da classe $C$ para avaliar o mérito de um atributo $a$ ou subconjunto de atributos. Já nos algoritmos de seleção de atributos **semi-supervisionados** são utilizados algumas poucas informações sobre a classe para selecionar os atributos. Por sua vez, as técnicas **não supervisionadas** não utilizam nenhuma informação da classe para realizar a seleção de atributos. Nesse tipo de solução, a seleção de atributos é realizada pela remoção de redundância, ou seja, atributos que são altamente correlacionados entre si podem ser desprezados. Neste trabalho o objetivo é avaliar estratégias de seleção de atributos supervisionadas.

A Figura 3.1 apresenta as categorias de seleção de atributos existente e destaca as que foram utilizadas neste trabalho.

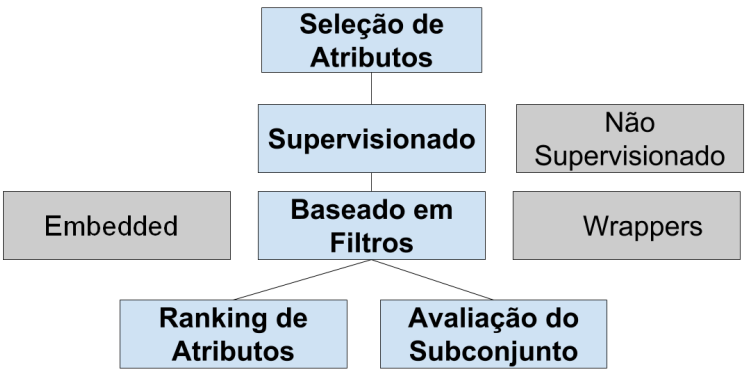


Figura 3.1: Resumo dos principais tipos de técnicas de seleção de atributos.

Conforme descrito anteriormente, as técnicas de seleção de atributos que utilizam a abordagem de ranking, retornam um ranking ordenado pelo mérito de cada atributo. Assim, para selecionar atributos é necessário alguma estratégia para determinar o ponto de corte do ranking. Neste trabalho, decidimos por não fixar o número de atributos e sim utilizar um limiar para corte no ranking.

Um exemplo de gráfico de ranking gerado pelas técnicas de seleção de atributos é mostrado na Figura 3.2. Para determinar o *limiar* de corte utilizamos uma abordagem baseada no joelho da curva, onde essa é calculada com auxílio de uma reta $R$ traçada do primeiro até o último ponto do ranking. Para o cálculo utilizamos os seguintes passos.

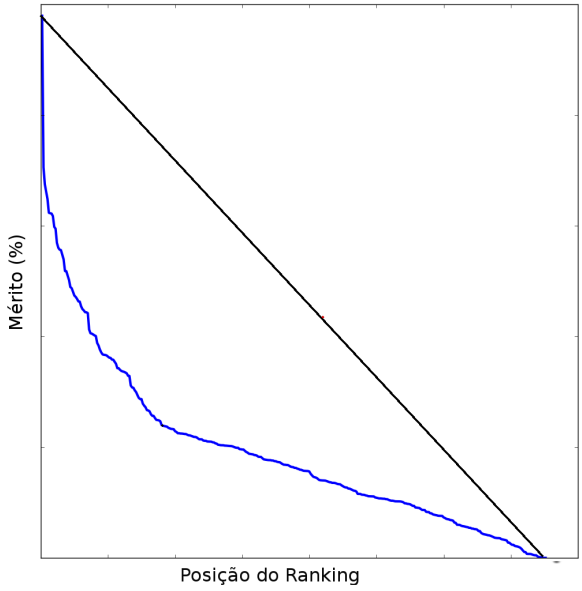


Figura 3.2: Exemplo de Ranking de méritos dos atributos, com uma reta $R$ traçada para cálculo de distância.

1. Traçar uma reta $R$ entre os Pontos $P_1(x_0, y_0)$ e $P_2(x_n, y_n)$
2. Calcular a distância $D_p$ entre os pontos $P_i | P_1 < P_i < P_2$
3. Ponto de corte corresponde ao maior $D_p$ encontrado.

Assim, um atributo $x$ somente será selecionado se o mérito de $x$ for maior ou igual ao valor de $D_p$. Para calcular o mérito de um subconjunto de atributos nos métodos de ranking, utilizamos a média das soma dos méritos dos atributos selecionados.

Na seção 3.2 descrevemos brevemente sobre *resampling* em dados desbalanceados, o qual é mais um conceito fundamental para o desenvolvimento deste trabalho.

## 3.2 Resampling em Dados Desbalanceados

Estratégias de *resampling* são utilizadas para alterar a distribuição entre as classes de um *dataset*. No contexto de classificação em base de dados desbalanceadas, essas estratégias são utilizadas para aumentar a proporção da classe minoritária em relação a classe majoritária. Assim, geralmente, pretende-se diminuir o desbalanceamento entre as classes para então aplicar algoritmos de aprendizado sobre a base de dados, já que o desbalanceamento entre as classes torna as tarefas de aprendizado mais peculiares e difíceis nesses cenários [Chawla, 2005].

Neste trabalho esses métodos serão utilizados para reduzir o desbalanceamento entre as classes, mas não para realizar a classificação, como são geralmente utilizados. Os métodos de *resampling* serão utilizados para reduzir o desbalanceamento antes da etapa de seleção de atributos.

Existem duas abordagens para aumentar a proporção de instâncias de uma classe em relação a outra, denominadas de *undersampling* e *oversampling* [Liu et al., 2009].

- Na estratégia de **Undersampling** são retirados instâncias da classe majoritária ($X_-$) e mantidas as instâncias da classe minoritária ($X_+$). Assim, a proporção $\frac{X_+}{X_-}$ aumenta já que o denominador diminui e o numerador continua o mesmo. O principal risco ao se aplicar estratégias de *undersampling* é retirar instâncias que contenham informações fundamentais para o aprendizado sobre a classe majoritária.

- Já na estratégia de **oversampling**, são replicadas, criadas sinteticamente ou projetadas instâncias para a classe minoritária ($X_+$), mantendo as instâncias da classe minoritária ($X_-$). Assim, também é possível aumentar a proporção de classe minoritária sobre a classe majoritária, já que aumenta-se o numerador e mantém o denominador. A principal desvantagem dessa estratégia é o risco de *overfitting* ao se replicar instâncias. Esse comportamento pode ocorrer caso o modelo se torne complexo e pouco generalizado, apresentando um ótimo desempenho na base de dados de treinamento e eficácia baixa nos testes. No *oversampling* como um padrão é replicado, ele pode ser apresentado para o modelo mais vezes do que os outros padrões, fazendo com o que o modelo se especialize nesse padrão, apresentando assim um melhor desempenho para classificar dados do treinamento e promovendo erros no teste.

De posse dos fundamentos conceituais mais importantes envolvendo esse trabalho, podemos definir as técnicas utilizadas nas duas principais abordagens de seleção

de atributos estudas nesse trabalho. No Capítulo 4 serão descritos os métodos que envolvem a abordagem de seleção de atributos sobre amostras de dados gerados por técnicas de *resampling*. Já no Capítulo 5 serão descritos os métodos de seleção de atributos voltados para dados desbalanceados.

# Capítulo 4

# Estratégia 1 - Seleção de atributos em dados com *resampling*.

Neste capítulo são descritos os métodos implementados para aplicação de *resampling* antes da seleção de atributos. Na Seção 4.1 são descritas 3 técnicas de seleção de atributos sensíveis ao desbalanceamento entre as classes,Além de uma estratégia para combinação dessas técnicas, que será descrita na Seção 4.2. A seleção de atributos foram aplicadas sobre amostras de dados com menor desbalanceamento entre as classes. Essas amostras foram geradas por métodos de *resampling*, que serão apresentados na Seção 4.3.

## 4.1 Técnicas de Seleção de Atributos para dados não desbalanceados

Para avaliar o comportamento da seleção de atributos na detecção de anomalias, escolhemos 3 técnicas de seleção de atributos consideradas estado-da-arte. As 3 técnicas escolhidas são baseadas em filtragem e não fazem nenhuma distinção para lidar com o alto desbalanceamento entre as classes. Antes de apresentar as técnicas é necessário definir o conceito de entropia da informação, que será utilizado nas 3 técnicas apresentadas nas subseções seguintes.

A **entropia** de um espaço, utilizando o conceito de entropia de *Shannon* [Shannon & Weaver, 1949], é caracterizada pelo menor número de *bits* necessários para codificar uma informação. Quanto maior a quantidade de bits necessários para codificar uma informação maior a entropia dessa [Yang & Pedersen, 1997]. Intuitivamente, podemos considerar que uma *string* desorganizada aleatoriamente tem alta entropia e

não pode ser comprimida, enquanto uma *string* ordenada pode ser escrita como uma codificação menor de *strings*.

A definição formal de entropia pode ser aplicada no contexto de classificação, onde a distribuição de instâncias ao longo das classes é tratada como a informação em questão. Assim, se as instâncias são distribuídas ao longo das classes, o número de bits para codificar a informação será alto, porque será necessário enumerar cada instância. Já se todas as instâncias estiverem em uma única classe, a entropia do espaço amostral é baixa, porque com um bit seria possível descrever se todas as instâncias estão na primeira classe ou não.

Assim, podemos dizer que um atributo, ou mesmo um espaço amostral, possui maior entropia se os dados estão espalhados ao longo das classes. A Equação 4.1 define matematicamente a entropia de um espaço amostral $D$ como a impureza dos dados, retornando valores entre 0 e 1, dependendo da homogeneidade dos dados para classificação.

$$H(D) = -\sum_{i=1}^{l}(\frac{n_i}{n})log(\frac{n_i}{n}), \tag{4.1}$$

onde o espaço amostral $D$ tem $n = |D|$ instâncias e $n_i$ membros nas classes $c_i, i = 1, ..., l$. A entropia de qualquer subconjunto $D' \subset D$, incluindo um subconjunto $D'$, pode ser encontrada através da Equação 4.1, utilizando apenas os atributos contidos em $D'$.

As próximas subseções apresentam os métodos tradicionais de seleção de atributos utilizados neste trabalho, que utilizam o conceito de entropia.

### 4.1.1 CFS

O método $CFS$ (*Correlation based Feature subset Selector*) é um método de seleção de atributos supervisionado, baseado em filtragem e com a avaliação de subconjunto [Hall, 2000].

O método avalia a força de um subconjunto de atributos, utilizando o conceito de relevância e redundância. A *relevância* de um atributo é medida pela correlação desse atributo com a classe e a *redundância* é medida pelo correlação desse atributo com os demais atributos do subconjunto. Esses conceitos são intuitivos e utilizam os dois principais objetivos de uma seleção de atributos: escolher atributos com maior poder de predição e diminuir a dimensionalidade retirando atributos com informações redundantes.

Assim, no método *CFS* procura-se escolher os atributos com maior relevância e menor redundância. A implementação pode usar uma busca local *forward*, a qual

somente adiciona um atributo $a$ em um subconjunto $S$ se não existe um atributo $a'$ que pertence a esse conjunto e que tenha maior correlação com a classe $C$ que $a$.

Várias medidas podem ser usadas para calcular a matriz de correlação, dependente da natureza dos dados. Tradicionalmente utiliza-se uma medida de incerteza simétrica, baseadas no conceitos de entropia. A Equação 4.2 define a métrica de informação mútua, dividida pela soma da entropia entre os atributos, geralmente utilizada para cálculo do mérito de um atributo no método *CFS*.

$$SU = 2.0 \times [\frac{H(x) + H(y) - H(x,y)}{H(y) + H(x)}], \tag{4.2}$$

onde $H(x)$ é a entropia de um atributo $x$ em relação ao subconjunto de atributos, $H(y)$ é a entropia da classe e $H(x,y)$ é a entropia de $x$ em relação a $y$.

De posse das correlações, calculada pela Equação 4.2 é utilizada a Equação 4.3 para calcular o mérito de um subconjunto de atributos.

$$Merito(S) = \frac{K \times \overline{r_{ac}}}{\sqrt{K + k(k-1)\overline{r_{aa}}}}, \tag{4.3}$$

onde *Merito(S)* é o mérito de um conjunto $S$ de atributos, $K$ é o número de atributos do subconjunto $S$, $\overline{r_{ac}}$ é a média de correlações entre atributos classes e $\overline{r_{aa}}$ é a média de correlações entre atributos.

A grande vantagem do método *CFS* é que diferentemente da maioria dos métodos baseados em filtragem ele não analisa somente a relevância do atributo, mas também a redundância, diminuindo assim a dimensionalidade do problema. O método *CFS* não faz nenhuma adequação para lidar com dados anômalos, tornando se um método sensível ao desbalanceamento entre as classes.

Essa sensibilidade deve-se à métrica utilizada para calcular a relevância. Tradicionalmente são usadas métricas que não levam em consideração o desbalanceamento dos dados e procuram correlação entre atributo classe, utilizando normalmente a incerteza simétrica mostrada na Equação 4.2.

Nesse sentido, não é ponderado a diferença de probabilidade entre as classes. Um exemplo de insucesso consiste em encontrar um atributo que em determinada amostras consegue identificar a grande maioria das instâncias da classe positiva existente, mas naturalmente tem a maior quantidade de atributos da classe negativa, esse atributo não seria considerado relevante pela métrica adotada. Entretanto, esse atributo poderia ser útil para a detecção da anomalia.

### 4.1.2 GainRatio

A técnica de seleção de atributos *GainRatio* é uma versão ponderada da técnica *Information Gain*, uma generalização da métrica ganho de informação. Assim, para descrever a técnica *GainRatio* utilizada neste trabalho, torna-se necessário uma descrição da técnica *Information Gain*.

O método **Information Gain** é supervisionado e se baseia em filtragem e com a avaliação univariada de atributo baseada em ranking. O mérito de um atributo é calculado através do seu ganho de informação para o modelo, semelhante a métrica de informação mútua, apresentada no método *CFS*, o ganho de informação também baseia-se no conceito de entropia. O ganho de informação de um atributo é determinado pela redução da sua entropia, apresentada matematicamente pela Equação 4.4, que descreve o ganho de informação de um atributo $x$.

$$InfoGain(x) = H(C) - H(C|x), \tag{4.4}$$

onde $H(C)$ é a entropia da classe e $H(C|x)$ é entropia da classe relativa ao atributo $x$. Calculada pela Equação 4.5.

$$H(C|x) = \sum_{j=1}^{m} \frac{(|x_j|}{n} H(C|x = x_j) = p(x, c) \times log(p(c|x), \tag{4.5}$$

onde $H(C|x = x_j)$ é a entropia relativa ao subconjunto de instâncias que tem um valor $x_j$ para o atributo $x$. Se $x$ é um bom descritor para a classe, cada valor de $x$ terá uma baixa entropia distribuído entre as classes, ou seja, cada valor deve estar predominantemente em uma classe.

A técnica **GainRatio** visa solucionar uma limitação da técnica de *Information Gain*, a qual tende a selecionar atributos com maior número de valores distintos. Um exemplo clássico seria um atributo com um identificador sequencial de cada instância. Se esse identificador fosse usado como um atributo, ele teria um ótimo ganho de informação e seria selecionado, uma vez que todas as instâncias com um determinado valor estão na mesma classe.

Para tratar essa deficiência, a técnica *GainRatio* procura selecionar atributos que maximizam o ganho de informação, enquanto miniminizam o número de valores de um atributo. Para isso, divide-se o ganho de informação, calculado na Equação 4.4, pela entropia do atributo. Essa adaptação é apresentada na Equação 4.6.

$$InfoGain(x) = H(C) - H(C|x)/H(x) \tag{4.6}$$

Após o cálculo do mérito de cada atributo pela Equação 4.6, é gerado um ranking e seleciona-se os tops $N$ atributos desse ranking de acordo com algum critério.

A técnica *GainRatio* se mostra sensível ao desbalanceamento entre as classes devido a métrica utilizada em seu corpo. Pela definição $H(c|x) = p(x,c) \times log(p(c|x)$ e derivação de fórmulas, podemos perceber que o ganho de informação é maior quando $P(c|x)$ (probabilidade de de $x$ pertencer a uma classe $c$) é próximo de 1 ou 0. Assim, tende-se a selecionar atributos a favor da classe majoritária, já que a probabilidade é significativamente maior para essa classe.

### 4.1.3 Relief

Da mesma forma que as demais técnicas aqui apresentadas, *Relief* é uma técnica supervisionada baseada em filtragem. Do mesmo modo que a técnica *GainRatio*, nesse método é realizada uma análise univariada de cada atributo em relação a classe. Entretanto, essa técnica não é a aplicação de uma medida direta como a técnica *GainRatio*. *Relief* apresenta um algoritmo um pouco mais elaborado, mostrado no Algoritmo 1, obtido em [Liu & Motoda, 2007].

No algoritmo é calculado, para cada instância, a instância mais próxima da mesma classe ( *nearest hit* $(x_H)$) e a instância mais próxima da classe diferente, *nearest miss* $(x_M)$. A função $diff$ é a responsável por retornar a diferença entre os valores de duas instâncias para cada *feature* $F_i$. Essa é definida de acordo com a Equação 4.7.

**Algoritmo 1:** Relief Feature Selection

**Entrada**: $M$ instâncias $x_k$ descritas por $N$ atributos; parâmetro de amostragem $m$

**Saída**: Para cada atributo $F_i$ um mérito $W| -1 \leq W[i] \geq 1$

**início**
  **para** $i = 1$ *até* $N$ **faça**
    $W[i]$ =0.0;
  **fim**
  **para** $l = 1$ *até* $m$ **faça**
    Escolha aleatoriamente uma instância $x_k$;
    Encontre seu *nearest hit* $(x_H)$ e seu *nearest miss* $(x_M)$;
    **para** *i=1 até* $N$ **faça**
      $W[i] = \frac{W[i]-diff(i,x_k,x_H)}{m} + \frac{W[i]-diff(i,x_k,x_M)}{m}$;
    **fim**
  **fim**
  **retorna** $W$
**fim**

$$diff(i, x_j, x_k) = \begin{cases} \dfrac{|x_{j,i} - x_{k,i}|}{max(F_i) - min(F_i)} & \text{se } F_i \text{ é numérico} \\ 0 & \text{se } x_{j,i} = x_{k,i} \wedge F_i \text{ é nominal} \\ 1 & \text{se } x_{j,i} \neq x_{k,i} \wedge F_i \text{ é nominal} \end{cases} \tag{4.7}$$

Podemos perceber que, ao ser utilizada em dados mistos (numéricos e nominais), a Equação 4.7 favorece atributos nominais, já que o mérito desse atributo é 1 para qualquer diferença. Assim, para solucionar esse problema foi proposta uma equação generalizada que usa um limiar para dados numéricos. Essa é apresentada na equação 4.8.

$$diff(i, x_j, x_k) = \begin{cases} 0 & \text{se } |x_{j,i} - x_{k,i}| \leq t_{eq} \\ 1 & \text{se } |x_{j,i} - x_{k,i}| > t_{diff} \\ \dfrac{|x_{j,i} - x_{k,i}| - t_{eq}}{t_{diff} - t_{eq}} & \text{se } t_{eq} < |x_{j,i} - x_{k,i}| \leq t_{diff}, \end{cases} \tag{4.8}$$

onde $t_{eq}$ é o limiar máximo para a distância entre dois valores de atributos serem consideradas iguais e $t_{diff}$ é o valor mínimo para a distância ser considerada diferente.

Neste trabalho foi utilizada uma versão do *Relief* que não analisa apenas a instância mais próxima para o cálculo de *nearest hit* e *nearest miss*, mas sim as $n$ instâncias mais próximas, calculando a média da contribuição para cada atributo.

O método *Relief*, embora seja mais elaborado, também não faz nenhuma adequação para lidar com dados desbalanceados.Essa inadequação ocorre pois a classe negativa (majoritária) tem uma quantidade muito maior de instância o *nearest hit* tende a ser muito mais alto. Enquanto que para classes positivas (minoritária) o *nearest hit* tende a ser menor. A mesma análise pode ser feita para o *nearest miss*, o qual em classes majoritárias apresentam maior probabilidade de ser próximo de instâncias também da classe majoritária.

Todas as técnicas descritas nesta seção foram utilizadas sobre uma amostra de dados com menor desbalanceamento entre as classes. Essa redução do desbalanceamento foi realiza pelos métodos de *resampling*, que serão apresentados na Seção 4.3.

## 4.2 Merge - Estratégia para combinação de métodos de seleção de atributos

Além dos métodos de seleção de atributos tradicionais apresentados na Seção 4.1, criamos um método que combina os atributos mais frequentes nos melhores conjuntos de seleção. Chamamos esse método de **Merge**. Como exemplo do funcionamento desse método, imaginamos que um atributo $x$ esteja presente em 95% dos melhores conjuntos, ou seja dos conjuntos de atributos que obtiveram os melhores resultados individualmente, o mérito desse atributo seria 0,95.

Assim, para cada atributo será gerado um score e, ordenando esses scores, será possível obter um *ranking* da mesma forma que qualquer técnica individual. De posse desse ranking, utilizamos alguma estratégia de corte no ranking para selecionar um subconjunto de atributos. A Figura 4.1 apresenta um exemplo de *ranking* gerado, onde no eixo y são apresentados os méritos de cada atributo, calculado pela porcentagem de vezes que esse atributo se encontra entre os melhores conjuntos de atributos gerados.

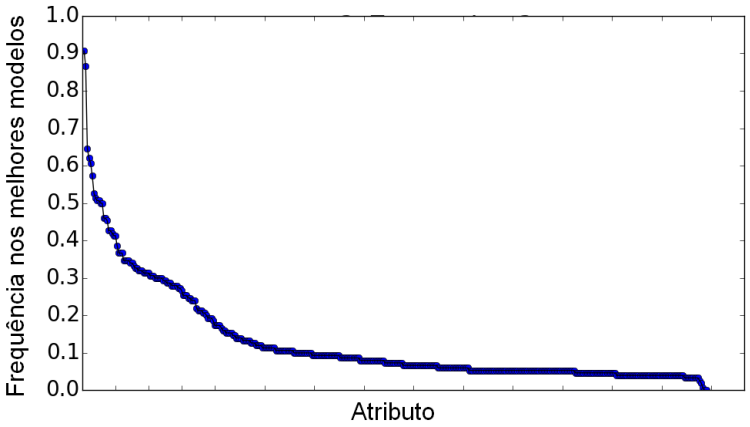


Figura 4.1: Exemplo de ranking gerado pelo método de combinação de atributos Merge.

Um fator crucial para esse método é uma boa avaliação sobre os conjuntos de atributos gerados. Assim, esse método somente pode ser aplicado após a utilização de um método de avaliação dos subconjuntos de atributos gerados pelas técnicas de seleção de atributos.

## 4.3 Métodos de Resampling

Existem diversas técnicas de *resampling* que implementam abordagens de *oversampling* e *undersampling* para diminuir o desbalanceamento entre as classes. Neste trabalho

utilizamos técnicas de *resampling* consideradas estado-da-arte para reduzir o desbalanceamento antes do treinamento para classificação e as avaliamos antes da etapa de seleção de atributos. Essas técnicas são apresentadas na Subseção 4.3.1;

Além disso, criamos uma técnica de *resampling* com conceitos que acreditamos ser importantes para redução do desbalanceamento antes da seleção de atributos, que será apresentada na Subseção 4.3.2.

Para implementação de algumas técnicas de *resampling* é necessário cálculo da distância entre 2 instâncias. Existem diversas métricas para realizar esses cálculos, tanto para dados numéricos, quanto para dados nominais. Entretanto, neste trabalho as base de dados são de natureza mista, ou seja possuem atributos contínuos e nominais. Assim,criamos uma distância par a par mista, denominada *Mixed_Dist* que é apresentada no Algoritmo 2.

**Algoritmo 2:** Mixed_Dist

**Entrada**: Duas instâncias $X$ e $Y$ com $n$ atributos
**Saída**: Distância entre $X$ e $Y$

**início**
  $x_{cont}$ e $x_{cont}$ são os atributos contínuos e nominais da instância $X$
  $y_{cont}$ e $y_{nom}$ são os atributos contínuos e nominais da instância $Y$
  $DC$ = Distância de $Minkowski(x_{cont}, y_{cont})$
  $Media\_DC = \frac{DC}{len(x_{cont})}$
  $DN$ = Distância de $Hamming(x_{nom}$ e $y_{nom})$
  **retorna** $DC + (Media\_DC * DN)$
**fim**

O Algoritmo 2 realiza o cálculo de uma distância par a par única para dados mistos. Para implementação dessa métrica, utilizamos a distância de *Minkowski* para os atributos contínuos e a distância de *Hamming* para atributos nominais.

### 4.3.1 Métodos de Resampling existentes para dados desbalanceados

Todos os métodos apresentados nesta seção são métodos consolidados para redução do desbalanceamento, entre as classes antes do treinamento de uma classificação em dados desbalanceados. Assim, esses métodos foram escolhidos para análise da estratégia de redução do desbalanceamento entre as classes na seleção de atributos para detecção de anomalias.

#### 4.3.1.1 Métodos de Resampling Aleatórios

A principal diferença entre as técnicas de *resampling* é a forma com que elas selecionam as instâncias a serem retiradas ou replicadas. Nos métodos aleatórios não é realizada nenhuma estratégia para essa seleção e nenhuma abordagem inteligente para replicação. Em contrapartida, essas técnicas são as de implementação mais simples e menor tempo computacional. Portanto, são consideradas *baselines* para *resampling* em dados desbalanceados.

As duas principais técnicas de *resampling* aleatórias são descritas a seguir:

- **Random Undersampling** ($RUS$) remove de forma aleatória instâncias da classe majoritária. O único parâmetro dessa técnica é a nova proporção desejada entre as classes, ou de forma mais direta o número de instâncias que deseja-se remover da classe majoritária. Como a escolha das instâncias a ser removida é feita de forma completamente aleatória o risco de se eliminar instâncias com informações fundamentais para classe majoritária é grande. Assim, ao se utilizar a estratégia $RUS$, pode se aumentar o erro na classificação de classes negativas, considerando essa como majoritária [Chawla et al., 2004].

- A técnica **Random Oversampling** ($ROS$) é a implementação do *oversampling* de maneira aleatória. Portanto, nessa técnica são escolhidas de forma aleatória instâncias da classe minoritária para serem replicadas. Conforme o ($RUS$), nesse método o único parâmetro informado é o número de instâncias da classe minoritária que se deseja replicar. A grande desvantagem dessa estratégia é o risco de realizar *overfitting* no modelo de treinamento [Drummond et al., 2003].

As seções seguintes apresentam as técnicas de *resampling*, que visam oferecer melhorias sobre as limitações das técnicas aleatórias.

#### 4.3.1.2 Smote Oversampling

O método Smote (*Synthetic Minority Over-sampling Technique* é um método de *Oversampling* inteligente proposto em [Chawla et al., 2002]. O método foi criado para solucionar um dos principais problemas da abordagem de *Random Oversampling*, o *overfitting* nos dados. Assim, na técnica *Smote*, não simplesmente replica-se as instâncias da classe minoritárias, mas cria-se novas instâncias "sintéticas".

Com a criação de instâncias sintéticas da classe minoritária e manutenção das demais instâncias, aumenta-se a proporção da classe minoritária. As instâncias sintéticas são criadas através de uma interpolação entre os $k$ vizinhos das instâncias minoritárias já existentes.

A criação das instâncias sintéticas no método Smote segue os seguintes passos:

1. Calcula-se a diferença entre o vetor de *features* da instância $x$, e o vetor de *features* da instância mais próxima $x_k$. Para *features* contínuas é utilizada uma medida de distância, já para *features* nominais é repetido o valor da categoria, mais presente entre os vizinhos.

2. Para cada *feature*, multiplica-se essa diferença por uma semente aleatório entre 0 e 1, fazendo com que o método não seja específico e crie instâncias mais generalizadas.

3. Crie a nova instância formada pelos valores sintéticos, composto pela diferença entre $x$ e $x_k$, multiplicadas por uma semente aleatória.

O Algoritmo 3 apresenta um pseudo-código para o método de *resampling* Smote.

O método *Smote* é um método inteligente e considerado estado-da-arte para *oversampling*. Entretanto, esse método ainda apresenta algumas limitações, entre as quais destacam-se:

- Criação de *Features Sintéticas* na base de dados;
- Impacto de algumas variáveis aleatórias, como a escolha aleatória de instâncias para replicar e a semente para cálculo de novas instâncias sintéticas.

Na Subseção 4.3.1.3 serão apresentados 3 métodos distintos de *resampling* para a remoção de instâncias da classe majoritária.

#### 4.3.1.3 KNN NearMiss UnderSampling

Nessa subseção são apresentados os métodos abordados em [Mani & Zhang, 2003] para remoção de instâncias da classe majoritária. A estratégia de *Random Undersampling* seleciona aleatoriamente um subconjunto de instâncias para serem removidas, ocasionando em remoções de *features* que poderiam ser importantes para predição. Visando solucionar esse problema, os autores propuseram uma seleção mais inteligente baseadas nos K vizinhos mais próximos ($KNN$). As 3 estratégias foram denominadas de *NearMiss* 1,2 e 3 e serão apresentadas a seguir:

- O primeiro método, denominado **NearMiss-1** (NM-1), procura remover instâncias da classe negativa que são próximas a outras instâncias da classe positiva. Assim, nesse método seleciona-se as instâncias da classe majoritária que tem a

**Algoritmo 3:** SMOTE

**Entrada**: Número de instâncias da classe minoritária ($T$), Número de vizinhos ($k$), Porcentagem de *resampling*($N\%$

**Saída**: $(N/100) * T$ instâncias sintéticas da classe minoritária

```
1  início
2      K = Número de vizinhos
3      numatribs = Número de atributos
4      Instancias[][] =instâncias originais da classe minoritária
5      novoindice =contador para instância sintéticas geradas
6      Sinteticos[][] =instâncias da classe sintética
7      se N < 100 então
8          Aleatorize as T instâncias da classe minoritária
9          T = (N/100) * T
10         N = 100
11     fim
12     N =(int)(N/100)
13     para i = 1 até T faça
14         nnarray= Índices dos K vizinhos mais próximos
15         Popula(N, i, nnarray)
16     fim
17 fim
18 Função Popula(N, i, nnarray)
19     a = b
20     while N ≠ 0 do
21         nn=número aleatório entre 1 e k
22         for atr = 1 até numatribs do
23             dif = Instancias[nnarray[nn][atr] − Instancias[i][atr]
24             gap =número aleatório entre 0 e 1
25             Sinteticos[novoindice][atr] = Instancias[i][atr] + gap * dif
26         end
27         novoindice + +
28         N = N − 1
29     end
```

menor distancia média $D(x, -)$ para os 3 vizinhos mais próximos da classe minoritária. A ideia dessa técnica é evitar instâncias que sejam muito parecidas com alguma instâncias da classe positiva.

- Já o método **NearMiss-2** (NM-2) visa remover as instâncias que são próximas a muitos exemplos da classe positiva ao mesmo tempo. Assim, ele remove as instâncias da classe majoritária que tem a menor distância média para as três instâncias da classe minoritária mais distante.

- O método **NearMiss-3**(NM-3) utiliza uma abordagem mais simples e seleciona um número $x$ de instâncias da classe negativa para cada instância da classe positiva. Esse método visa garantir que cada instância da classe positiva esteja cercada por algum exemplo da classe negativa.

Todos os métodos *NearMiss undersampling* citados acima, recebem um argumento $N$, referente ao número de instâncias que deseja remover e as remoções são feitas de acordo com a regra de cada método.

A Subseção 4.3.2 apresenta o método de *resampling* que criamos neste trabalho.

### 4.3.2 Método criado: Sampling Outlier - Técnica de Resampling voltada para Seleção de Atributos para Detecção de Anomalia

Nesta seção apresentamos o método de *resampling* **Sampling Outlier**($SO$), criado por nós como uma alternativa específica para redução do desbalanceamento antes na seleção de atributos para detecção de anomalias. Essa técnica utiliza conceitos que acreditamos serem importantes para seleção de atributos em cenários de detecção de anomalias. A principal vantagem dessa técnica é a inexistência do parâmetro que informa a proporção de *resampling* desejada.

O Algoritmo 4 apresenta o pseudo código para técnica *Sampling Outlier*, criada neste trabalho.

O método $SO$ realiza *undersampling* removendo as instâncias mais raras na classe majoritária e *oversampling* replicando as instâncias mais raras da classe minoritária, usando para isso o algoritmo *Smote*. A ideia por trás desse método é que *outliers* são comportamentos diferentes do padrão e que um método de seleção de atributos precisa selecionar atributos que melhor discriminam as classes.

Assim, no método $SO$ assume-se que instâncias raras da classe negativa tem maior chance de serem confundidas com *outliers* e podem atrapalhar a escolha de atributos que poderiam ser úteis. Com esse pensamento são removidas as instâncias mais raras da classe negativa. Em contrapartida, como anomalias são comportamentos raros em uma base de dados, instâncias raras da classe positiva podem ter comportamentos que precisam ser investigados. Logo, são replicadas as instâncias raras da classe positiva.

Além da seleção de instâncias para remover ou replicar, baseadas nos critérios explicados anteriormente, o algoritmo $SO$ combina as duas abordagens de *resampling* existentes *oversampling* e *undersampling* para tentar eliminar as principais desvantagem de cada uma.

**Algoritmo 4:** Método de *resampling* - Sampling Outlier

**Entrada**: $N$ instâncias com $M$ atributos
**Saída**: Conjunto de instâncias replicadas da classe positiva e instâncias mantidas

```
início
    i_pos Instâncias da classe minoritária
    i_neg Instâncias da classe majoritária
    F_pos vetor de tamanho tam(i_pos)
    F_neg vetor de tamanho tam(i_neg)
    para i ∈ i_pos faça
        knn[i][]= Calcule os t KNN de i
        para k ∈ Knn[i][] faça
            F_pos[k]+=1
        fim
    fim
    para j ∈ i_neg faça
        knn[j][]= Calcule os z kNN de j
        para k ∈ Knn[j][] faça
            F_neg[K]+=1
        fim
    fim
    M_F_pos = Média(F_pos)
    Dp_F_pos =Desvio Padrão de (F_pos)
    M_F_neg =Média(F_neg)
    Dp_F_neg =Desvio Padrão de (F_neg)
    Corte_pos = (M_F_pos − Dp_F_pos)
    Corte_neg = (M_F_neg − Dp_F_pos)
    para i ∈ i_pos faça
        se F_pos[i] < Corte_pos então
            Replicar i usando SMOTE
        fim
    fim
    para j ∈ i_neg faça
        se F_neg[j] < Corte_neg então
            Remover j
        fim
    fim
fim
```

O algoritmo 4 mantém um vetor de frequência de vizinhos ($F_{pos}$ e $F_{neg}$) para cada instância $i$ da classe positiva e negativa. Nesse vetor são guardados a quantidade de instâncias $x$ que a instância $i$ faz parte dos vizinhos mais próximos. Assim, procura-se pelas instâncias mais raras, ou seja, aquelas que tem menores valores de $VF$ ou valores

de $VF$ abaixo de um limiar, que calculamos como o valor da média subtraída pelo desvio padrão.

No Capítulo 5 descrevemos os métodos de seleção de atributos insensíveis ao desbalanceamento entre as classes, que consistem na segunda abordagem de seleção de atributos para detecção de anomalias, analisadas neste trabalho.

# Capítulo 5

# Estratégia 2 - Seleção de Atributos para Dados Desbalanceados

Na seção 4.1 do capítulo 4 foram apresentados métodos de seleção de atributos que utilizam métricas sensíveis ao desbalanceamento entre as classes. Neste capítulo apresentamos métodos que tentam lidar com o desbalanceamento entre as classes e poderiam ser eficazes para seleção de atributos na detecção de anomalias.

Na Seção 5.1 são apresentados métodos encontrados na literatura, com algumas adaptações para aplicação neste trabalho. Já na Seção 5.2 é apresentada uma estratégia para otimização multi-objetivo, que foi adequada para combinação de métricas de seleção de atributos.

## 5.1 Métodos Existentes para Dados Desbalanceados

Conforme foi apresentado no Capítulo 2 não existe um método seleção de atributos considerado estado-da-arte para detecção de anomalias. Embora tenhamos encontrado alguns trabalhos com métodos para dados desbalanceados, até então nenhuma dessas foi consolidada como método ideal para selecionar atributos para detecção de anomalias. Assim, os métodos descritos nesta seção foram métodos que obtiveram bons resultados para selecionar atributos em cenários com desbalanceamento entre as classes, mas não foram testados na seleção de atributos para detectar anomalias.

### 5.1.1 Método de Seleção de Atributos Fast

O método *Fast Feature Assessment by Sliding Tresholds* (FAST) foi proposto em [Chen & Wasikowski, 2008] e baseia-se na escolha de atributos pela métrica $AUC$ (*area under the curve*), uma métrica popular que apresenta uma boa forma de avaliação para dados desbalanceados. A $AUC$ é calculada pela área abaixo da curva $ROC$, a qual é gerada através de um gráfico de taxas de falso positivo e taxa de verdadeiro positivo. A métrica $AUC$ é uma das métricas de avaliação que usamos neste trabalho e será apresentada detalhadamente na seção 6.1.2.4.

O método *Fast* realiza uma classificação linear para cada *feature* do conjunto de treinamento, classificando como positivo valores acima de um *tresholds* $\alpha$. Entretanto, em um problema de classificação entre classes desbalanceadas, escolher o *treshold* para classificar uma instância pode ser uma tarefa difícil. Assim, nesse método é realizada a classificação para distintos *treshold* e em seguida calculada a taxa de verdadeiro positivo e falso positivo para cada um desses, sendo possível assim construir uma curva $ROC$ e escolher os atributos que apresentem a maior área abaixo dessa curva ($AUC$).

Entretanto, para se construir uma curva $ROC$ é preciso determinar onde serão feitos os cortes para gerar cada *treshold*. No método *Fast* é utilizado um histograma modificado, denominado pelos autores como uma *even-bin distribution*. Nesse histograma fixa-se o número de pontos em cada *bin*, em seguida calcula-se a média de cada *bin* como *tresholds*.

O Algoritmo 5 apresenta o pseudo código do método *Fast*.

O método *Fast* original foi desenvolvido apenas para dados numéricos contínuos. O método utiliza um *treshold* contínuo para realizar a classificação e a sua utilização em dados categóricos depende de alguma adaptação. Entretanto, neste trabalho iremos trabalhar com base de dados reais, cujos dados são mistos (numéricos e nominais). Como o problema chave dessa pesquisa trata-se de seleção de atributos, decidimos por manter a natureza dos atributos e não realizamos nenhuma transformação de tipos para seleção de atributos.

Assim, todos os métodos aqui apresentados, que não realizarem transformação implícita de tipos de atributos ou não obtiverem versões para lidar com dados mistos serão adaptados. Nesse sentido, para solucionar a limitação do método *Fast* em trabalhar apenas com dados numéricos, criamos uma alternativa para lidar com dados categóricos

Na versão apenas para dados contínuos as classificações lineares são geradas por distintos *tresholds*. A ideia por trás da classificação por *tresholds* é que se existe algum valor de *treshold* que consiga distinguir entre as classes, atingindo boas taxas de $fpr$

**Algoritmo 5:** Fast Feature Selection

**Entrada**: $N$ instâncias com $M$ atributos
**Saída**: Valor de $AUC$ para cada feature

**início**
  $K$ é o número de *bins*;
  $Split = 0$ até $N$ com um passo de tamanho $N/K$;
  **para** $i = 1$ *até* $M$ **faça**
    $X$ é um vetor de instâncias para o atributo $i$;
    Ordene $X$;
    **para** $j = 1$ *ate* $K$ **faça**
      $Inicio = round(Split(j)) + 1$;
      $Fim = round(Split(j + 1))$;
      $MU = media(X(Inicio \text{ até } Fim))$;
      Classifica $X$ usando $MU$ como *treshold*;
      $tpr(i, j) = \frac{tp}{\#positivos}$;
      $fpr(i, j) = \frac{fp}{\#negativos}$;
    **fim**
  **fim**
  Calcula a área abaixo da curva ROC, por $tpr, fpr$;
**fim**

e *tpr* para um determinado atributo, esse pode ser considerado um bom atributo para predição.

A mesma ideia pode ser replicada para um atributo categórico, onde a classificação através do atributo será feita de acordo com uma determinada categoria ou um conjunto de categorias. Assim, um atributo será selecionado se for um bom separador para a classe, ou seja, existirá uma categoria que obtém alto *tpr* e baixo $fpr$, obtendo por consequência alto valor de $AUC$. Visto isso, criamos uma versão do método *Fast*, capaz de trabalhar com dados mistos.

Nessa versão são realizadas classificações sucessivas de acordo com cada categoria do atributo, ou seja, a cada iteração as instâncias com valores de uma categoria do atributo são classificadas como classe positiva. Em seguida, calcula-se o $fpr$ e *tpr* normalmente como no método *Fast*. O Algoritmo 6 apresenta o método *Fast* adaptado neste trabalho para dados mistos.

A Subseção 5.1.2 apresenta o segundo método de seleção de atributos que utiliza métricas insensíveis ao desbalanceamento entre as classes.

**Algoritmo 6:** Adequação do método Fast para aplicação em dados mistos

**Entrada**: $N$ instâncias com $M$ atributos
**Saída**: Valor de $AUC$ para cada atributo

```
início
    para x ∈ M faça
        se x é contínuo então
            Executar pseudo código do método Fast normalmente
        fim
        senão
            cat = Categorias de x;
            i = 0;
            para c ∈ cat faça
                Classifique como positivo todas instâncias com valor c;
                tpr(x, i) = tp / #positivos;
                fpr(x, i) = fp / #negativos;
                i = i + 1;
            fim
            Calcula a área abaixo da curva ROC, por tpr, fpr;
        fim
    fim
fim
```

### 5.1.2 Hell - Seleção de Atributos baseado na Distância de Hellinger

O método **Hell**, proposto em [Yin et al., 2013], é um método de seleção de atributos baseado na distância de *Hellinger*, a qual é uma boa métrica para dados desbalanceados. A distância de *Hellinger* ($DH$) é utilizada para medir a similaridade entre duas distribuições de probabilidade [Cieslak & Chawla, 2008, Kailath, 1967], permitindo obter a noção de afinidade entre medidas de probabilidades, em um espaço finito.

Para realizar o cálculo da distância de *Hellinger*, os atributos contínuos são discretizados em $p$ partições ou *bins*. Em seguida, são calculadas as distâncias sobre as frequências agregadas em todas as partições das classes $X_+$ e $X_-$. A Equação 5.1 apresenta a definição matemática para cálculo da distância de *Hellinger* para um atributo $x$.

$$DH_x = \sqrt{\sum_{j=1}^{p} \left( \sqrt{\frac{|X_+ j|}{X_+}} - \sqrt{\frac{|X_- j|}{X_-}} \right)^2}, \tag{5.1}$$

onde $X_+j$ e $X_-j$, indica a quantidade de atributos com valores contidos na partição $j$

que são da classe positiva e negativa respectivamente.

A distância de *Hellinger* é simétrica, não negativa e retorna valores entre $[0, \sqrt{2}]$. Dado $P$ e $Q$ duas distribuições de probabilidade. Se $P = Q$, então essas tem máxima afinidade e $DH = 0$; se $P$ e $Q$ são completamente disjuntas essas contém zero afinidade e $DH = \sqrt{2}$. Portanto, serão selecionados os atributos que tem menor afinidade entre as classes, ou seja, os atributos que sejam capazes de discriminar melhor entre as classes.

O método *Hell* não foi utilizado originalmente para atributos categóricos. Entretanto, de modo semelhante ao método *Fast*, o método *Hell* pode ser facilmente adaptado para permitir a utilização de dados mistos. Neste trabalho, utilizamos cada partição como uma das possíveis categorias e calculamos a distância *Hellinger* normalmente. O Algoritmo 7, descreve o método *Hell* que foi utilizado neste trabalho com as devidas adaptações.

**Algoritmo 7:** Hell Feature Selection

**Entrada**: $N$ instâncias com $M$ atributos
**Saída**: Distância de Hellinger para cada atributo

**início**
  **para** $x \in M$ **faça**
    $DH_x = 0$
    **se** $x$ *é contínuo* **então**
      Discretize o atributo em $p$ partições
    **fim**
    **senão**
      $p =$ Categorias de $x$;
    **fim**
    **para** $j \in p$ **faça**
      $DH_x = DH_x + \sqrt{\left(\sqrt{\frac{|X_{+j}|}{X_+}} - \sqrt{\frac{|X_{-j}|}{X_-}}\right)^2}$
    **fim**
  **fim**
  **retorna** $DH_x$
**fim**

A próxima seção apresenta um método geral para seleção de atributos e diversas métricas que foram utilizados em conjunto com esse método.

### 5.1.3 Threshold-based Feature Selection (TBFS)

O método $TBFS$, proposto em [Van Hulse et al., 2012], é uma generalização dos métodos de seleção de atributos, baseando-se em uma classificação realizada com cada

atributo individualmente. Assim, nesse método cada atributo é classificado isoladamente e medido o seu desempenho a partir de uma métrica de avaliação.

Conforme os demais métodos para dados desbalanceados apresentados, para realizar a classificação foi necessário utilizar uma solução genérica que permita dados mistos. Assim, para realizar a classificação neste trabalho, utilizamos um procedimento semelhante ao descrito no Algoritmo 7, onde os atributos contínuos foram discretizados em $k$ *bins* e os atributos categóricos foram classificados de acordo com cada uma das suas categorias. O Algoritmo 8 descreve o funcionamento geral do método de seleção de atributos $TBFS$.

Como métricas de avaliação para o método $TBFS$ utilizamos 6 métricas de avaliação para classificação, que segundo alguns trabalhos relacionados ([Forman, 2003, Batuwita & Palade, 2012]), obtiveram bons resultados na avaliação da classificação em dados desbalanceados. Essas métricas são descritas brevemente na Tabela 5.1.

**Algoritmo 8:** Threshold-based feature selection techniques (TBFS)

**Entrada**: $N$ instâncias com $M$ atributos; $Mt$ Métrica de avaliação
**Saída**: Ranking de atributos de acordo com a métrica utilizada

```
1  início
2      para x ∈ M faça
3          se x é contínuo então
4              Discretize o atributo em t partições
5          fim
6          senão
7              t = Categorias de x;
8          fim
9          para j ∈ t faça
10             Classifique x = j como positivo e x ≠ j como negativos.
11             //Avalia a classificação de acordo com a métrica utilizada.
12             TBFS_Mt[j]= Mt(j)
13         fim
14         //Calcule o valor máximo da métrica Mt obtido.
15         Merito(x)= max(TBFS_Mt)
16     fim
17     retorna Merito
18 fim
```

Além dos métodos individuais voltados para dados desbalanceados, propusemos uma solução que combina algumas métricas insensíveis ao desbalanceamento entre as

| Nome | Descrição | Formula |
|---|---|---|
| ACC2 | Acurácia balanceada | $\|tpr - fpr\|$ |
| AGM | Média Geométrica ajustada | $\frac{(Gm+sp*Nn)}{(1+Nn)}$ |
| BNS | Bi-Normal Separation | $\|F^{-1}(tpr) - F^{-1}(fpr)\|$ F é a c.d.f |
| F1 | Média harmônica precisão e revocação | $\frac{2*tp}{(pos+tp+fp)}$ |
| MCC | Coeficiente de correlação de Matthew | $\frac{p*tn-fp*fn}{\sqrt{(tp+fp)(tp+fn)(tn+fp)(tn+fn)}}$ |
| PRC | Área sobre a curva precisão x revocação. | - |

**Notas:**

tp: verdadeiro positivo
fp: falso positivo
fn: falso negativo
pos: número de casos positivos
neg: número de casos negativos
tn: verdadeiro negativo
tpr= $tp/pos$
fpr= $fp/neg$
se= $tp/(tp + fn)$
precisão= $tp/(tp + fp)$
revocação= $tpr$
sp= $tn/(tn + fp)$
gm= $\sqrt{(se * sp)}$
Nn= $neg/pos + neg$

Tabela 5.1: Métricas de seleção de atributos utilizadas para *TBFS*

classes. A solução adotada neste trabalho foi a utilização da *Fronteira de Pareto*, a qual é explicada na Subseção 5.2.

## 5.2 Estratégia Fronteira de Pareto para Seleção de Atributos

A Fronteira de Pareto, também conhecida por *skyline* ou maximal vector [Godfrey et al., 2007], consiste num subconjunto de pontos, tal que nenhum desses pontos seja dominado por qualquer outro. Um ponto em $R^d$ consiste em um vetor de números reais de tamanho $d$. Um ponto domina o outro se ele é melhor em todas as dimensões ou melhor em pelo menos uma e igual nas demais

Neste trabalho será utilizado a fronteira de Pareto para selecionar atributos. Assim, cada atributo será um ponto, formado por diferentes dimensões, as quais são métricas insensíveis ao desbalanceamento entre as classes. Portanto, espera-se encontrar atributos que não sejam dominados por nenhum outro atributo, em outras palavras, espera-se encontrar atributos que tenham um bom desempenho em diferentes métricas de avaliação.

Para encontrar a fronteira de Pareto, utilizaremos a estratégia *Block-nested-loops* (BNL), desenvolvida em [Börzsönyi et al., 2001] e apresentada no Algoritmo 9. Nesse algoritmo é utilizado uma janela (*window*) que a cada iteração armazena os pontos candidatos a fazerem parte da fronteira.

**Algoritmo 9:** Fronteira de Pareto por BNL

```
1  window=0 para todo ponto r ∈ R faça
2      naoDominado = True
3      para todo ponto w ∈ window faça
4          se r domina w então
5              remova w de window
6          fim
7          senão
8              se r é dominado por w então
9                  remova w de window
10                 Insira w no inicio de window
11                 naoDominado = False
12                 break
13             fim
14         fim
15     fim
16     se naoDominado então
17         Insira r no fim de window
18     fim
19 fim
20 retorna window
```

No Capítulo 6 apresentamos a Metodologia de avaliação das duas abordagens de seleção de atributos propostas neste trabalho.

# Capítulo 6

# Metodologia Experimental

Neste capítulo apresentamos os métodos de avaliação e a metodologia experimental utilizada para avaliação das abordagens de seleção de atributos para detecção de anomalias em transações eletrônicas. É importante ressaltar que a metodologia aqui apresentada é genérica e pode ser usada para diferentes estudos de caso.

Na Seção 6.1 descrevemos os métodos e métricas utilizados para avaliação dos subconjuntos de atributos gerados pelos métodos apresentados na duas estratégias de seleção de atributos.

## 6.1 Métodos de Avaliação

Toda a avaliação sobre os subconjuntos de atributos gerados é realizada através de uma classificação utilizando o subconjunto. A Seção 6.1.1 descreve os métodos de classificação utilizados para essa avaliação, para avaliar essa classificação são utilizadas métricas de avaliação adequadas para cenários de desbalanceamento entre as classes, tais métricas são descritas na Seção 6.1.2.

### 6.1.1 Técnicas de Classificação

Nesta seção abordaremos as técnicas de classificação que serão utilizadas para comparar a eficiência das técnicas de seleção de atributos e identificar as anomalias.

As técnicas de classificação detectam anomalias através do reconhecimento de padrões, onde um padrão pode ser considerado como um conjunto de características semelhantes. Uma definição prática para padrões pode ser dada a partir da descrição de um problema recorrente para o qual existe uma solução que pode ser reutilizada diversas vezes em situações diferentes [Cerqueira, 2010].

Cabe ressaltar que o foco deste trabalho não é a etapa de classificação e sim a etapa de seleção de atributos. Portanto, foram utilizadas técnicas de classificação consideradas estado-da-arte na literatura e não foram testadas técnicas específicas mais elaboradas para detecção de anomalias. As técnicas de classificação utilizadas foram Regressão Logística, Redes Bayesianas de classificação e Árvore de Decisão, as quais são descritas brevemente a seguir.

- **Redes Bayesianas** de classificação são grafos acíclicos dirigidos que representam dependência entre as variáveis de um modelo probabilístico, onde os nós são os atributos e os arcos representam os relacionamentos de influência entre as variáveis. De posse do grafo, é possível determinar um conjunto de variáveis dependentes e usando o teorema de Bayes gerar uma tabela de probabilidade condicional. Em seguida, podemos encontrar a probabilidade de um evento a partir da tabela de probabilidade condicional [Maes et al., 2001].

- **Regressão Logística** é uma técnica estatística que produz, a partir de um conjunto de variáveis explicativas, um modelo que permite a predição de valores dados por uma variável dependente categórica. Assim, usando um modelo de regressão logística, é possível calcular a probabilidade de um evento através de uma função de ligação [Hosmer, 2000]. Neste trabalho foi utilizada um modelo de regressão logístico binário, sendo esse um caso especial de um modelo linear generalizado com a função de ligação *logit* [Dobson, 1990].

- **Árvore de Decisão** gera um modelo representado por uma árvore, essa consiste de folhas, indicando atributos, nós de decisão especificando um teste nos valores de cada atributo e um ramo para cada resposta possível, o qual conduzirá para uma nova árvore ou uma nova folha [Salzberg, 1994]. Neste trabalho utilizamos o algoritmo de Árvore de Decisão $C.45$, construindo as árvores através de um conjunto de dados de treinamento e usando o conceito de entropia da informação [Quinlan, 1993].

Além das técnicas citadas, realizamos alguns experimentos com a técnica de classificação SVM *(Support Vector Machines)* com kernel RBF (*Radial Basis Function*). Entretanto o *SVM*, que foi citado em muitos trabalhos como uma das técnicas de classificação estado-da-arte para detecção de fraude [Bhattacharyya et al., 2011], apresentou tempo computacional para aprendizado bem maior que as demais técnicas e resultados similares. Assim, decidiu-se não adotar essa técnica neste trabalho. Os parâmetros para as técnicas de classificação foram estimados através de sugestões da literatura e testes empíricos.

### 6.1.2 Métricas de Avaliação

Existem diversas métricas que podem ser utilizadas para avaliar a eficácia de um modelo de classificação. Entretanto, embora detectar anomalias se trate de um problema de classificação, a avaliação nesse tipo de cenário não pode ser feita da mesma maneira, já que algumas métricas são sensíveis ao desbalanceamento entre as classes.

Assim, surge a necessidade de utilização de métricas de avaliação adequadas para cenários com alto desbalanceamento. As métricas padrões e as adequadas para detecção de anomalias utilizadas neste trabalho serão descritas nas próximas subseções. Em comum, a todas essas técnicas temos o conceito de matriz de confusão que é descrita a seguir.

#### 6.1.2.1 Matriz de Confusão

A performance dos algoritmos de aprendizado de máquina são geralmente avaliadas em métricas, baseada por uma matriz de confusão, que é ilustrada na Figura 6.1, onde as colunas são a classe predita e as linhas são a classe real.

| | Valor Predito NEGATIVO | Valor Predito POSITIVO |
|---|---|---|
| Valor Real NEGATIVO | TN | FP |
| Valor Real POSITIVO | FN | TP |

Figura 6.1: Matriz de confusão.

A partir da matriz de confusão é possível obter as seguintes métricas:

- **True positive (TP)**: Medida de acerto nos testes que indicam o número de instâncias positivas que foram corretamente identificadas. Neste trabalho significa instâncias que o modelo classificou como anomalias e realmente eram anomalias.
- **False Positive (FP)**: Medida de erro nos testes que representa que o modelo previu erroneamente uma instância como positiva. Aplicadas neste trabalho, significa que o modelo classificou uma instância como anomalia, mas na verdade era uma instância de padrão normal.
- **True Negative (TN)**: Medida de acerto nos testes que retorna o número de instâncias negativas que foram corretamente classificadas. Aplicadas neste tra-

balho, indica o número de instâncias do padrão normal que foram corretamente classificadas como padrão normal.

- **False Negative (FN)**: Medida de erro no teste que retorna o número de instâncias classificadas como negativa, mas que na verdade eram positivas. Ou seja, aplicadas neste trabalho corresponde ao número de instâncias classificadas como padrão normal, mas que na verdade eram anomalias.

#### 6.1.2.2 Precisão e Revocação

A precisão e revocação são métricas que podem ser calculadas separadamente para cada classe, ou gerar uma métrica única para um modelo (macro). Entretanto, para o problema de detecção de anomalia, o alto desbalanceamento entre as classes faz com que a macro precisão não seja uma boa alternativa. Por exemplo, dado um cenário com 100 instâncias, sendo 95 instâncias da classe negativa e apenas 5 da classe positiva. Se aplicarmos um método de detecção de anomalias que não detecta nenhuma anomalia e classificássemos todas as instâncias como negativa, ao medirmos a precisão desse método encontraríamos uma precisão de 95%. Essa precisão parecia ser uma boa precisão para um modelo de classificação, entretanto, o método não conseguiu detectar nenhuma anomalia.

Assim, nesse cenário torna-se mais interessante utilizar a precisão e revocação para classes separadamente. Precisão e Revocação podem ser definidas para classe positiva, em função de $TP$, $FP$ e $FN$, explicados na seção 6.1.2.1, de acordo com as equações 6.1 e 6.2.

$$\text{Precisão} = TP/(TP + FP) \tag{6.1}$$

$$\text{Revocação} = TP/(TP + FN) \tag{6.2}$$

A partir das equações 6.1 e 6.2, percebemos que a precisão para classe positiva é o numero de instâncias classificadas corretamente como positivas, sobre o total de instâncias classificadas como positivas. Já revocação é o numero de instâncias classificadas corretamente como positivas sobre o total de instâncias positivas existentes.

Um modelo de aprendizado perfeito apresentaria uma revocação e precisão de 100%. Entretanto, aumentar a revocação mantendo uma boa precisão não é tarefa simples. Assim, torna-se necessário métricas que permitam realizar a análise da taxa de precisão e revocação simultaneamente. Uma dessas métricas é o *F1-Score* ou *F-Measure*, apresentada na Subseção 6.1.2.3.

#### 6.1.2.3 F1 Score

O grande interesse para um cenário de detecção de anomalias é obter uma boa revocação para classes positivas, sem diminuir a precisão da classe negativa. Ou seja, encontrar o maior número de anomalias sem elevar a taxa de falso positivo. A métrica **F1** ou *F-Measure* é uma das métricas que procuram combinar esse *trade-off* entre precisão e revocação, resultando em um único valor, que reflete a eficácia de um modelo na presença de classes raras [Chawla, 2005].

A métrica $F1$ retorna um valor entre 0 e 1 , referente a média harmônica entre precisão e revocação para cada classe, conforme mostrado na equação 6.3.

$$F1 = 2 * \frac{\text{precisão} \times \text{revocação}}{\text{precisão} + \text{revocação}} \tag{6.3}$$

Neste trabalho, é realizado uma classificação binária. Assim, obtemos um valor de $F1$ para classe positiva ($F1\_pos$) e um valor de $F1$ para classe negativa ($F1\_neg$). Para facilitar o comparativo entre os resultados é calculado a média aritmética entre os valores de $F1$ para cada classe, denominando essa medida de $AVG\_F1$, apresentado na equação 6.4.

$$AVG\_F1 = \frac{F1_{pos} + F1_{neg}}{2} \tag{6.4}$$

Desse ponto em diante sempre que falarmos da métrica $F1$ estaremos adotando o $F1$ médio ($AVG\_F1$).

#### 6.1.2.4 AUC

A curva $ROC$ (*receiver operating characteristic*) é considerada uma técnica padrão para sumarizar a eficácia de um classificador binário. Ela é calculada sobre uma faixa de *tresholds*, que corresponde a um valor na escala contínua de predição que discrimina entre essas duas classes, utilizando as medidas de *true positive* e *false positivo*. Mais precisamente, para gerar a curva cria-se um gráfico que mostra no eixo x a taxa de *false positive* ($FPR$), também chamada de especifidade e no eixo y a taxa de *true positive* ($TPR$), conhecida como revocação ou sensibilidade.

A área abaixo da curva ROC (***AUC***) é uma métrica muito utilizada para comparações entre modelos. Essa pode ser interpretada como a probabilidade que um classificador irá obter um alto *score*, para uma escolha aleatória de *treshold*. Como a $AUC$ é uma porção da área do espaço $ROC$, seus valores variam entre 0 a 1. Entretanto como classificadores piores que os aleatórios não são encontrados no espaço $ROC$, não existem classificadores com $AUC$ menor que 0,5. Assim, os valores de $AUC$ variam

entre $0,5$ e 1, onde um modelo aleatório obteria valores de $AUC$ próximos de 0,5 e os melhores modelos próximos de 1.

Uma das vantagens da utilização dessa métrica para avaliar modelos de detecção de anomalias, é que a métrica $AUC$ é independente do critério de decisão (*treshold*) para classificação e das probabilidades a priori entre as classes. A Figura 6.2 apresenta um exemplo de gráfico de curva $ROC$, para a explicação dos principais aspectos aqui apresentados.

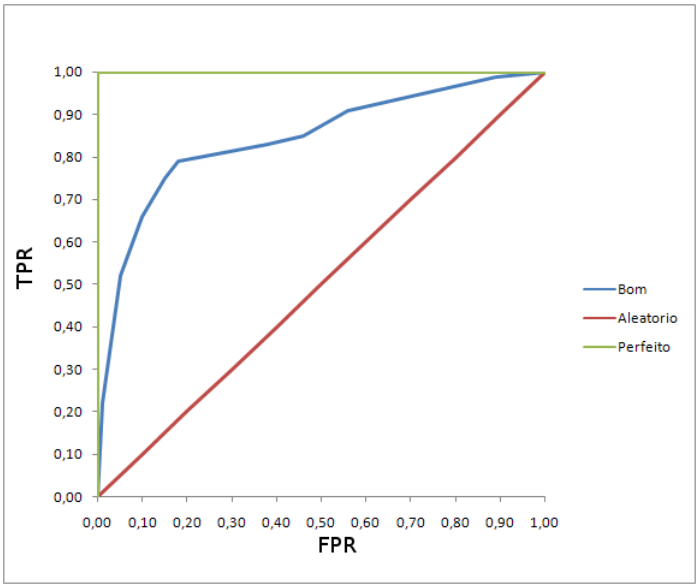


Figura 6.2: Curva ROC principais conceitos

#### 6.1.2.5 Eficiência Econômica

Um dos cenários clássicos de detecção de anomalias é a detecção de fraude. Neste trabalho foram realizados dois estudos de caso para detecção de fraudes em transações eletrônicas. Em cenários como estes, a principal meta de um modelo de detecção de fraudes é a redução do prejuízo financeiro ocasionado por essas. Em transações que envolvem valores financeiros o custo de um falso positivo e um falso negativo não é o mesmo, estima-se que esse custo na grande parte das companhias é da ordem de 1:100, ou seja, se perde bem mais prevendo que uma operação fraudulenta é legal, do que o contrário.

Portanto, embora se trate de um problema de classificação, a análise do número de instâncias classificadas corretamente não é suficiente, já que o prejuízo acarretado pelas fraudes é um fator crucial nesses cenários. Assim, criamos uma métrica denominada **Eficiência Econômica** ($EE$) para avaliar o possível ganho financeiro de uma

companhia após a aplicação dos modelos [Lima & Pereira, 2012]. A métrica não é sensível ao desbalanceamento entre as classes e retorna uma avaliação financeira real de quão poderoso seriam os modelos para detectar fraudes. A Equação 6.5 apresenta a definição matemática da Eficiência Econômica.

$$EE = k \cdot TN_{Value} - ((1-k) \cdot FN_{Value} + p \cdot FP_{Value}), \tag{6.5}$$

onde $k$ é uma constante que representa o percentual que a empresa recebe em cada transação; $p$ é a penalização sobre um falso positivo; $TN_{Value}$, $FN_{Value}$ e $FP_{Value}$ é a soma dos valores das transações que foram respectivamente *true negative*, *false negative* e *false positive*.

Para facilitar a comparação entre os modelos adotamos a Eficiência Econômica Relativa, que utilizam os seguintes conceitos:

- Eficiência Econômica Máxima ($EE_{Max}$), a qual determina o maior valor que poderia ser arrecadado pela companhia. A $EE_{Max}$ ocorreria em um modelo ideal com 100% de *True Positive* e 0 de *False Positive*.
- Eficiência Econômica Real ($EE_{Real}$) é a eficiência obtida pelo mercado eletrônico na atualidade com seus métodos para detecção de fraudes.

A Equação 6.6 apresenta a $EE\_Relativa$ que será utilizada neste trabalho.

$$EE\_Relativa = \frac{EE - EE_{Real}}{EE_{Max} - EE_{Real}} \tag{6.6}$$

Desse ponto em diante sempre que falarmos em Eficiência Econômica ($EE$) estaremos nos referindo a Equação 6.6.

#### 6.1.2.6 Testes Estatísticos

Para todas essas métricas utilizamos os testes estatísticos de *Friedman* e *Wilcoxon*. Esses dois testes foram escolhidos porque são testes não paramétricos. Uma vantagem dos testes não paramétricos é a não necessidade de uma distribuição normal [Demšar, 2006]. Como neste trabalho as base de dados apresentam grande desbalanceamento entre as classes, optamos pela utilização de testes não paramétricos.

O teste de **Friedman** é um teste estatístico não paramétrico usado para detectar diferença estatística em múltiplos testes, similar ao teste paramétrico ANOVA. O teste trabalha com as hipóteses $H0$, a qual diz que não há diferença entre os tratamentos e $H1$, que afirma que pelo menos um tratamento é diferente. Quando a hipótese nula $H_0$

é rejeitada, temos que ao menos um dos grupos é diferente dos demais. Neste trabalho $H0$ é rejeitada com 95% de confiança se *p-value* é menor que 0.05 [Friedman, 1940]. Entretanto, utilizando o teste de *Friedman* padrão não temos a informação de quais grupos são diferentes, assim surge a necessidade de utilizar alguma abordagem par a par.

Para uma comparação par a par utilizamos uma versão do teste estatístico de *Wilcoxon*. O teste de **Wilcoxon** é um método não-paramétrico para comparação de duas amostras, substituindo a versão paramétrica do teste $t$ de *Student* para amostras pareadas. No teste de *Wilcoxon* se aceitarmos a hipótese nula temos que a mediana da diferença entre duas amostras é nula, ou seja, as populações não diferem. Já se a hipótese nula for rejeitada, ou seja, se a mediana da diferença não for nula, temos que as populações diferem. Na versão par a par do teste de *Wilcoxon*, é necessário realizar um ajuste no *p-value*. Neste trabalho, esse ajuste foi realizado pelo método de *Benjamini & Hochberg* [Benjamini & Hochberg, 1995].

A Figura 6.3 apresenta um resumo do principais métodos utilizados nesse trabalho, em cada uma das estratégias de seleção de atributos, apresentadas nos Capítulos 4 e 5, avaliados pelos métodos de avaliação que foram apresentados nesta seção.

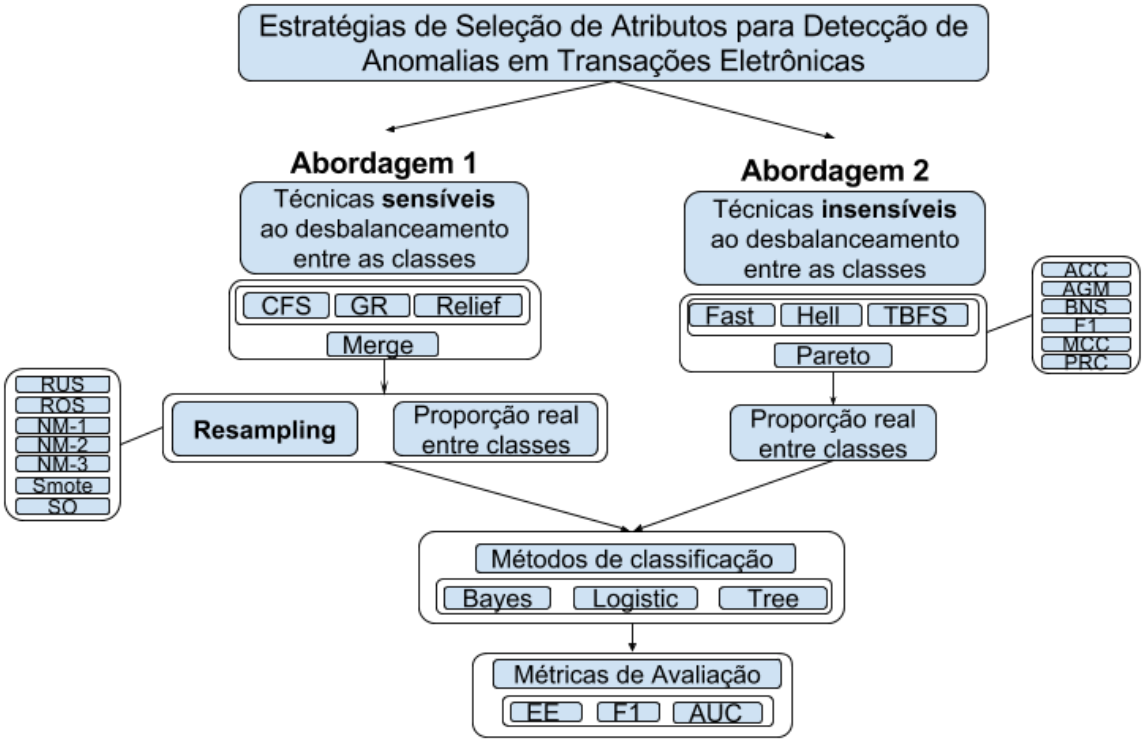


Figura 6.3: Metódos utilizados pelas estratégias de seleção de atributos para detectar anomalias.

A Seção 6.2 descreve a metodologia para utilização dos métodos descritos na Figura 6.3 duas principais abordagens de seleção de atributos desenvolvida neste trabalho.

## 6.2 Metodologia Experimental

A Figura 6.4 apresenta, de forma resumida, o processo de descoberta de conhecimento (*Knowledge Discovery in Databases* - KDD) utilizado neste trabalho para investigação das hipóteses de pesquisa. O principal diferencial deste trabalho está nas etapas 2 e 3, onde são investigadas e criadas estratégias interessantes de seleção de atributos para detecção de anomalias.

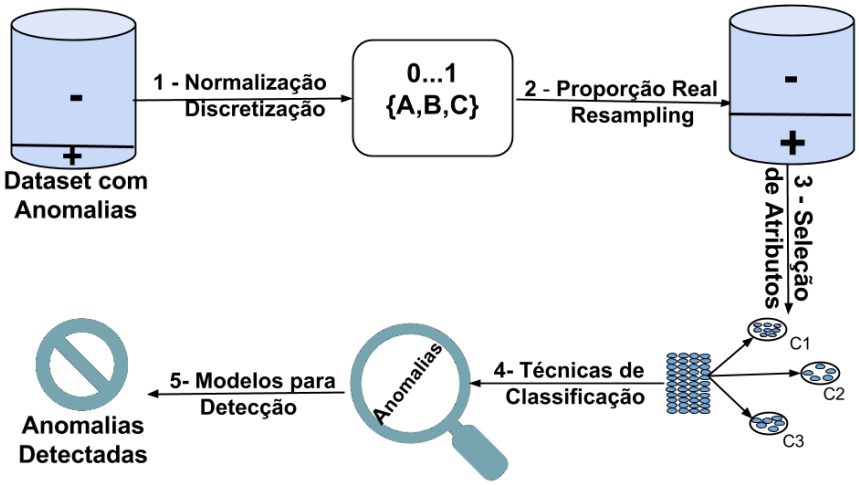


Figura 6.4: Metodologia para construção do modelo de detecção de anomalias.

O **passo inicial** do trabalho é a obtenção de uma **base de dados rotulada com anomalias**. A característica marcante para um problema de detecção de anomalias é o alto desbalanceamento entre as classes. Nesse cenário, a classe majoritária (não fraude) é representada como negativa e apresenta um número muito maior de instâncias do que a classe minoritária (fraude), a qual é representada como positivo.

De posse da base de dados, é necessário uma sistemática etapa de **caracterização e pré-processamento**. Como as técnicas de aprendizagem lidam diretamente com um conjunto de dados, é importante entender e conhecer a natureza de cada atributo. A natureza de um atributo irá influenciar diretamente no tratamento a ele dado.

Os atributos podem ser de natureza qualitativa ou quantitativa. Os atributos qualitativos podem ser divididos em categóricos nominais e categóricos ordinais. Nos nominais não existem ordenações nas possíveis categorias, enquanto nos ordinais os

valores são discriminados em uma determinada ordem. Já os atributos quantitativos representam valores numéricos que podem ser classificados em discretos, assumindo valores dentro de um conjunto de números específicos ou atributos contínuos,assumindo valores em um intervalo contínuo de números.

Após a obtenção de um conjunto de dados (*dataset*) rotulado e o entendimento da natureza de cada atributo, podemos realizar um conjunto de passos que permitirão validar as hipóteses desta pesquisa.

A **primeira etapa** da metodologia deste trabalho é a **normalização da base de dados**, ajustando os atributos contínuos para uma escala real entre 0 e 1 e os atributos discretos como categorias não ordinais. Após essa etapa de pré-processamento, separamos um subconjunto de dados de treinamento chamados de **dados para seleção**, que serão utilizados para selecionar atributos, por todas as técnicas de seleção de atributos. Esse conjunto de dados para seleção é formado pelos mesmos atributos, mas com instâncias completamente diferentes dos dados utilizados para treinamento e validação dos modelos, garantindo, assim, a generalidade da seleção.

Na etapa 2, iniciamos os procedimentos para a verificação da nossa primeira hipótese de pesquisa. Para verificar se o alto desbalanceamento entre as classes reduz a eficácia da seleção de atributos na detecção de anomalias em transações eletrônicas, **utilizamos duas abordagens de distribuição entre as classes, antes da seleção de atributos**. A primeira abordagem utiliza o conjunto de dados para seleção com a **distribuição real** entre as classes para selecionar atributos, já a segunda abordagem utiliza diferentes estratégias de ***resampling*** sobre os dados de seleção para aumentar a proporção entre as classes.

A intenção dessa etapa é verificar se, ao mudar a proporção entre as classes, os atributos selecionados serão diferentes. Mais precisamente, através dessa abordagem, podemos verificar se, ao aumentar a proporção da classe positiva (anomalia), será possível obter um subconjunto de atributos melhor para detectar anomalias que mantendo a proporção real. Para realizar o *resampling*, **utilizamos 6 técnicas consideradas estado-da-arte em *resampling* para classificação em dados desbalanceados** e **criamos uma técnica de *resampling*** denominada *Sampling Outlier*, com conceitos que consideramos serem importantes na seleção de atributos para detectar anomalias. Todas essas 7 técnicas foram apresentadas na Seção 3.2.

Para confirmar a hipótese, bastaria a utilização de um dos métodos de *resampling*, mas um dos diferencias deste trabalho é a **comparação das estratégias de *resampling* antes da seleção de atributos**. Assim, com a utilização dessas estratégias de *resampling* antes da seleção de atributos, será possível não somente confirmar a primeira hipótese da pesquisa, como construir um referencial sobre as peculiaridades,

paradigmas e melhores estratégias para aplicação de técnicas de *resampling* antes da seleção de atributos.

Em comum, as técnicas de *resampling*, com exceção da técnica que criamos, recebem um parâmetro que informa a proporção da classe minoritária que deseja-se obter após o *resampling*. Determinar qual a melhor distribuição entre as classes é uma tarefa difícil. Neste trabalho **criamos diversos subconjuntos variando a proporção entre as classes** em 5%, 10%, 15%, 20%, 25%, 30%, 40% e 50%. Essa variação é importante para encontrar a melhor distribuição para cada técnica e verificar comportamentos gerais. Um desses comportamentos é a hipótese de que quanto maior o balanceamento melhor deve ser a performance das técnicas de seleção de atributos.

Assim, para 6 das 7 técnicas de *resampling* utilizadas neste trabalho foram criados 8 subconjuntos, variando a distribuição entre as classes. Como a técnica *SO* não é dependente da escolha da distribuição, essa gerou apenas um subconjunto. Utilizando a abordagem de *resampling* antes da seleção de atributos, obtemos 50 conjuntos distintos para seleção, cuja a distribuição é apresentada na Tabela 6.1.

| **Resampling** | **Distribuição Utilizada (%)** | **# Subconjuntos** |
|---|---|---|
| NM_1 | 5:30,40,50 | 8 |
| NM_2 | 5:30,40,50 | 8 |
| NM_3 | 5:30,40,50 | 8 |
| ROS | 5:30,40,50 | 8 |
| RUS | 5:30,40,50 | 8 |
| Smote | 5:30,40,50 | 8 |
| SO | X | 1 |
| Real | X | 1 |

Tabela 6.1: Subconjuntos para seleção de atributos gerados pelas técnicas de *resampling*.

De posse dos subconjuntos para seleção, gerados pelas técnicas de *resampling* e pelo subconjunto com a distribuição real, podemos iniciar a **etapa 3**. Nessa etapa **realizamos a seleção de atributos através de duas abordagens diferentes**. A primeira compreende a **aplicação de 3 técnicas de seleção de atributos consideradas estado-da-arte, mas que não levam em consideração o desbalanceamento entre as classes**, sob os diferentes dados de seleção. As técnicas utilizadas foram *CFS, GainRatio, Relief*, descritas na Seção 4.1. Além da aplicação do método *Merge*, para combinação dessas técnicas, descrito na Seção 4.2.

Já na segunda abordagem, **aplicamos 7 técnicas de seleção de atributos que utilizam métricas insensíveis ao desbalanceamento** entre as classes sobre a o conjunto de dados para seleção com proporção real. Além das técnicas individuais,

utilizamos a estratégia de Fronteira de Pareto para combinar as melhores métricas. Todas essas técnicas insensíveis ao desbalanceamento entre as classes foram descritas no Capítulo 5.1 .

Através da primeira abordagem, pretendemos verificar a hipótese de que os métodos tradicionais de seleção de atributos não são adequados para serem utilizados em cenários onde visa-se detectar anomalias e que estratégias de *resampling* podem melhorar a eficácia da seleção de atributos. Para confirmar essas hipóteses, comparamos a seleção de atributos realizada sobre os conjuntos de dados com *resampling* para diminuir o desbalanceamento entre as classes e o conjunto com a distribuição real.

Neste trabalho não utilizamos um número fixo de atributos. A escolha pelo número de atributos foi feita a partir de um limiar, que foi explicada na Seção 3.1, fazendo com que cada técnica gere números diferentes de atributos para cada subconjunto de seleção.

Portanto, cada uma das 3 técnicas seleção de atributos gera um subconjunto diferente em cada um dos 50 subconjuntos para seleção apresentados na Seção 6.1. Além disso, criamos um conjunto sem nenhuma técnica de seleção de atributos *No_FS* e criamos conjuntos a partir da abordagem *Merge* explicada anteriormente. Assim, temos um total de 152 subconjuntos de atributos, resumidos da seguinte maneira:

- 50 subconjuntos de atributos selecionados utilizando a técnica **CFS**, sobre cada um dos dados de seleção apresentados na Tabela 6.1.
- 50 subconjuntos de atributos selecionados utilizando a técnica **GainRatio**, sobre cada um dos dados de seleção apresentados na Tabela 6.1.
- 50 subconjuntos de atributos selecionados utilizando a técnica **Relief**, sobre cada um dos dados de seleção apresentados na Tabela 6.1.
- 1 subconjunto de acordo com o método **Merge**, gerado pela combinação dos atributos mais frequentes nos melhores modelos acima.
- 1 subconjunto, sem a utilização de seleção de atributos, isto é o conjunto que contém todos os atributos **No_FS**.

Após a análise da utilização de *resampling* antes da seleção de atributos, partimos para a segunda abordagem de seleção de atributos para detecção de anomalias. Através da segunda abordagem será possível validar se os métodos de seleção de atributos insensíveis ao desbalanceamento entre as classes serão adequadas para selecionar atributos em um cenário de detecção de anomalias, já que em detecção de anomalias o

desbalanceamento entre as classes é mais alto que o desbalanceamento já testado pelas técnicas utilizadas.

Toda essa investigação, entre os passos 2 e 3 da nossa metodologia, representa a principal contribuição deste trabalho. Para avaliar essas etapas, ou seja avaliar a eficácia de um conjunto de atributos, existem abordagens que utilizam métricas independentes de um classificador, relacionados com o mérito de um subconjunto. Entretanto, essas abordagens não são unificadas e dependem da métrica escolhida. Portanto, neste trabalho verificamos se o mérito do conjunto poderia ser um bom indicador da eficácia de uma seleção de atributos.

Na **etapa 4**, **utilizamos 3 técnicas de classificação consideradas estado-da-arte para detectar as anomalias e avaliar os subconjuntos de atributos selecionados**. Foram utilizadas as técnicas *redes bayesianas de classificação, regressão logística* e *árvore de decisão*, as quais foram descritas na Seção 6.1.1.

É importante ressaltar que, para avaliação das estratégias de seleção de atributos, utilizamos um conjunto de dados diferente do que o utilizado nas etapas anteriores. Para garantir a generalidade das soluções, utilizamos a estratégia *8-fold Cross Validation* e mantivemos a proporção original de anomalias entre os *folds*. Ou seja, cada *fold* possui exatamente a mesma porcentagem de anomalias.

Além disso, nessa etapa de treinamento para classificação não utilizamos nenhuma estratégia de *resampling*, pois como gostaríamos de avaliar o impacto do *resampling* na seleção de atributos e comparar os diferentes métodos de seleção, se utilizássemos qualquer estratégia de *resampling* antes do treinamento para classificação, poderíamos estar favorecendo algum método que deseja-se comparar.

Acreditamos que a utilização de algum método de *resampling* antes do treinamento para classificação acarretaria em melhor performance para o classificador. Entretanto, essa estratégia já foi comprovada em diversas pesquisas e não era o foco central deste trabalho. Neste trabalho, o foco central são as estratégias de seleção de atributos. Assim, para avaliação dos modelos, foi utilizado o treinamento dos classificadores sobre a base de dados com a distribuição real e com os atributos selecionados por cada subconjunto.

Para avaliar os resultados dessa etapa, utilizamos 3 métricas de avaliação adequadas para classificação em dados desbalanceados. Foram usadas as técnicas $AUC$ e *F1-Score* e a métrica *Eficiência Econômica* ($EE$), criada por nós para medir o ganho financeiro oferecido pela solução. Essas métricas foram descritas detalhadamente na Seção 6.1.2. Todas essas análises foram validadas estatisticamente através dos testes de *Friedman* e *Wilcoxon*, que foram brevemente descritos na Seção 6.1.2.6.

Após a análise utilizando as métricas aqui descritas, é possível construir o que

denominamos de **modelos de detecção de anomalias**. Esses modelos são compostos por uma técnica de classificação e uma estratégia de seleção de atributos, que obtiveram as melhores performances.

No Capítulo 7 serão apresentados os resultados dos 2 distintos estudos de casos onde aplicamos toda essa metodologia.

# Capítulo 7

# Estudos de Casos

Neste capítulo serão descritos os cenários de aplicação de detecção de anomalias em transações eletrônicas e os resultados experimentais obtidos pela aplicação da nossa metodologia em dois estudos de caso, utilizando base de dados reais.

Nessas bases de dados a anomalia representa fraude em transações eletrônicas, mais especificamente em operações com cartão de crédito. Em ambos os cenários, cada transação é composta por centenas de atributos de diferentes tipos, provenientes do vendedor, do comprador e atributos da própria transação. Portanto, é fundamental a criação e adequação de métodos para selecionar os melhores atributos.

Um desses atributos refere-se ao status das transações, podendo acarretar em transações seguras ou em transações em que ocorreram estorno ou *chargeback*. O *chargeback* é o cancelamento de uma venda feita com cartão de crédito, que pode acontecer por dois motivos:

- Não reconhecimento da compra por parte do titular do cartão, gerando indícios de ocorrência de fraude;
- Não cumprimento das regulamentações previstas nos contratos de compra e venda.

Nesta pesquisa, iremos considerar apenas a primeira situação de *chargeback*.

O processo que pode levar ao *chargeback* em um sistema de pagamento eletrônico, pode ser melhor entendido a partir da Figura 7.1.

Devido a um acordo de confidencialidade com as empresas fornecedoras da base de dados, não podemos divulgar informações quantitativas sobre os dados. Assim, serão apresentadas apenas informações gerais sobre os mesmos, preservando, no entanto, a descrição do cenário utilizado nessa pesquisa. As análises serão feitas individual-

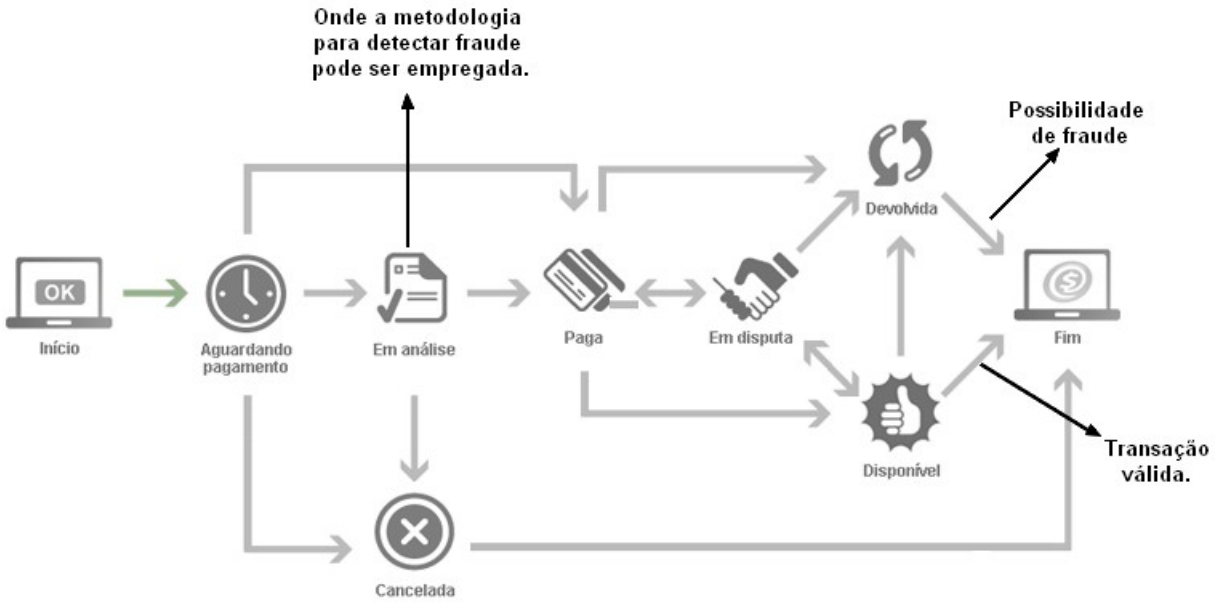


Figura 7.1: Processo pós compra em um sistema de pagamento eletrônico.

mente para cada estudo de caso e posteriormente uma análise geral, comparando os desempenhos dos modelos de detecção de anomalias nos 2 cenários.

## 7.1 Estudo de Caso 1 - Detecção de Fraude utilizando Dados do Sistema de Pagamento Eletrônico UOL PagSeguro

No Estudo de Caso 1, utilizamos uma base de dados real do sistema de pagamento eletrônico PagSeguro[1] para validar nossa metodologia. PagSeguro é um dos sistemas de pagamento eletrônico mais utilizados na América Latina. Portanto, fica clara a necessidade de técnicas eficientes para identificar fraudes nesse sistema para prover mais segurança aos usuários e reduzir o deficit econômico causado pelas fraudes.

### 7.1.1 Caracterização da base de dados

A base de dados provida pelo PagSeguro contém milhões de transações ocorridas entre 2012 e 2013, onde cada transação é composta por 380 atributos (*features*) de diferentes tipos. Existem *features* relativas aos vendedores, compradores e da própria transação,

[1] http://pagseguro.uol.com.br

ficando clara a necessidade de técnicas para selecionar as informações mais relevantes neste cenário. Um desses atributos refere-se ao status da transação, que pode ser uma transação válida ou uma transação anômala, que pode ser fraudulenta.

Uma das mais importantes características deste cenário é o alto desbalanceamento entre as classes, com números próximos a 1% de transações fraudulentas, caracterizando um típico problema de detecção de anomalias. A Tabela 7.1 apresenta as principais informações sobre esta base de dados.

| | |
|---|---|
| **Número de atributos** | 380 |
| **Números de atributos contínuos** | 248 |
| **Números de atributos categóricos** | 132 |
| **Proporção de fraude** | $\cong 1\%$ |
| **Período de análise** | 2012/2013 |

Tabela 7.1: Base de Dados - Visão Geral

Na Tabela 7.1 podemos notar o grande volume de atributos de tipos diferentes e o alto desbalanceamento entre as classes. Essas duas característica tornam a seleção de atributos peculiar e representa um grande desafio deste trabalho.

Na Seção 7.1.2 apresentamos os resultados da aplicação da nossa metodologia sobre a base de dados apresentada na Seção 7.1.1, a qual é uma base de dados real de um dos principais sistemas de pagamento eletrônico do Brasil.

### 7.1.2 Resultados Experimentais

Nesta seção comparamos os resultados obtidos após a aplicação dos modelos para detecção de fraude, usando a metodologia explicada na Seção 6, sobre a base de dados real provenientes do sistema de pagamento eletrônico PagSeguro, explicado na Seção 7.1.

Os experimentos são realizados para comparar a eficiência de duas abordagens utilizadas de seleção de atributos para detecção de anomalias. A primeira abordagem utiliza técnicas de seleção de atributos tradicionais aliadas com métodos de *resampling* antes da seleção de atributos. Esses resultados serão apresentadas na Subseção 7.1.2.1. A segunda abordagem utiliza técnicas de seleção de atributos com métricas insensíveis ao desbalanceamento entre as classes, apresentando os resultados na Subseção 7.1.2.2.

#### 7.1.2.1 Abordagem 1 - Métodos de *Resampling* antes da Seleção de Atributos

A aplicação dos métodos de *resampling* geraram conjuntos diferentes com as distintas proporções de fraude. A exceção foi a técnica *Sampling Outlier* (*SO*), onde não varia-

mos a taxa de *resampling* ($TR$) e a técnica *NearMiss-3*(NM-3) que, de acordo com o funcionamento explicado na seção 4.3.1.3, gerou os mesmos conjuntos para a proporção de fraude entre 5 e 30%. Assim, obtivemos 45 conjuntos diferentes para selecionar atributos nesse cenário.

As Tabelas 7.2, 7.3 e 7.4 apresentam o mérito e o número de atributos selecionados pelas técnicas *CFS*, *GainRatio* e *Relief*, respectivamente, sobre as amostras de dados diferentes geradas por métodos de *resampling*, variando a proporção. Nessa primeira etapa o mérito de cada conjunto foi calculado utilizando a métrica da própria técnica de seleção, sem a utilização de técnicas de classificação para determinar a eficácia da seleção de atributos.

A partir desses resultados, pretendemos avaliar se o mérito calculado pela técnica de seleção de atributos é um bom indicador da eficácia dos subconjuntos gerados e se esse poderia ser utilizado para comparações. Para isso, é necessário a comparação dos méritos dos subconjuntos com os resultados obtidos após a classificação. Assim, será possível verificar se as métricas adotadas para mérito de um subconjunto de atributos poderia ser um bom indicador para qualidade da seleção de atributos.

A Tabela 7.2 apresenta o mérito de cada subconjunto e entre parenteses o número de atributos selecionados para a técnica *CFS*. Nessa técnica de seleção de atributos o mérito do conjunto é calculado de acordo com a Equação 4.3, mostrada na Seção 4.1.1. Em **negrito** destacamos a proporção de fraudes que apresentou o melhor mérito para cada método de *resampling*.

| **F(%)** | **NM-1** | **NM-2** | **NM-3** | **ROS** | **RUS** | **SMOTE** | **SO** | **Real** |
|---|---|---|---|---|---|---|---|---|
| # | - | - | - | - | - | - | 0,303 (28) | 0,15 (11) |
| **5%** | 0,633 (34) | 0,362 (46) | 0,315 (27) | 0,220 (22) | 0,226 (20) | 0,334 (28) | - | - |
| **10%** | 0,788 (19) | 0,369 (43) | 0,315 (27) | 0,266 (24) | 0,270 (27) | 0,422 (27) | - | - |
| **15%** | 0,786 (20) | **0,458 (32)** | 0,315 (27) | 0,285 (29) | 0,289 (32) | 0,466 (22) | - | - |
| **20%** | **0,791 (19)** | 0,457 (28) | 0,315 (27) | 0,299 (28) | 0,305 (33) | 0,495 (23) | - | - |
| **25%** | 0,780 (18) | 0,448 (28) | 0,315 (27) | **0,305 (27)** | 0,312 (28) | 0,516 (22) | - | - |
| **30%** | 0,770 (19) | 0,441 (26) | 0,315 (27) | 0,305 (27) | **0,317 (25)** | 0,526 (21) | - | - |
| **40%** | 0,752 (18) | 0,423 (22) | **0,319 (26)** | 0,300(23) | 0,312 (23) | **0,536 (21)** | - | - |
| **50%** | 0,732 (14) | 0,407 (22) | 0,317 (26) | 0,284 (26) | 0,310 (28) | 0,535 (20) | - | - |

Tabela 7.2: Mérito do conjunto e número de atributos selecionado pela técnica de seleção de atributos **CFS** sobre o conjunto de dados para seleção. A primeira coluna representa a porcentagem de fraudes utilizada (F(%)), a primeira linha o método de *resampling*, entre parenteses encontra-se o número de atributos selecionados por cada combinação e em **negrito** os subconjuntos com maior mérito para cada método de *resampling*.

A informação mais relevante apresentada na Tabela 7.2 é que todas as técnicas de seleção de atributos quando utilizaram alguma abordagem de *resampling* obtiveram melhores méritos do que quando utilizaram a distribuição real entre as classes. Em-

bora, a partir dessa métrica, não podemos afirmar que os métodos com melhor mérito são os que contém os melhores atributos para detectar anomalias, podemos afirmar que a técnica de *CFS* encontrou melhores méritos para subconjuntos que utilizaram estratégias de *resampling*.

A partir da Tabela 7.2, também podemos perceber que não existe uma relação direta entre número de atributos selecionados e mérito do subconjunto de atributos. Outra observação importante é que a porcentagem de fraudes não é linearmente correlacionada com o mérito do subconjunto. Ou seja, não se pode afirmar que aumentando a proporção de fraude aumenta-se o mérito do subconjunto. Analisando por essa métrica, notamos que não existe uma taxa única para todas os métodos de *resampling*, cada método atingiu melhor performance com uma proporção de fraude.

As Tabelas 7.3 e 7.4 permitem analisar os mesmos aspectos, utilizando respectivamente as técnicas *GainRatio* e Relief. Para essas duas técnicas, o mérito do subconjunto foi calculado utilizando a média da soma dos méritos de cada atributo selecionado pelo corte no *ranking* (como mostrado na Figura 3.1), já que essas técnicas avaliam cada atributo individualmente. Assim, nas tabelas são apresentadas o mérito de cada subconjunto, que podem variar de 0 a 1 e o número de atributos selecionados, a partir da estratégia explicada na Seção 3.1.

| **Tr** | **NM-1** | **NM-2** | **NM-3** | **ROS** | **RUS** | **SMOTE** | **SO** | **Real** |
|---|---|---|---|---|---|---|---|---|
| # | - | - | - | - | - | - | 0,188 (86) | 0,044 (77) |
| **5%** | 0,394 (325) | 0,273 (192) | **0,149 (44)** | 0,086 (81) | 0,094 (74) | 0,125 (67) | - | - |
| **10%** | 0,463 (334) | 0,259 (191) | **0,149 (44)** | 0,120 (77) | 0,135 (52) | 0,160 (69) | - | - |
| **15%** | 0,423 (340) | **0,366 (21)** | **0,149 (44)** | 0,145 (45) | 0,145 (52) | 0,168 (81) | - | - |
| **20%** | 0,389 (340) | 0,334 (23) | **0,149 (44)** | 0,142 (57) | **0,152 (63)** | 0,177 (78) | - | - |
| **25%** | 0,357 (341) | 0,301 (26) | **0,149 (44)** | 0,140 (56) | 0,144 (70) | 0,176 (84) | - | - |
| **30%** | 0,328 (341) | 0,272 (31) | **0,149 (44)** | **0,147 (34)** | 0,147 (42) | 0,176 (80) | - | - |
| **40%** | **0,465 (45)** | 0,221 (41) | 0,142 (37) | 0,129 (37) | 0,128 (51) | **0,182 (37)** | - | - |
| **50%** | 0,389 (56) | 0,179 (44) | 0,126 (41) | 0,111 (35) | 0,132 (12) | 0,164 (54) | - | - |

Tabela 7.3: Mérito do conjunto e número de atributos selecionados pela técnica de seleção de atributos **GainRatio** sobre o conjunto de dados para seleção. Nessa abordagem o mérito do subconjunto de atributos foi calculado pela média da soma dos méritos dos atributos selecionados. A primeira coluna representa a porcentagem de fraudes utilizada (F(%)), a primeira linha o método de *resampling*, entre parenteses encontra-se o número de atributos selecionados por cada combinação e em **negrito** os subconjuntos com maior mérito para cada método de *resampling*.

A partir da Tabela 7.3 podemos reiterar, utilizando a técnica de seleção de atributos *GainRatio*, as considerações feitas para a técnica *CFS*. Ou seja, todos os conjuntos que utilizaram *resampling* obtiveram maior mérito do que o conjunto com a distribuição real. O mérito e o número de atributos não são correlacionados, a porcentagem de

fraude não é linearmente correlacionada com o mérito do subconjunto de atributos e não existe uma proporção adequada para todos os métodos de *resampling*.

| TR | **NM-1** | **NM-2** | **NM-3** | **ROS** | **RUS** | **SMOTE** | **SO** | **Real** |
|---|---|---|---|---|---|---|---|---|
| # | | | | | | | 0,188 (33) | 0,102 (29) |
| **5%** | 0,089 (26) | 0,077 (31) | **0,033 (32)** | 0,09 (26) | **0,085 (30)** | 0,09 (26) | | |
| **10%** | **0,099 (19)** | 0,073 (28) | **0,033 (32)** | 0,076 (31) | **0,085 (23)** | 0,073 (32) | | |
| **15%** | 0,066 (26) | 0,087 (22) | **0,033 (32)** | 0,082 (32) | 0,062 (27) | 0,083 (31) | | |
| **20%** | 0,062 (23) | 0,073 (25) | **0,033 (32)** | 0,094 (31) | 0,049 (30) | 0,086 (34) | | |
| **25%** | 0,05 (26) | **0,089 (17)** | **0,033 (32)** | 0,105 (30) | 0,043 (29) | 0,101 (31) | | |
| **30%** | 0,046 (26) | 0,061 (27) | **0,033 (32)** | 0,11 (31) | 0,042 (25) | 0,094 (38) | | |
| **40%** | 0,046 (20) | 0,067 (20) | 0,029 (30) | 0,139 (27) | 0,034 (26) | 0,138 (27) | | |
| **50%** | 0,055 (17) | 0,072 (18) | 0,03 (22) | **0,157 (27)** | 0,031 (22) | **0,151 (28)** | | |

Tabela 7.4: Mérito do conjunto e número de atributos selecionado pela técnica de seleção de atributos **Relief** sobre o conjunto de dados para seleção. Nessa abordagem o mérito do subconjunto de atributos foi calculado pela média da soma dos méritos dos atributos selecionados. A primeira coluna representa a porcentagem de fraudes utilizada (F(%)), a primeira linha o método de *resampling*, entre parenteses encontra-se o número de atributos selecionados por cada combinação e em **negrito** os subconjuntos com maior mérito para cada método de *resampling*.

Ao analisar a Tabela 7.4, conseguimos descrever alguns aspectos importantes do algoritmo *Relief* para seleção de atributos. Especificamente para essa técnica, nem sempre que se utilizou *resampling* na seleção de atributos obteve-se maior mérito. Esse comportamento não indica uma aleatoriedade para essa técnica, mas sim que pode ser necessário uma calibração da porcentagem de fraudes, para a utilização de *resampling* antes de aplicar o método de seleção *Relief*. Já que essa técnica parece sofrer menos com o desbalanceamento que as demais técnicas.

Embora o mérito do subconjunto de atributos ofereça algumas intuições sobre o comportamento de uma técnica de seleção de atributos, o poder preditivo será apresentado quando utilizado um classificador sobre cada conjunto de dados. A partir desse momento, poderemos afirmar a verdadeira performance de um subconjunto de atributos e fazer conclusões sobre as métricas de mérito utilizadas.

Para auxiliar na análise da influência da proporção de fraude na seleção de atributos, calculamos a distância de *Jaccard* [Jaccard, 1901], par a par para cada técnica de seleção de atributos, de acordo com os seguintes passos:

1. Geramos subconjunto de atributos, para amostras com diferentes proporções de fraude geradas por métodos de *resampling*.

2. Calculamos o número de atributos em comum entre dois subconjuntos de atributos $A$ e $B$ ($|A \cap B|$).

3. Calculamos a distância de *Jaccard* entre cada dois conjuntos dada por:

$$D_j(A, B) = \frac{|A \cup B| - |A \cap B|}{|A \cup B|}$$

Após o cálculo da distância de *Jaccard*, para cada técnica de seleção de atributos tradicional utilizada, obtivemos uma matriz de distância entre os subconjuntos de atributos gerados. Para facilitar a visualização dessa matriz, construímos um gráfico de projeção 2D (*multi-dimensional scaling*) para as distâncias entre cada técnica de seleção de atributos.

Assim, os gráficos da Figura 7.2 apresentam a projeção $2D\ MDS$ para cada um dos subconjuntos gerados pelas técnicas de seleção de atributos tradicionais, utilizando métodos de *resampling* ou mantendo a proporção real de fraudes antes dessa seleção. Os números em cada ponto refere-se à porcentagem de fraude contida na amostra onde foram selecionados os subconjuntos de atributos.

De acordo com os gráficos é possível perceber que a técnica de *CFS* foi mais insensível à mudança da proporção de fraude. Já que os *cluster* formados por essa técnica foram mais coesos, mostrando depender muito mais do método de *resampling* do que a proporção utilizada.

A técnica de *GainRatio* apresentou maior poder de separação entre as diferentes estratégias de *resampling*, já a técnica de *Relief*, obteve *clusters* menos coesos entre os mesmos métodos de *resampling*. Nessa técnica a distância entre as proporções de fraude foram menores que as demais técnicas, formando grupos de subconjuntos com proporções de fraudes semelhantes.

As técnica *CFS* e *GainRatio* apresentaram as maiores distâncias entre os conjuntos gerados utilizando algum método de *resampling* e a proporção real antes da aplicação dessas técnicas. Esse comportamento, adicionado com a análise das Tabelas 7.2, 7.3 e 7.4, onde esses 2 métodos tiveram méritos melhores em qualquer subconjunto utilizando *resampling*, permite-nos concluir que para essa base de dados os métodos *CFS* e *GainRatio* são mais sensíveis ao desbalanceamento entre as classes do que o método *Relief*.

Embora, nas Tabelas 7.2, 7.3 e 7.4 a proporção de fraude não é linearmente correlacionada com os méritos dos conjuntos, percebemos nos gráficos de distância que para grande maioria das técnicas, os subconjuntos mais próximos da distribuição real foram os subconjuntos selecionados em datasets com a distribuição próxima da distribuição real.

A distância entre esses subconjuntos aumenta, conforme a proporção de fraudes,

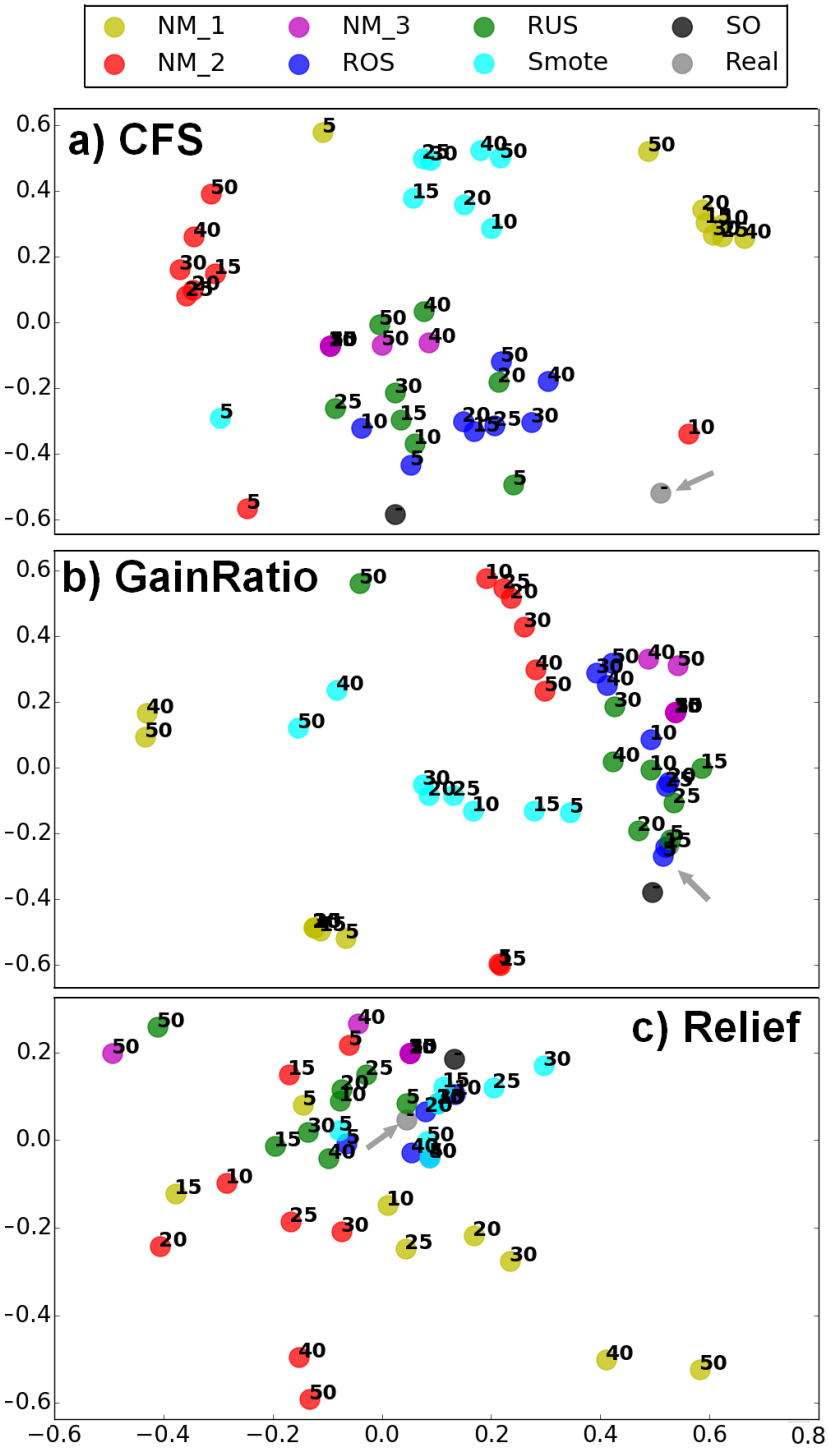


Figura 7.2: Distâncias entre os subconjuntos selecionados pelas técnica *CFS*, *GainRatio* e *Relief* para diferentes métodos de *resampling* e proporção de fraude, comparados com a seleção de atributos realizadas sobre a proporção real de fraudes (Real).

até atingir um limiar $\alpha$. Após a proporção de fraude atingir o valor $\alpha$, o qual varia de acordo com o método de *resampling* utilizado, os dados seguem uma certa alea-

toriedade. Esse comportamento é notado principalmente nas técnicas que formaram *clusters* mais coesos, até que se atinja um limiar de proporção de fraude de 40%.

A fim de analisar a distribuição dos atributos sobre os conjuntos de atributos selecionados, apresentamos a Tabela 7.5, que apresenta um resumo dos atributos mais frequentes em dois cenários distintos: o cenário utilizando *Resampling* antes da seleção de atributos e o cenário mantendo a proporção real. Para os dois cenários, a frequência de um atributo foi calculada pela porcentagem de conjuntos que o atributo está contido. Assim, se um atributo estivesse presente em todos os conjuntos de seleção ele obteria frequência 1.

| Observação de Frequência | Número de Atributos | |
|---|---|---|
| # | ***Resampling*** | **Real** |
| **Presente em algum conjunto** | 375 | 107 |
| **Presente em mais que 25%** | 63 | 107 |
| **Presente em mais que 50%** | 12 | 9 |
| **Presente em mais que 75%** | 2 | 2 |
| **Presente em todos conjuntos** | 0 | 2 |

Tabela 7.5: Frequência dos atributos selecionados, utilizando *resampling* antes da seleção de atributos ou mantendo a proporção real.

Após as análises utilizando apenas as informações sobre os conjuntos de atributos selecionados, aplicamos as técnicas de classificação *Redes Bayesianas* (*Bayes*), *Regressão Logística*(*Logistic*) e *árvore de decisão* (*Tree*), explicadas na Seção 6.1.1. Assim, é possível analisar a eficácia das estratégias de seleção de atributos utilizadas. A partir da classificação sobre todos os subconjuntos de atributos gerados pelas diferentes estratégias, poderemos enfim afirmar quais obtiveram melhor performance e fundamentar nossas hipóteses de pesquisa.

Conforme descrito na metodologia do Capítulo 6, neste trabalho visamos comparar apenas a eficácia dos diferentes métodos de seleção de atributos para detectar anomalias, portanto não utilizamos nenhuma estratégia de *resampling* na classificação.

Todo o treinamento para classificação foi feito sobre conjuntos de dados com proporção real de fraude. Para garantir a generalidade utilizamos *8-fold-cross-validation* e validamos estatisticamente as soluções através dos testes de *Friedman* e *Wilcoxon* (explicados na Seção 6.1.2.6). Para avaliação das soluções, utilizamos as métricas $AUC$, F1 médio ($F1$) e a métrica criada por nós para medir o ganho financeiro das soluções, denominada Eficiência Econômica ($EE$). Tais métricas foram detalhadamente explicadas na Seção 6.1.2.

Os gráficos das Figuras 7.3 apresentam a métrica *AUC* obtida após a classificação em distintos subconjuntos de atributos. Esses subconjuntos foram gerados por cada técnica de seleção de atributos tradicional aplicadas em um *dataset* com distintas proporções de fraude geradas pelos métodos de *resampling* ou mantendo a proporção real.

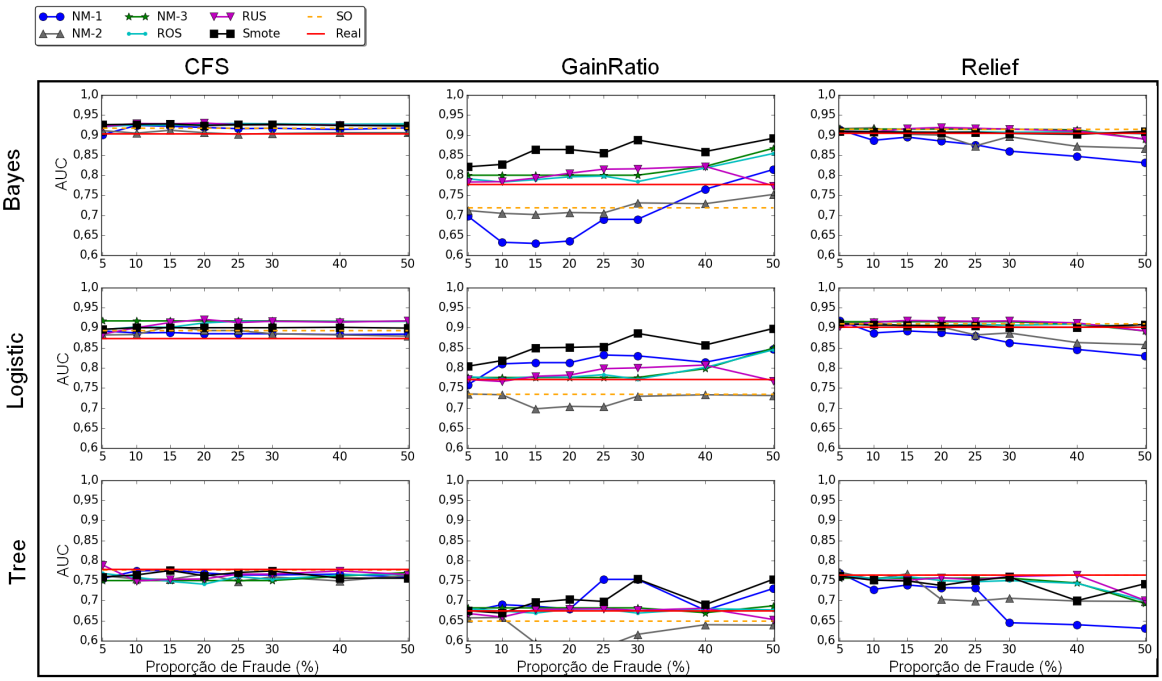


Figura 7.3: *AUC* obtidas por técnicas de classificação, sobre subconjuntos de atributos selecionados pelas técnicas de seleção de atributos em um *dataset* com aplicação de métodos de *resampling* em diferentes proporções.

Podemos notar nos gráficos da Figura 7.3 que, para qualquer técnica de seleção de atributos adotada e qualquer técnica de classificação, sempre houve um método de *resampling* com alguma proporção que foi eficaz para melhorar a performance da seleção de atributos e, consequentemente, da classificação. Assim, é possível afirmar que, a partir da aplicação de um método *resampling* antes da seleção de atributos, podemos construir modelos mais eficazes para detecção de fraudes nesse cenário.

A técnica de seleção de atributos *GainRatio* se mostrou muito mais sensível ao desbalanceamento entre as classes, já que à medida que se variou a proporção de fraude se obteve maiores variações nos resultados encontrados. Nas técnicas *CFS* e *Relief* as variações de resultados de acordo com a proporção de fraudes foram menores. Esse comportamento já havia sido notado no gráfico da Figura 7.2. Mesmo com a menor variação quando utilizado *resampling* antes da seleção de atributos, foram selecionados atributos mais eficazes para detectar fraudes.

Através dos gráficos podemos notar que a técnica de classificação árvore de decisão ($Tree$) obteve os piores valores de $AUC$, para qualquer seleção de atributos feita. Esse comportamento mostra que a técnica de árvore de decisão, com as configurações adotadas, se mostraram mais sensível ao desbalanceamento entre as classes. Assim, as análises para construção de modelos de detecção de fraudes serão realizadas sobre as técnicas de Regressão Logística e Redes Bayesianas.

Um comportamento importante observado nos gráficos foi que, no geral, os métodos de *resampling* ($RUS$) e ($Smote$) alcançaram os melhores resultados para diminuir o desbalanceamento entre as classes antes da seleção de atributos.

O método $SO$, criado por nós e independente da taxa de *resampling* escolhida, apresentou bons resultados quando utilizado antes da seleção de atributos realizada pelas técnicas *CFS* e *Relief*. O método $SO$ superou alguns métodos de *resampling* em diversas proporções de fraudes. Entretanto, quando essa proporção é devidamente calibrada, esse método foi superado. Assim, $SO$, mostrou ser um bom método para aumentar a proporção de exemplos da classe fraude antes da seleção pela técnica *CFS* e *Relief*, quando não se sabe a proporção de anomalias adequada.

Para facilitar a análise sobre as diferentes proporções de fraudes utilizadas pelos métodos de *resampling* e determinar a proporção adequada para construir os modelos de detecção de fraudes, construímos os gráficos das Figuras 7.4 e 7.5. Nessas figuras ajustamos a escala de cada subgrafo, a fim de facilitar a visualização entre os diferentes pontos de proporções de fraude.

Os gráficos da Figura 7.4 apresentam os mesmos resultados que os gráficos da Figura 7.3, entretanto, com uma visualização mais próxima entre os distintos pontos de proporção de fraude. Os gráficos da Figura 7.5 ilustram a análise similar para a métrica $EE$. Cabe ressaltar que o principal intuito desses gráficos é a comparação entre as diferentes proporções de fraudes para uma mesma técnica de seleção de atributos e classificação. Apesar de estarem na mesma figura, os gráficos estão em escalas diferentes, inviabilizando a comparação entre técnicas distintas. Os resultados para a técnica de classificação árvore de decisão não foram apresentados nesses gráficos, pois, conforme mostrado na Figura 7.3, essa técnica obteve os piores resultados.

Como primeira análise dos gráficos das Figuras 7.4 e 7.5, os comparamos com as tabelas 7.2, 7.3 e 7.4, as quais mostram o mérito da seleção de atributos através de uma função de mérito, sem a utilização de técnicas de classificação. Através dessa comparação, pretendemos analisar se os subconjuntos de atributos com melhor mérito obtiveram os melhores resultados para classificação. Assim, poderemos concluir se essas funções poderiam ser utilizadas para avaliar a performance da seleção de atributos nesse cenário de detecção de anomalias.

Notamos que o mérito de cada conjunto de atributos não influenciou diretamente nos resultados obtidos após a classificação. Através do mérito do subconjunto de atributos não foi possível inferir qual o melhor método de *resampling* para aplicação antes da seleção de atributos.

Para indicar a melhor proporção de fraude, em cada método de *resampling*, o mérito do conjunto obteve um comportamento mais próximo do encontrado no classificador. Entretanto, ainda assim, não obteve resultados exatos para se confirmar como uma estratégia eficaz para esse tipo de análise. Portanto, a análise pela estratégia de mérito do subconjunto somente pode ser uma alternativa válida, para indicar a proporção ideal de anomalias do método de *resampling*, quando o custo computacional de se realizar uma classificação para essa análise seja alto.

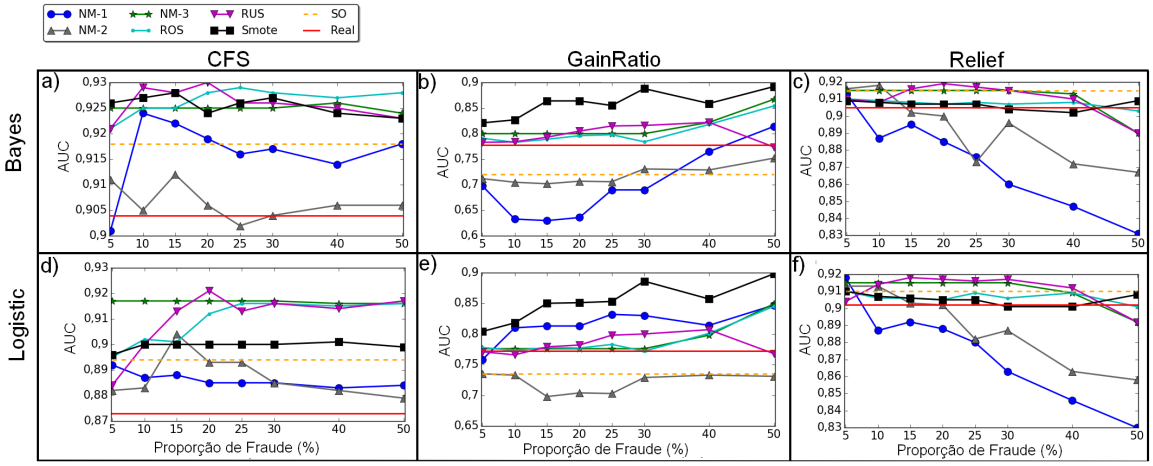


Figura 7.4: Visualização das melhores proporções de fraudes, através da métrica **AUC** obtidas por técnicas de classificação, sobre subconjuntos de atributos selecionados pelas técnicas de seleção de atributos em um dataset com aplicação de métodos de *resampling* em diferentes proporções.

Através da análise dos gráficos das Figuras 7.4 e 7.5, podemos notar alguns comportamentos interessantes, referentes à aplicação dos métodos de *resampling* com diferentes proporções de fraudes antes da seleção de atributos. Uma característica interessante é que a eficácia das estratégias não crescem linearmente de acordo com o aumento da proporção de fraude antes da seleção de atributos.

Para as técnicas de seleção de atributos *CFS* e *Relief*, os melhores resultados foram alcançados utilizando as proporções de fraudes de 5% a 25%. Ou seja, à medida que se aumentou a proporção de fraude, após um certo limiar, houve uma diminuição na eficácia dos conjuntos de atributos selecionados. Uma possível explicação para esse fator é a remoção de instâncias fundamentais para uma boa seleção de atributos, em

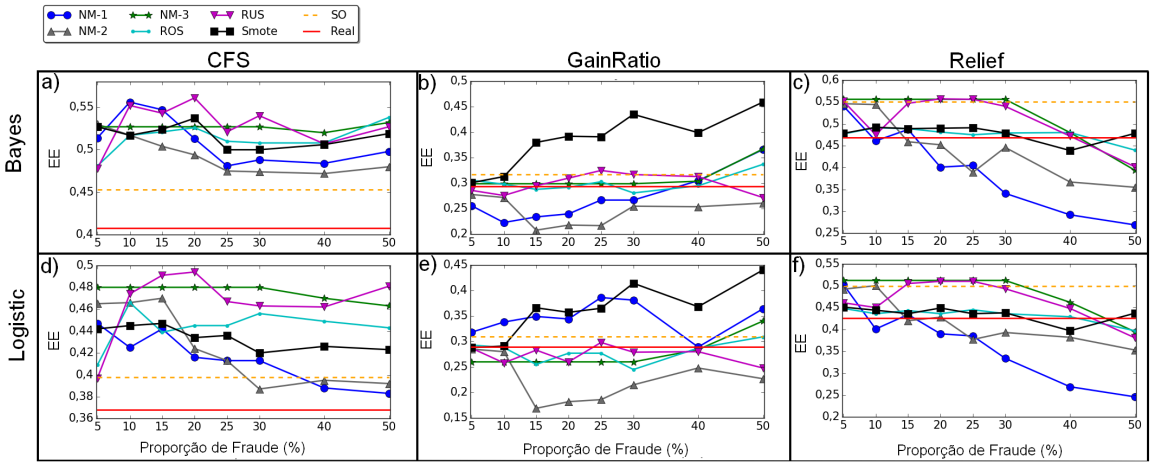


Figura 7.5: Visualização das melhores proporções de fraudes, através da métrica **Eficiência Econômica** obtidas por técnicas de classificação, sobre subconjuntos de atributos selecionados pelas técnicas de seleção de atributos em um dataset com aplicação de métodos de *resampling* em diferentes proporções.

casos de aplicação de métodos de *undersampling* e o *overfitting* causados por replicação de instâncias em casos de métodos de *oversampling*.

Por outro lado, a técnica *GainRatio* selecionou melhores atributos em distribuições com maiores proporções de fraudes. Uma possível explicação é que a técnica *GainRatio* se mostrou a mais sensível ao desbalanceamento ente as classes do que as demais técnicas de seleção de atributos utilizadas. Assim, percebe-se que o balanceamento entre as classes é fator limitante para métrica de seleção utilizada pela técnica *GainRatio*.

Outro fator interessante observado nos gráficos é a comparação entre as técnicas aleatorias de *resampling*. Podemos notar que, em geral, a técnica de *Random Undersampling* obteve melhores resultados que as técnicas de *Random Oversampling*, para grande maioria das proporções de fraudes. Esse comportamento havia sido observado em [Drummond et al., 2003], entretanto para aplicação de *resampling* na fase de treinamento para classificação.

Podemos observar nos gráficos das Figuras 7.4 e 7.5 que, em geral, não existe uma mesma taxa ideal de proporção de fraude para todos os métodos de *resampling*. Percebemos que cada método obteve melhor performance utilizando uma determinada proporção de fraude, fazendo com que esse parâmetro seja ajustado conforme o método.

Portanto, neste trabalho padronizamos a escolha da proporção de fraude através da combinação de método de *resampling* e método de seleção de atributos. A Tabela 7.6 apresenta essa proporção de fraude para cada método de *resampling* e o ganho

percentual em comparação com a mesma combinação de técnicas, mas utilizando a proporção real antes da etapa de seleção de atributos.

Devido a limitação de espaço, após observar que os resultados obtidos pela métrica de classificação de árvore de decisão foram significativamente abaixo dos demais, a Tabela 7.6 apresenta os ganhos obtidos quando se utilizou as técnicas redes bayesianas (*Bayes*) e regressão logística (*Logistic*).

| | | | Bayes | | | Logistic | | |
|---|---|---|---|---|---|---|---|---|
| FS | Resampling | % | AUC | F1 | EE | AUC | F1 | EE |
| CFS | NM-1 | 10 | 2,21 | 3,02 | 36,27 | 1,60 | 1,42 | 15,49 |
| | NM-2 | 5 | 0,77 | 3,02 | 29,90 | 1,03 | 3,68 | 26,36 |
| | NM-3 | 5:30 | 2,32 | 3,84 | 29,17 | 5,04 | 4,11 | 30,43 |
| | ROS | 25 | 2,77 | 3,98 | 25,00 | 4,93 | 3,12 | 20,92 |
| | RUS | 20 | **2,88** | **5,21** | **37,50** | **5,50** | **5,38** | **34,24** |
| | Smote | 10 | 2,54 | 3,70 | 26,72 | 3,09 | 3,54 | 20,92 |
| | SO | X | 1,55 | 2,06 | 11,03 | 2,41 | 3,54 | 8,15 |
| Gain Ratio | NM-1 | 50 | 4,63 | 1,48 | 24,49 | 9,72 | 2,38 | 25,95 |
| | NM-2 | 50 | -3,34 | -3,99 | -11,22 | -5,31 | -4,76 | -21,45 |
| | NM-3 | 50 | 11,44 | 3,40 | 24,83 | 9,97 | 1,79 | 17,65 |
| | ROS | 50 | 9,77 | 2,07 | 14,63 | 9,46 | 0,74 | 6,92 |
| | RUS | 40 | 5,66 | 1,03 | 6,46 | 4,53 | 0,45 | -3,11 |
| | Smote | 50 | **14,65** | **6,65** | **56,12** | **16,32** | **7,89** | **52,25** |
| | SO | X | -7,46 | -2,66 | 7,82 | -4,79 | -2,23 | 7,27 |
| Relief | NM-1 | 5 | 0,88 | 2,17 | 15,60 | **1,77** | 1,93 | 18,08 |
| | NM-2 | 5 | 1,22 | 2,99 | 16,67 | 1,22 | 1,38 | 15,49 |
| | NM-3 | 5:30 | 1,10 | 3,26 | 18,80 | 1,44 | **2,20** | **20,19** |
| | ROS | 25 | 0,33 | 1,36 | 1,50 | 0,78 | 1,65 | 4,46 |
| | RUS | 20 | **1,55** | **3,67** | **19,02** | 1,66 | 1,65 | 19,72 |
| | Smote | 25 | 0,22 | 1,09 | 4,91 | 0,33 | 0,96 | 2,35 |
| | SO | | 1,10 | 2,72 | 17,52 | 0,89 | 1,10 | 17,14 |

Tabela 7.6: Ganho percentual na detecção de fraude utilizando *resampling* antes da seleção de atributos, comparados com as mesmas técnicas utilizando proporção real antes da etapa de seleção de atributos.

Na Tabela 7.6 destacamos em **negrito** os melhores ganhos percentuais, utilizando *resampling* antes da seleção de atributos, para cada técnica de classificação e de seleção de atributos. Conforme observado em análises anteriores, a técnica *GainRatio* apresenta as maiores diferenças entre os modelos que utilizaram a proporção real e os modelos que utilizaram *resampling* na seleção de atributos. Mostrando assim que a técnica *GainRatio* é muito sensível ao alto desbalanceamento entre as classes. Já o método *Relief* se mostrou menos sensível ao desbalanceamento entre as classes. Ainda assim, a estratégia de *resampling* antes da seleção de atributos melhorou a performance dos modelos que utilizaram *Relief* para selecionar os atributos.

Podemos perceber que utilizando os métodos de *resampling* antes da seleção de atributos obtêm-se ganhos em termos de $AUC$, $F1$ e $EE$, mostrando que essa pode ser uma boa estratégia para melhorar a eficácia da seleção de atributos antes da detecção

de fraude. Os maiores ganhos foram obtidos quando utilizamos a Eficiência Econômica ($EE$) como métrica de avaliação. O principal fator para obter ganhos econômicos é uma alta revocação para fraudes. Já que o custo de um verdadeiro positivo ($TP$) e um falso positivo não é o mesmo nessa métrica. Simulando o que acontece em um cenário real, o custo é de 97% para $TP$ e de 3% para $FP$. Enquanto que nas demais métricas utilizadas neste trabalho o custo de um $TP$ e $FP$ é o mesmo.

Portanto, percebemos neste trabalho que quando são utilizados métodos de *re-sampling* antes da seleção de atributos aumenta-se consideravelmente a revocação de fraudes e, consequentemente, a *Eficiência Econômica* dos modelos. Cabe ressaltar que, apesar da métrica $EE$ ser um bom indicador da eficácia dos modelos de detecção de fraude, essa não foi utilizada de forma isolada nesse trabalho, pois existem problemas além do financeiro em cometer altas taxas de falso positivos em cenários de detecção de fraudes. Como por exemplo, perda de clientes e piora da reputação da empresa.

De posse dos subconjuntos de atributos que atingiram os melhores resultados para detecção de fraude, para cada técnica de classificação criamos o método de seleção de atributos ***Merge***. Esse seleciona os atributos mais frequentes nos subconjuntos de atributos que obtiveram os melhores resultados, conforme explicado no Capítulo 6. Para seleção de atributos, nesse método foi criado um *ranking* de acordo com a frequência do atributo e realizado o corte no *ranking* através do joelho da curva, conforme descrito na Seção 3.1.

Vale ressaltar que, utilizando a estratégia de seleção pelo método $Merge$, os melhores subconjuntos foram gerados pela combinação de uma técnica de seleção de atributos e método de *resampling*. Não foi encontrado nenhum subconjunto de atributos entre os melhores, que foi selecionado com a aplicação de técnicas de seleção de atributos sobre a proporção real de fraude.

Enfim, após diversas análises, podemos consolidar e comparar os modelos de detecção de fraude, formados por uma estratégia de seleção de atributos e uma técnica de classificação. Todos os modelos utilizam as mesmas técnicas de classificação, mas se diferenciam pela estratégia utilizada na seleção de atributos. Assim, podemos definir 4 tipos de modelos de detecção de fraude, que se diferenciam pela abordagem utilizada para selecionar os atributos. São eles:

1. Os modelos gerados pela aplicação de técnicas de classificação sobre o conjunto total de atributos, ou seja sem nenhuma seleção de atributos, denominados de **No_FS**.

2. Modelos construídos com os atributos selecionados pelas técnicas de seleção de atributos sobre a base de dados com a **proporção real de fraudes**.

3. Modelos construídos com os atributos selecionados sobre a amostra de dados **com a utilização de *resampling***, utilizando a proporção de fraude apresentada na Tabela 7.6.

4. Modelos que utilizaram os atributos selecionados pela estratégia *Merge*, explicada anteriormente.

A fim de tornar mais clara a comparação dos modelos de detecção de fraude que utilizam a mesma técnica de classificação, a partir da validação estatística dos resultados obtidos, criamos um novo *score* baseado na comparação par a par desses modelos. Essa métrica foi denominada *pontos* e foi calculada da seguinte forma:

- 3 pontos por cada vitória, ou seja, se um modelo $x$ é estatisticamente melhor que um modelo $y$ para uma determinada métrica, o modelo $x$ recebe 3 pontos.

- 1 ponto por empate, isto é, quando não há diferença estatística entre os dois modelos para uma determinada métrica.

As comparações estatísticas foram realizadas de forma pareada, através do método de *Wilcoxon* (explicado na Seção 6.1.2.6).

As Tabelas 7.7, 7.8, 7.9 apresentam os valores das métricas e os *pontos* (Ptos) de cada modelo. A pontuação máxima para cada modelo, utilizando uma mesma técnica de seleção de atributos é de 81 pontos.

| | **Bayes** | | | | **Logistic** | | | | **Tree** | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | AUC | F1 | EE | Ptos | AUC | F1 | EE | Ptos | AUC | F1 | EE | Ptos |
| **NM-1** | 0,924 | 0,751 | 0,556 | 52 | 0,887 | 0,716 | 0,425 | 18 | 0,774 | 0,715 | 0,399 | 35 |
| **NM-2** | 0,911 | 0,751 | 0,53 | 32 | 0,882 | 0,732 | 0,465 | 29 | 0,762 | 0,71 | 0,379 | 29 |
| **NM-3** | 0,925 | 0,757 | 0,527 | 45 | 0,917 | 0,735 | 0,48 | 53 | 0,75 | 0,716 | 0,397 | 35 |
| **Real** | 0,904 | 0,729 | 0,408 | 9 | 0,873 | 0,706 | 0,368 | 5 | 0,778 | 0,699 | 0,309 | 11 |
| **ROS** | 0,929 | 0,758 | 0,51 | 49 | 0,916 | 0,728 | 0,445 | 39 | 0,76 | 0,718 | 0,376 | 36 |
| **RUS** | 0,93 | 0,767 | 0,561 | **71** | 0,921 | 0,744 | 0,494 | **67** | 0,754 | 0,718 | 0,385 | 34 |
| **Smote** | 0,927 | 0,756 | 0,517 | 46 | 0,9 | 0,731 | 0,445 | 35 | 0,764 | 0,713 | 0,384 | **38** |
| **SO** | 0,918 | 0,744 | 0,453 | 20 | 0,894 | 0,731 | 0,398 | 27 | 0,777 | 0,707 | 0,352 | 25 |
| ***Merge** | 0,908 | 0,759 | 0,541 | 38 | 0,913 | 0,749 | 0,501 | 62 | 0,774 | 0,716 | 0,389 | 31 |
| ***No_FS** | 0,799 | 0,723 | 0,437 | 2 | 0,859 | 0,726 | 0,468 | 25 | 0,763 | 0,703 | 0,368 | 25 |

Tabela 7.7: Performance dos modelos de detecção de fraude utilizando **CFS** com *resampling*. *Os conjuntos de atributos *Merge* e *No_FS* foram utilizados para comparação.

A primeira análise sobre as Tabelas 7.7, 7.8, 7.9 nos permite comprovar a importância da aplicação de técnicas de seleção de atributos para construção de modelos de detecção de fraude. Podemos notar que, para qualquer técnica de classificação utilizada, o modelo sem a aplicação de nenhuma técnica de seleção de atributos foi superado

| | Bayes | | | | Logistic | | | | Tree | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | AUC | F1 | EE | Pontos | AUC | F1 | EE | Pontos | AUC | F1 | EE | Pontos |
| **NM-1** | 0,814 | 0,687 | 0,366 | 38 | 0,847 | 0,688 | 0,364 | 41 | 0,73 | 0,684 | 0,308 | 52 |
| **NM-2** | 0,752 | 0,65 | 0,261 | 4 | 0,731 | 0,64 | 0,227 | 1 | 0,639 | 0,627 | 0,177 | 0 |
| **NM-3** | 0,867 | 0,7 | 0,367 | 53 | 0,849 | 0,684 | 0,34 | 45 | 0,687 | 0,678 | 0,264 | 46 |
| **Real** | 0,778 | 0,677 | 0,294 | 15 | 0,772 | 0,672 | 0,289 | 20 | 0,675 | 0,661 | 0,235 | 20 |
| **ROS** | 0,854 | 0,691 | 0,337 | 41 | 0,845 | 0,677 | 0,309 | 31 | 0,677 | 0,666 | 0,238 | 22 |
| **RUS** | 0,822 | 0,684 | 0,313 | 29 | 0,807 | 0,675 | 0,28 | 24 | 0,681 | 0,664 | 0,232 | 21 |
| **Smote** | 0,892 | 0,722 | 0,459 | **68** | 0,898 | 0,725 | 0,44 | **68** | 0,753 | 0,709 | 0,38 | **66** |
| **SO** | 0,72 | 0,659 | 0,317 | 9 | 0,735 | 0,657 | 0,31 | 10 | 0,65 | 0,643 | 0,274 | 19 |
| **Merge** | 0,908 | 0,759 | 0,541 | **81** | 0,913 | 0,749 | 0,501 | **79** | 0,774 | 0,716 | 0,389 | **71** |
| **No_FS** | 0,799 | 0,723 | 0,437 | 53 | 0,859 | 0,726 | 0,468 | 64 | 0,763 | 0,703 | 0,368 | 64 |

Tabela 7.8: Performance dos modelos de detecção de fraude utilizando **GainRatio** com estratégias de *resampling*.*Os conjuntos de atributos *Merge* e *No_FS* foram utilizados para comparação.

| | Bayes | | | | Logistic | | | | Tree | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | AUC | F1 | EE | Pontos | AUC | F1 | EE | Pontos | AUC | F1 | EE | Pontos |
| **NM-1** | 0,913 | 0,752 | 0,541 | 42 | 0,918 | 0,74 | 0,503 | 46 | 0,769 | 0,708 | 0,394 | 35 |
| **NM-2** | 0,916 | 0,758 | 0,546 | 52 | 0,913 | 0,736 | 0,492 | 36 | 0,771 | 0,71 | 0,403 | **37** |
| **NM-3** | 0,915 | 0,76 | 0,556 | 51 | 0,915 | 0,742 | 0,512 | **66** | 0,756 | 0,711 | 0,392 | 35 |
| **Real** | 0,905 | 0,736 | 0,468 | 11 | 0,902 | 0,726 | 0,426 | 14 | 0,764 | 0,703 | 0,344 | 22 |
| **ROS** | 0,908 | 0,746 | 0,475 | 18 | 0,909 | 0,738 | 0,445 | 21 | 0,747 | 0,701 | 0,342 | 18 |
| **RUS** | 0,919 | 0,763 | 0,557 | **59** | 0,917 | 0,738 | 0,51 | 44 | 0,757 | 0,711 | 0,389 | 5 |
| **Smote** | 0,907 | 0,744 | 0,491 | 24 | 0,905 | 0,733 | 0,436 | 16 | 0,75 | 0,701 | 0,345 | 19 |
| **SO** | 0,915 | 0,756 | 0,55 | 50 | 0,91 | 0,734 | 0,499 | 32 | 0,764 | 0,708 | 0,377 | 26 |
| **Merge** | 0,908 | 0,759 | 0,541 | 44 | 0,913 | 0,749 | 0,501 | 56 | 0,774 | 0,716 | 0,389 | 35 |
| **No_FS** | 0,799 | 0,723 | 0,437 | 2 | 0,859 | 0,726 | 0,468 | 11 | 0,763 | 0,703 | 0,368 | 27 |

Tabela 7.9: Performance dos modelos de detecção de fraude utilizando **Relief** com estratégias de *resampling*. *Os conjuntos de atributos *Merge* e *No_FS* foram utilizados para comparação.

ou ao menos igualado estatisticamente por algum modelo, independente da técnica de seleção de atributos utilizada.

A segunda análise sobre as tabelas reitera como o alto desbalanceamento entre as classes reduz a eficácia das técnicas de seleção de atributos tradicionais. Podemos notar que a grande maioria dos modelos que utilizaram *resampling* antes da seleção de atributos, obtiveram resultados superiores aos modelos do tipo 2, os quais mantém a proporção real para realizar a seleção de atributos. Através dessa análise reiteramos que a utilização de *resampling* antes da seleção de atributos pode ser uma boa estratégia para melhorar a eficácia da seleção de atributos.

O método *Merge*, criado com os atributos mais frequentes dos melhores modelos, se mostrou apenas um método regular. Esse método somente se mostrou eficaz quando comparado com a técnica de seleção de atributos *GainRatio*. Para as outras duas técnicas *CFS* e Relief a combinação do método de seleção de atributos, com um método de *resampling*, aplicado antes da seleção, se mostrou mais eficaz. O destaque para essa

tarefa foi o método *Random Undersampling* (*RUS*) que, apesar de ser um método simples, alcançou bons resultados ao diminuir o desbalanceamento entre as classes antes da seleção de atributos.

O melhor modelo para cada técnica de seleção de atributos, dentre cada uma das Tabelas (7.7, 7.8 e 7.9), são agrupados e comparados na Tabela 7.10, através das 3 métricas de avaliação (*AUC*, *F*1 e *EE*). Essa tabela apresenta os melhores modelos de detecção de fraudes utilizando a estratégia de *resampling* antes da seleção de atributos.

| **FS** | **Resampling** | **Classificação** | **Métricas** | | |
|---|---|---|---|---|---|
| | | | AUC | F1 | EE |
| **CFS** | RUS | Bayes | **0,93 ± 0,006** | **0,767 ± 0,005** | **0,561 ± 0,033** |
| **GainRatio** | Smote | Logistic | 0,898 ± 0,006 | 0,725 ± 0,004 | 0,44 ± 0,026 |
| **Relief** | RUS | Bayes | 0,919 ± 0,005 | 0,763 ± 0,004 | 0,557 ± 0,021 |

Tabela 7.10: Melhores modelos de detecção de fraude para cada técnica de seleção de atributos realizada sobre um conjunto de dados com menor desbalanceamento entre as classes, gerados por algum método de *resampling*.

É possível notar na Tabela 7.10 que o modelo de detecção de fraude composto pela técnica de classificação *Redes Bayesianas*, a técnica de seleção de atributos *CFS* e o método de *resampling Random Undersampling*, obteve os melhores resultados. Esse modelo alcançou eficiência econômica de 0,561. Em outras palavras, se esse modelo fosse usado os prejuízos econômicos ocasionados por fraude na companhia, poderia ser reduzido em 56,1%.

A Subseção 7.1.2.2 apresenta os resultados experimentais para a aplicação da segunda abordagem de seleção de atributos analisada no Estudo de Caso 1. Serão apresentados os resultados e análises após a aplicação de métodos de seleção de atributos considerados insensíveis ao desbalanceamento entre as classes.

#### 7.1.2.2 Abordagem 2 - Aplicação de Técnicas de Seleção de Atributos Insensíveis ao Desbalanceamento entre Classes

A segunda etapa de resultados apresenta as análises sobre as aplicações das técnicas de seleção de atributos que utilizam métricas insensíveis ao desbalanceamento entre as classes. Foram analisadas as técnicas *Fast e Hell* ( ver Seções 5.1.1 e 5.1.2) que obtiveram bons resultados em outros cenários de dados desbalanceados e o método *TBFS* (ver Seção 5.1.3), que utiliza 5 métricas consideradas insensíveis ao desbalanceamento entre as classes (apresentadas na Tabela 5.1).

Cabe ressaltar que todas essas técnicas geram um *ranking* de atributos, esse corte no *ranking* novamente foi feito de acordo com o joelho da curva. A Tabela 7.11 apresenta o número de atributos selecionados por cada uma dessas técnicas.

| Técnica | Número de Atributos |
|---|---|
| Fast | 167 |
| Hell | 128 |
| TBFS-ACC22 | 45 |
| TBFS-AGM | 130 |
| TBFS-BNS | 130 |
| TBFS-F1 | 16 |
| TBFS-MCC | 128 |
| TBFS-PRC | 346 |

Tabela 7.11: Número de Atributos Selecionados pelas técnicas de seleção de atributos que utilizam métricas insensíveis ao desbalanceamento entre as classes.

A intenção dessa etapa é analisar se técnicas existentes para dados desbalanceados ou as adequações de técnicas realizadas nesse trabalho são eficazes para seleção de atributos em um cenário de detecção de anomalias. Nessa etapa, todas as técnicas de seleção de atributos foram aplicadas sobre a proporção real de fraudes, sem a utilização de nenhum método de *resampling*. Todos os resultados foram avaliados sobre as 3 métricas de performance utilizadas nesse trabalho, $AUC$, $F1$ e $EE$.

As Tabelas 7.12, 7.13 e 7.14 apresentam os resultados comparativos dos modelos de detecção de fraude utilizando, respectivamente, as técnicas de classificação *redes bayesianas, regressão logística e árvore de decisão* sobre os subconjuntos de atributos gerados pelas técnicas que utilizam métricas insensíveis ao desbalanceamento entre as classes.

Nessas tabelas são feitas comparações de performance par a par, utilizando o teste estatístico *Wilcoxon*. Assim, é atribuído v se a técnica representada pela linha superou a técnica da coluna, x em casa de derrota, e $-$ se as duas técnicas forem estatisticamente iguais. Conforme explicado anteriormente, novamente atribuímos pontuação para as técnicas de acordo com o número de vitórias (3 pontos) e empates (1 ponto).

Destacamos em **negrito** nas Tabelas 7.12, 7.13 e 7.14 os melhores resultados alcançados em cada técnica de classificação. É possível notar que o método de seleção de atributos ($TBFS$), utilizando a métrica $BNS$ ($Bi\ Normal\ Separation$), em geral, obteve os melhores resultados. Outros dois métodos que obtiveram destaques foram $Fast$ e $Hell$, que são métodos propostos especificamente para seleção de atributos em dados desbalanceados.

O método $TBFS$, utilizando as métrica $PRC$ (área sobre a curva precisão x revocação) e $F1$ (media harmônica entre precisão e revocação) obtiveram os piores resultados. Notamos nessa tabela que alguns métodos foram melhores em uma métrica de avaliação e piores em outros. Isso ocorre devido a natureza do método de seleção que explora conceitos diferentes para selecionar os atributos.

| | Método de Seleção de Atributos | | | | | | | | Score | V | E | Ptos |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **ACC2** | **AGM** | **BNS** | **Fast** | **F1** | **Hell** | **MCC** | **PRC** | | | | |
| **ACC2** | | v | x | - | - | v | v | v | 0,886 | 8 | 6 | 30 |
| | | - | x | x | v | x | - | v | 0,718 | | | |
| | | - | x | x | v | x | - | v | 0,386 | | | |
| **AGM** | | | x | v | v | v | - | v | 0,882 | 10 | 5 | 35 |
| | | | x | v | v | x | - | v | 0,729 | | | |
| | | | x | x | v | v | - | v | 0,497 | | | |
| **BNS** | | | | - | - | - | v | v | **0,9** | **15** | **6** | **51** |
| | | | | v | v | - | v | v | **0,753** | | | |
| | | | | - | v | - | v | v | **0,531** | | | |
| **Fast** | | | | | - | - | x | v | 0,875 | 9 | 6 | 33 |
| | | | | | v | x | x | v | 0,728 | | | |
| | | | | | v | - | v | v | 0,509 | | | |
| **F1** | | | | | | - | x | v | 0,869 | 2 | 5 | 11 |
| | | | | | | x | x | v | 0,689 | | | |
| | | | | | | x | x | - | 0,363 | | | |
| **Hell** | | | | | | | x | v | 0,867 | 11 | 6 | 39 |
| | | | | | | | v | v | 0,74 | | | |
| | | | | | | | v | v | 0,484 | | | |
| **MCC** | | | | | | | | v | 0,881 | 9 | 5 | 32 |
| | | | | | | | | v | 0,731 | | | |
| | | | | | | | | v | 0,444 | | | |
| **PRC** | | | | | | | | | 0,814 | 0 | 1 | 1 |
| | | | | | | | | | 0,537 | | | |
| | | | | | | | | | 0,366 | | | |

Tabela 7.12: Comparativo da classificação por **Redes Bayesianas** sobre técnicas de seleção de atributos com métricas insensíveis ao desbalanceamento entre as classes. A referência de comparação deve ser sempre a linha.

Para tentar utilizar apenas os pontos fortes dos melhores métodos de seleção, implementamos a *fronteira de Pareto*, explicada na Seção 5.2, para selecionar os atributos que satisfazem as condições de fronteira. A escolha das métricas para formar a dimensão da fronteira de Pareto foi feita de forma analítica e empírica, observando as melhores métricas individuais e testando algumas configurações com as possíveis melhores métricas.

Assim, foram utilizadas como dimensão para fronteira todas as métricas para dados desbalanceados mostradas na Seção 5.1, exceto a métrica $PRC$, que obteve os piores resultados individuais com o método $TBFS$ para selecionar atributos. Essa abordagem de fronteira de Pareto selecionou 33 atributos, dentro de um conjunto de 380 possíveis para detectar anomalias.

A Tabela 7.15 apresenta os resultados obtidos pelas técnicas de classificação redes bayesianas e regressão logística sobre os atributos selecionados pela estratégia de fronteira de Pareto, comparados com os métodos individuais $TBFS - BNS$, $Fast$ e $Hell$ que obtiveram os melhores resultados individualmente.

Para facilitar a comparação dos resultados na Tabela 7.15, acrescentamos os

| | Técnicas de Seleção de Atributos | | | | | | | | Score | V | E | Ptos |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **ACC2** | **AGM** | **BNS** | **Fast** | **F1** | **Hell** | **MCC** | **PRC** | | | | |
| **ACC2** | | v | - | v | v | - | v | v | 0,872 | 8 | 7 | 31 |
| | | - | x | x | v | x | - | v | 0,688 | | | |
| | | - | x | x | v | x | - | - | 0,347 | | | |
| **AGM** | | | x | x | - | x | - | v | 0,864 | 4 | 7 | 19 |
| | | | x | x | v | x | - | v | 0,703 | | | |
| | | | x | x | v | x | - | - | 0,338 | | | |
| **BNS** | | | | - | v | - | v | v | **0,886** | **14** | **7** | **49** |
| | | | | - | v | - | v | v | **0,726** | | | |
| | | | | - | v | - | v | v | **0,414** | | | |
| **Fast** | | | | | - | - | - | v | 0,869 | 12 | 8 | 44 |
| | | | | | v | - | v | v | 0,727 | | | |
| | | | | | v | - | v | v | 0,492 | | | |
| **F1** | | | | | | x | - | v | 0,855 | 3 | 3 | 12 |
| | | | | | | x | x | v | 0,68 | | | |
| | | | | | | x | x | v | 0,274 | | | |
| **Hell** | | | | | | | v | v | **0,872** | **14** | **7** | **49** |
| | | | | | | | v | v | **0,729** | | | |
| | | | | | | | v | v | **0,46** | | | |
| **MCC** | | | | | | | | v | 0,866 | 4 | 8 | 20 |
| | | | | | | | | v | 0,702 | | | |
| | | | | | | | | - | 0,344 | | | |
| **PRC** | | | | | | | | | 0,626 | 0 | 3 | 3 |
| | | | | | | | | | 0,632 | | | |
| | | | | | | | | | 0,239 | | | |

Tabela 7.13: Comparativo da classificação realizada por **Regressão Logística** sobre técnicas de seleção de atributos com métricas insensíveis ao desbalanceamento entre as classes. A referência de comparação deve ser sempre a linha.

resultados obtidos pelas técnicas de seleção de atributos *CFS*, *GainRatio* e *Relief*, as quais são consideradas sensíveis ao desbalanceamento entre as classes. A técnica de árvore de decisão foi omitida dessa tabela, por limitação de espaço, após terem obtido os piores resultados.

Através da Tabela 7.15, é possível perceber que a estratégia de seleção de atributos pela fronteira de Pareto obteve bons resultados, principalmente quando utilizada a técnica de classificação de *Redes Bayesianas*. Para *regressão logística* essa técnica ficou estatisticamente entre as melhores técnicas.

Podemos notar que a técnica de seleção de atributos $Relief$, embora seja uma técnica sensível ao desbalanceamento entre as classes, obteve resultados competitivos com as técnicas que utilizam métricas insensíveis. Esse resultado contraria o comportamento observado em [Chen & Wasikowski, 2008], onde a técnica $Fast$ superava a técnica $Relief$ ao selecionar atributos para um cenário de desbalanceamento entre as classes.

Uma possível explicação para esse comportamento é a natureza dos problemas.

| | Técnicas de Seleção de Atributos | | | | | | | Score | V | E | Ptos |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | **ACC2** | **AGM** | **BNS** | **Fast** | **F1** | **Hell** | **MCC** | **PRC** | | | | |
| **ACC2** | | x | x | x | - | x | - | x | 0,627 | 1 | 5 | 8 |
| | | x | x | x | v | x | x | x | 0,653 | | | |
| | | - | x | x | - | x | - | x | 0,255 | | | |
| **AGM** | | | x | x | v | x | - | x | 0,718 | 7 | 4 | 25 |
| | | | x | x | v | x | - | v | 0,678 | | | |
| | | | x | x | v | x | - | v | 0,308 | | | |
| **BNS** | | | | - | v | - | v | x | **0,735** | **14** | **6** | **48** |
| | | | | - | v | - | v | v | **0,689** | | | |
| | | | | - | v | - | v | v | **0,393** | | | |
| **Fast** | | | | | v | - | v | - | **0,757** | **14** | **7** | **49** |
| | | | | | v | - | v | v | **0,697** | | | |
| | | | | | v | - | v | v | **0,357** | | | |
| **F1** | | | | | | x | x | x | 0,646 | 0 | 2 | 2 |
| | | | | | | x | x | x | 0,641 | | | |
| | | | | | | x | x | x | 0,231 | | | |
| **Hell** | | | | | | | v | v | **0,793** | **15** | **6** | **51** |
| | | | | | | | v | v | **0,702** | | | |
| | | | | | | | v | v | **0,366** | | | |
| **MCC** | | | | | | | | x | 0,716 | 5 | 5 | 20 |
| | | | | | | | | v | 0,687 | | | |
| | | | | | | | | x | 0,273 | | | |
| **PRC** | | | | | | | | | 0,777 | 10 | 1 | 31 |
| | | | | | | | | | 0,673 | | | |
| | | | | | | | | | 0,284 | | | |

Tabela 7.14: Comparativo da classificação por **Árvore de Decisão** sobre técnicas de seleção de atributos com métricas insensíveis ao desbalanceamento entre as classes. A referência de comparação deve ser sempre a linha.

| | **Bayes** | | | **Logistic** | | |
|---|---|---|---|---|---|---|
| | **AUC** | **F1** | **EE** | **AUC** | **F1** | **EE** |
| **BNS** | 0,865 ± 0,009 | 0,736 ± 0,004 | 0,486 ± 0,041 | 0,872 ± 0,006 | **0,726** ± 0,009 | **0,468** ± 0,05 |
| **Fast** | 0,87 ± 0,012 | 0,727 ± 0,008 | 0,481 ± 0,044 | 0,867 ± 0,012 | **0,727** ± 0,009 | **0,487** ± 0,047 |
| **Hell** | 0,864 ± 0,008 | 0,742 ± 0,006 | 0,492 ± 0,043 | 0,870 ± 0,006 | **0,726** ± 0,011 | **0,460** ± 0,046 |
| **CFS** | 0,903 ± 0,005 | 0,728 ± 0,007 | 0,390± 0,05 | 0,873 ± 0,008 | 0,705 ± 0,01 | 0,369 ± 0,042 |
| **GR** | 0,777 ± 0,009 | 0,677 ± 0,006 | 0,296 ± 0,037 | 0,767 ± 0,013 | 0,671 ± 0,007 | 0,294 ± 0,032 |
| **Relief** | 0,905 ± 0,007 | 0,735 ± 0,003 | 0,465 ± 0,026 | **0,907** ± 0,011 | **0,727** ± 0,015 | **0,444** ± 0,067 |
| **Pareto** | **0,916**± 0,006 | **0,741** ± 0,005 | **0,508** ± 0,031 | **0,902** ± 0,06 | **0,724** ± 0,007 | **0,460** ± 0,055 |

Tabela 7.15: Modelos de Detecção de Fraude utilizando técnicas de seleção de atributos voltadas para dados desbalanceados.

Os cenários analisados anteriormente não tratavam de um problema de detecção de anomalias, onde o desbalanceamento entre as classes é muito maior do que um cenário comum de classificação em dados desbalanceados.

Todavia, as técnicas também sensíveis ao desbalanceamento entre as classes *GainRatio* e $CFS$ foram superadas pelas técnicas que utilizam métricas para dados desbalanceados. Mostrando mais uma vez serem mais sensíveis ao desbalanceamento entre as classes nesse cenário do que a técnica $Relief$.

Por fim, através da abordagem de fronteira de Pareto para selecionar os atributos, construímos o melhor modelo de detecção de fraude, sem a utilização de *resampling*. O modelo utilizando a técnica de *Redes Bayesianas*, sobre um conjunto de dados selecionados através da estratégia de fronteira de Pareto, alcançou um AUC de 0.916 e ganhos econômicos de aproximadamente 51% sobre o cenário real da companhia.

## 7.2 Estudo de Caso 2 - Detecção de Fraude utilizando Dados do Sistema de Pagamento Eletrônico Moip

No segundo estudo de caso, utilizamos a base de dados real do sistema de pagamento eletrônico Moip[2] para avaliar nossa metodologia. A Moip é uma das principais empresas de sistema de pagamento eletrônico do Brasil. A empresa brasileira atua no mercado de pagamentos *online* com a oferta de soluções para *e-commerce*, *marketplaces*, aplicativos e *mobile*. Em 2016 a *Moip* foi vendida para uma empresa alemã, chamada Wirecard [3].

Cabe ressaltar que esse estudo de caso é um trabalho inicial, devido à recém parceria firmada com a empresa Moip. Assim, os resultados para esse estudo de caso são uma complementação das análises realizadas no estudo de caso anterior.

### 7.2.1 Caracterização da base de dados

A base de dados fornecida pela companhia apresenta milhões de transações ocorridas entre os meses de agosto a ovembro de 2015. Do mesmo modo que na descrição da base anterior, cada uma dessas transações contém atributos dos mais variados tipos. Além desses atributos, existe um atributo especial que descreve se uma transação foi fraudulenta ou não, através do processo de *chargeback*, já explicado anteriormente.

O alto desbalanceamento entre as classes também é uma característica marcante nessa base de dados, que classifica esse cenário como um problema de detecção de anomalias. Devido ao acordo de confidencialidade com a companhia, não podemos publicar informações quantitativas sobre os dados e seus atributos, entretanto podemos garantir que esses são suficientemente representativos para esta pesquisa.

[2] http://moip.com.br
[3] www.wirecard.com

### 7.2.2 Resultados Experimentais

Devido ao acordo de confidencialidade, nesse estudo de caso não abordaremos valores absolutos de nenhuma métrica. Os resultados serão apresentados em ganhos percentuais sobre os *baselines*. Mesmo assim, é possível, através desses resultados, analisar os comportamentos desejados e ratificar nossas hipóteses de pesquisa. Do mesmo modo que no estudo de Caso da Seção 7.1.2, dividiremos nossos resultados entre as duas abordagens utilizadas para melhorar a eficácia da seleção de atributos para detectar anomalias.

A Subseção 7.2.2.1 apresenta os resultados encontrados para aplicação de métodos de *resampling* antes da seleção de atributos. Já na Subseção 7.2.2.2 são apresentados os resultados obtidos para aplicação dos métodos de seleção de atributos considerados insensíveis ao desbalanceamento entre as classes.

#### 7.2.2.1 Abordagem 1 - Aplicação de Métodos de *Resampling* para Reduzir o Desbalanceamento entre as Classes Antes da Seleção de Atributos

Cada técnica de seleção de atributos aplicada sobre métodos de *resampling*, com diferentes proporções de fraude, geraram diferentes conjuntos de atributos selecionados. A Tabela 7.16 apresenta o percentual de atributos gerados sobre o total disponível para cada estratégia de seleção de atributos, utilizando *resampling*, com diferentes proporções ou a proporção real de fraude.

| | **NM-1** | | | **NM-2** | | | **NM-3** | | | **ROS** | | | **RUS** | | | **Smote** | | | **SO** | **Real** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | □ | △ | ○ | □ | △ | ○ | □ | △ | ○ | □ | △ | ○ | □ | △ | ○ | □ | △ | ○ | □ △ ○ | □ △ ○ |
| % | | | | | | | | | | | | | | | | | | | 14 20 26 | 11 6 26 |
| **5** | 22 | 22 | 34 | 19 | 22 | 42 | 14 | 16 | 38 | 10 | 4 | 32 | 21 | 4 | 36 | 11 | 12 | 32 | | |
| **10** | 25 | 33 | 36 | 22 | 26 | 41 | 14 | 16 | 38 | 11 | 12 | 30 | 15 | 8 | 34 | 14 | 12 | 32 | | |
| **15** | 29 | 41 | 36 | 19 | 32 | 10 | 14 | 16 | 38 | 11 | 12 | 32 | 12 | 26 | 32 | 16 | 12 | 30 | | |
| **20** | 26 | 45 | 33 | 18 | 32 | 14 | 14 | 16 | 38 | 10 | 11 | 32 | 12 | 12 | 32 | 15 | 11 | 29 | | |
| **25** | 25 | 47 | 37 | 15 | 32 | 25 | 14 | 16 | 38 | 8 | 27 | 27 | 14 | 5 | 37 | 15 | 11 | 26 | | |
| **30** | 25 | 33 | 30 | 15 | 33 | 21 | 18 | 15 | 37 | 8 | 27 | 27 | 15 | 29 | 40 | 15 | 11 | 25 | | |
| **40** | 16 | 8 | 41 | 18 | 42 | 27 | 15 | 15 | 27 | 8 | 27 | 27 | 14 | 16 | 27 | 14 | 19 | 27 | | |
| **50** | 16 | 11 | 25 | 15 | 53 | 22 | 18 | 16 | 22 | 7 | 26 | 26 | 12 | 19 | 23 | 11 | 19 | 26 | | |

Tabela 7.16: Estudo de Caso 2: Porcentagem de atributos selecionados sobre o total disponível por cada estratégia de seleção de atributos, utilizando *resampling* ou mantendo a proporção real de fraudes. Legenda: □:CFS; △:GainRatio; ○:Relief

Podemos notar na Tabela 7.16 que qualquer técnica de seleção de atributos, quando aplicada sobre o conjunto com *resampling*, utilizando o método *NearMiss 3* ($NM-3$), gerou o mesmo número de atributos para a faixa entre 5% e 25% de fraudes. Esse comportamento foi explicado no estudo de caso da Seção 7.1.2, onde foram gerados os mesmos conjuntos mas para as proporções de fraude entre 5 e 30%.

A mesma metodologia utilizada no estudo de caso anterior para encontrar a melhor porcentagem de fraude foi utilizada novamente. A Tabela 7.17 apresenta as melhores proporções de fraude e o ganho percentual obtido por cada uma das estratégias de seleção de atributos utilizada. Novamente a técnica de árvore de decisão obteve resultados muito inferiores as demais técnicas de classificação. Assim, na Tabela 7.17 são apresentados os resultados para a classificação feita com as técnicas redes bayesianas e regressão logística.

| | | | **Bayes** | | | **Logistic** | | |
|---|---|---|---|---|---|---|---|---|
| | | % | AUC | F1 | EE | AUC | F1 | EE |
| **CFS** | NM-1 | 25 | 0,96 | 0,38 | 123,08 | 1,26 | 0,38 | 3,39 |
| | NM-2 | 5 | -0,41 | 1,14 | 53,85 | -1,26 | 0,00 | -0,11 |
| | NM-3 | 5:25 | 3,29 | 3,04 | 61,54 | 3,37 | 3,07 | 135,59 |
| | ROS | 10 | -5,34 | -0,76 | -230,77 | -3,79 | -1,72 | -122,88 |
| | RUS | 5 | **2,47** | **3,98** | **384,62** | **3,37** | **3,64** | **162,71** |
| | Smote | 15 | -0,68 | 0,38 | 7,69 | -0,84 | -1,15 | 9,32 |
| | SO | X | 0,55 | 0,95 | 23,08 | 0,28 | 0,00 | 34,75 |
| **Gain Ratio** | NM-1 | 25 | **8,78** | **6,04** | **107,58** | **7,61** | **2,53** | **9,52** |
| | NM-2 | 40 | 10,27 | 6,63 | 104,55 | 9,70 | 2,73 | -0,073 |
| | NM-3 | 50 | 6,99 | 3,51 | 86,36 | 4,48 | 0,19 | 22,22 |
| | ROS | 50 | 8,78 | 3,12 | 101,52 | 7,61 | 1,17 | 38,10 |
| | RUS | 40 | **8,63** | **3,31** | **109,85** | **6,57** | **1,36** | **28,57** |
| | Smote | 50 | 7,44 | 3,31 | 88,64 | 4,63 | 0,58 | 9,84 |
| | SO | X | 5,80 | 1,95 | 23,48 | 6,12 | 1,95 | 19,84 |
| **Relief** | NM-1 | 10 | 0,00 | 0,36 | 3,70 | 0,14 | 0,18 | 1,37 |
| | NM-2 | 5 | **11,39** | **0,00** | **1,23** | 0,28 | 0,00 | 0,08 |
| | NM-3 | 5:25 | 0,00 | -1,43 | -11,11 | **0,14** | **0,18** | **1,37** |
| | ROS | 5 | 0,94 | 0,00 | 0,00 | 0,00 | -0,36 | -15,07 |
| | RUS | 10 | **1,09** | **1,61** | **2,47** | 0,00 | 0,00 | 0,00 |
| | Smote | 10 | 0,94 | -1,25 | -38,27 | 0,00 | -0,36 | -15,07 |
| | SO | X | -1,40 | 0,00 | -7,41 | 0,00 | 0,00 | 0,00 |

Tabela 7.17: Estudo de Caso 2: Ganho percentual na detecção de fraude utilizando *resampling* antes da seleção de atributos, sobre as mesmas técnicas utilizando a proporção real de fraudes antes da etapa de seleção de atributos.

Na Tabela 7.17 destacamos em **negrito** os maiores ganhos obtidos. É possível notar nessa tabela que alguns comportamentos observados no Estudo de Caso 1 também foram observados nesse estudo de caso. Novamente as proporções de fraudes não foram linearmente correlacionadas com os melhores resultados. As melhores proporções de fraudes variaram de acordo com as estratégias de seleção de atributos escolhidas.

Da mesma forma que no estudo de caso da Seção 7.1.2, podemos perceber que as técnicas de seleção de atributos *Relief* e *CFS* foram menos sensíveis ao desbalanceamento entre as classes. Nessas técnicas, a melhor proporção de fraude para cada método de *resampling* foi menor que a técnica *GainRatio*. A técnica *GainRatio* se mostrou a mais sensível ao desbalanceamento entre as classes. Nesta técnica, qualquer método de *resampling*, aplicado antes da seleção de atributos, melhorou os resultados da seleção de atributos.

A técnica *CFS* se mostrou sensível ao desbalanceamento principalmente ao medir

a Eficiência Econômica. Nessa técnica a grande maioria dos métodos de *resampling* aplicados antes da seleção de atributos, também melhorou os resultados. Já a técnica *Relief*, se mostrou a menos sensível ao desbalanceamento dentre as três técnicas. Ainda assim, utilizando alguns métodos de *resampling* antes da seleção de atributos feitas pela técnica *Relief* nesse estudo de caso, foi possível construir modelos mais eficazes.

O método de *resampling Random Undersampling* ($RUS$) superou o método *Random Overrsampling* ($ROS$) para reduzir o desbalanceamento entre as classes antes da seleção de atributos. Esse comportamento já havia sido observado no Estudo de Caso 1. Em linhas gerais, 3 métodos de *resampling* se destacaram para reduzir o desbalanceamento entre as classes antes da seleção de atributos. Foram eles, os métodos $RUS$, $NM-1$ e $NM-2$, sendo o primeiro o que alcançou os melhores resultados.

Nesse estudo de caso também utilizamos a estratégia $Merge$, que combina os melhores atributos, e o método $No\_FS$, que utiliza todos os atributos disponíveis. O método $Merge$ não obteve melhoras significativas que justificasse o seu uso. O método $No\_FS$ foi superado por todos os modelos, inclusive pelos modelos que utilizam as técnicas de seleção de atributos sobre a proporção real de fraude, mostrando a necessidade da seleção de atributos também nesse segundo cenário.

Em geral, foi possível perceber que, para qualquer técnica de seleção de atributos, a aplicação de algum método de *resampling*, com a proporção adequada de fraude melhorou a eficácia da seleção de atributos para detectar fraudes. Mostrando, mais uma vez, que essa pode ser uma estratégia adequada para ser utilizada na construção de modelos de detecção de fraude.

A Subseção 7.2.2.2 apresenta os resultados obtidos através da aplicação de métodos da segunda abordagem de seleção de atributos, sobre a base de dados do nosso segundo estudo de caso.

#### 7.2.2.2 Abordagem 2 - Aplicação de Técnicas de Seleção de Atributos Insensíveis ao Desbalanceamento entre Classes

Do mesmo modo que no estudo de caso 1 (Seção 7.1.2.2), a segunda abordagem de seleção de atributos consiste na análise de métodos considerados insensíveis ao desbalanceamento entre as classes. Foram utilizadas os métodos $Fast$, $Hell$ e $TBFS$, o qual utilizou as métricas $ACC2, AGM, BNS, F1, MCC$ e $PRC$, gerando 6 diferentes técnicas. As Tabelas 7.18 e 7.19 apresentam uma matriz de resultados comparativos entre essas técnicas.

Nessa tabela, para cada método representada por uma linha, apresentamos a comparação par a par entre os demais métodos (representados na coluna) e a sua

performance em relação a esse. Assim, se um método (analisando pela linha) venceu o método da coluna em uma determinada métrica é atribuído v para aquela célula, em casa de derrota é atribuído x, e − se as duas técnicas forem estatisticamente iguais. Conforme explicado anteriormente, novamente atribuímos pontuação para as técnicas de acordo com o número de vitórias (3 pontos) e empates (1 ponto).

| | AGM | BNS | Fast | F1 | Hell | MCC | PRC | v | E | Ptos |
|---|---|---|---|---|---|---|---|---|---|---|
| ACC | v | v | v | v | v | v | v | 13 | 3 | **42** |
| | x | - | - | v | - | x | v | | | |
| | x | x | v | v | v | x | v | | | |
| AGM | | x | x | x | x | x | x | 14 | 0 | **42** |
| | | v | v | v | v | v | v | | | |
| | | v | v | v | v | v | v | | | |
| BNS | | | v | v | x | v | v | 12 | 3 | **39** |
| | | | - | v | - | x | v | | | |
| | | | v | v | v | x | v | | | |
| Fast | | | | v | x | x | v | 8 | 3 | 27 |
| | | | | v | - | x | v | | | |
| | | | | v | v | x | v | | | |
| F1 | | | | | x | x | - | 1 | 3 | 6 |
| | | | | | x | x | - | | | |
| | | | | | x | x | - | | | |
| Hell | | | | | | v | v | 10 | 3 | 33 |
| | | | | | | x | v | | | |
| | | | | | | x | v | | | |
| MCC | | | | | | | v | 16 | 0 | **48** |
| | | | | | | | v | | | |
| | | | | | | | v | | | |
| PRC | | | | | | | | 1 | 3 | 6 |

Tabela 7.18: Estudo de Caso 2 - Comparativo da classificação por **Redes Bayesianas** sobre técnicas de seleção de atributos com métricas insensíveis ao desbalanceamento entre as classes. A referência de comparação é a linha, foi utilizado v para vitória, x para derrota e − se as duas técnicas forem estatisticamente iguais, utilizando o teste pareado de Wilcoxon.

Destacamos nas Tabelas 7.18 e 7.19 os melhores resultados em **negrito**. Podemos perceber que o método $TBFS$, utilizando $F1$ e $PRC$, novamente alcançou os piores resultados. Esse comportamento já havia sido observado no estudo de caso 1. Os melhores atributos para classificação foram selecionados pelo método $TBFS$, utilizando as métricas $AGM$ e $MCC$.

Para tentar combinar as vantagens de algumas métricas de seleção de atributos, novamente utilizamos a fronteira de Pareto. As dimensões da fronteira foram definidas com bases nos resultados individuais de cada métrica. Assim, essas dimensões foram formadas pelas métricas com pontuação em **negrito** nas Tabelas 7.18 e 7.19.

A Tabela 7.20 apresenta o ganho percentual do método de seleção de atributos fronteira de Pareto, comparados com os melhores métodos de seleção considerados

| | AGM | BNS | Fast | F1 | Hell | MCC | PRC | v | E | Ptos |
|---|---|---|---|---|---|---|---|---|---|---|
| **ACC** | x | v | v | v | - | x | v | 10 | 1 | 31 |
| | x | x | v | v | x | x | v | | | |
| | x | x | v | v | x | x | v | | | |
| **AGM** | | v | v | v | v | v | v | 21 | 0 | **63** |
| | | v | v | v | v | v | v | | | |
| | | v | v | v | v | v | v | | | |
| **BNS** | | | v | v | x | x | v | 12 | 1 | **37** |
| | | | v | v | - | x | v | | | |
| | | | v | v | v | x | v | | | |
| **Fast** | | | | - | x | x | - | 2 | 4 | 10 |
| | | | | - | x | x | - | | | |
| | | | | v | x | x | v | | | |
| **F1** | | | | | x | x | - | 0 | 5 | 5 |
| | | | | | x | x | - | | | |
| | | | | | x | x | - | | | |
| **Hell** | | | | | | x | v | 12 | 2 | **38** |
| | | | | | | x | v | | | |
| | | | | | | x | v | | | |
| **MCC** | | | | | | | v | 18 | 0 | **54** |
| | | | | | | | v | | | |
| | | | | | | | v | | | |
| **PRC** | | | | | | | | 0 | 5 | 5 |

Tabela 7.19: Estudo de Caso 2 - Comparativo da classificação por **Regressão Logística** sobre técnicas de seleção de atributos com métricas insensíveis ao desbalanceamento entre as classes. A referência de comparação é a linha, foi utilizado v para vitória, x para derrota e − se as duas técnicas forem estatisticamente iguais, utilizando o teste pareado de Wilcoxon.

insensíveis ao desbalanceamento entre as classes. A classificação desses modelos foi feita pelas técnicas redes bayesianas e regressão logística. Para avaliar os resultados foram utilizadas as métricas $AUC$, $F1$ e $EE$.

| | **Bayes** | | | **Logistic** | | |
|---|---|---|---|---|---|---|
| | AUC | F1 | EE | AUC | F1 | EE |
| **ACC** | 1,83 | 4,91 | 47,10 | 3,25 | 5,62 | 43,84 |
| **AGM** | 18,52 | 0,36 | 6,54 | 0,00 | -0,91 | 0,00 |
| **MCC** | 6,48 | 1,65 | 21,28 | 0,27 | 0,00 | 5,53 |

Tabela 7.20: Estudo de Caso 2 - Ganho Percentual do método de seleção de atributos **fronteira de Pareto** sobre os melhores métodos utilizando a proporção real de fraudes na seleção de atributos.

É possível notar na Tabela 7.20 que, em geral, a estratégia de fronteira de Pareto obteve melhores resultados que qualquer métodos de seleção de atributos utilizado individualmente. Alcançando ganhos econômicos 6,54% maiores que o melhor método de seleção de atributos considerado insensível ao desbalanceamento entre as classes.

Os gráficos da Figura 7.6 ilustram os resultados obtidos pelos melhores modelos de seleção de atributos apresentados na Tabela 7.20. Para melhorar a comparação foram adicionados nos gráficos os modelos que utilizam as 3 técnicas de seleção de atributos consideradas sensíveis ao desbalanceamento entre as classes (CFS, GainRatio e Relief).

Cabe ressaltar que devido ao acordo de confidencialidade com a empresa fornecedora dos dados, os gráficos não mostram a escala numérica dos resultados. Entretanto, ainda assim, nesses gráficos é possível realizar uma comparação da eficácia relativa obtida por cada modelo.

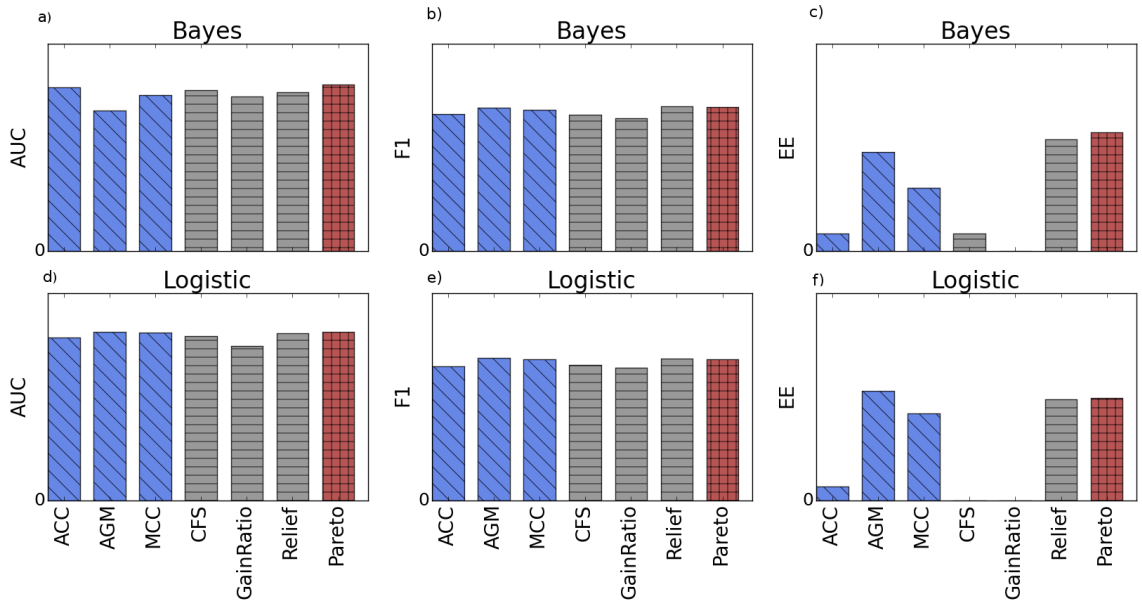


Figura 7.6: Estudo de Caso 2 - Modelos de detecção de anomalias utilizando diferentes estratégias para seleção de atributos sobre a proporção real entre as classes.

Através dos gráficos da Figura 7.6 é possível identificar uma leve melhora nos resultados utilizando a fronteira de Pareto. Os métodos $TBFS$, utilizando as métricas insensíveis ao desbalanceamento entre as classes $ACC$ e $AGM$ apresentaram bons desempenhos, mas sobre diferentes perspectivas. O método utilizando $ACC$ alcançou bons resultados, ao se avaliar o $AUC$, mas não obteve a mesma performance ao observar a $EE$. Enquanto o método, utilizando $AGM$ obteve comportamento inverso.

Esse comportamento pode ser mais uma vez explicado pela natureza das métricas $AUC$ e $EE$, as quais foram explicadas anteriormente. Assim, podemos concluir que o método utilizando $AGM$ obteve boa revocação para detectar fraudes, mas obteve alta taxa de falso positivo. Já o método utilizando $ACC$, obteve baixa revocação, mas alta taxa de falso positivo. Portanto, unindo as vantagens de cada métrica, o método de fronteira de Pareto se mostrou mais eficaz.

Outro comportamento novamente observado é em relação ao método *Relief*, que apesar de ser um método que utiliza uma métrica sensível ao desbalanceamento entre as classes, alcançou resultados melhores que métodos que utilizam métricas insensíveis ao desbalanceamento entre classes, com exceção da combinação de métricas através da fronteira de Pareto. Esses comportamentos já haviam sido observados no estudo de caso 1. Assim, confirmamos a hipótese de que a combinação de métricas através da fronteira de Pareto pode ser uma abordagem útil para selecionar atributos na detecção de anomalias.

No Capítulo 8 apresentamos as principais conclusões obtidas através dos resultados experimentais e direções para trabalhos futuros.

# Capítulo 8

# Conclusões e Trabalhos Futuros

Neste trabalho foram analisadas estratégias de seleção de atributos para detecção de anomalias em transações eletrônicas. Para realizar essas análises, dividimos o trabalho em duas abordagens principais. Na primeira abordagem reduzimos o desbalanceamento entre as classes antes da seleção de atributos através de 7 métodos distintos, incluindo um criado nesse trabalho, de *resampling* com diferentes proporções de fraude. Para selecionar os atributos sobre a base de dados com as novas proporções utilizamos 3 diferentes técnicas de seleção de atributos, consideradas sensíveis ao desbalanceamento entre as classes.

Já na segunda abordagem utilizamos 8 métodos de seleção de atributos considerados insensíveis ao desbalanceamento entre as classes, além da criação de um método que utiliza o conceito de fronteirade Pareto para combinação das métricas. Nessa etapa toda a seleção foi realizada sobre a base de dados com a proporção real entre as classes positiva e negativa.

A validação sobre a eficácia das duas abordagens foi realizada através da construção de modelos de detecção de fraudes, que consistem na aplicação de 3 diferentes técnicas de classificação sobre os atributos selecionados pelas distintas estratégias. Para avaliação desses modelos utilizamos 3 métricas adequadas para cenários de alto desbalanceamento entre as classes. Entre elas, inclui-se uma métrica que criamos para avaliar a Eficiência Econômica dos modelos.

Toda a metodologia desse trabalho foi avaliada sobre dois estudos de caso para detecção de fraudes em cenários reais de dois sistemas de pagamento eletrônico. No primeiro estudo de caso, avaliamos a seleção de atributos nos dados fornecidos pelo sistema de pagamento eletrônico *Uol PagSeguro*. Já no segundo estudo de caso, realizamos experimentos sobre dados reais fornecidos pelo sistema de pagamento eletrônico *Moip*.

Através dos experimentos realizados foi possível validar nossas hipóteses de pesquisa, endereçar questões relevantes e trazer contribuições para estratégias de seleção de atributos para detectar fraudes. As principais conclusões obtidas foram:

- **O alto desbalanceamento entre as classes impactou diretamente a seleção de atributos** nos dois cenários de detecção de anomalias.
- Os 3 **métodos tradicionais de seleção de atributos não foram adequados para serem utilizados no cenário de detecção de fraudes**. Dentre os 3 métodos, o método *Relief* se mostrou menos sensível ao desbalanceamento entre as classes e o método *GainRatio* o mais sensível.
- **O mérito do conjunto de atributos**, calculado sem a utilização da classificação, **não foi um bom indicador da performance da seleção de atributos** em um cenário de detecção de anomalias. Assim, para avaliação de um subconjunto de atributos foi necessário a utilização de um classificador.
- A aplicação de **métodos de *resampling* antes da seleção de atributos melhorou a performance das técnicas de seleção de atributos**, e consequentemente, dos modelos de detecção de fraude.
- **A proporção de fraude em um método de *resampling* não é diretamente correlacionada com a eficácia da seleção de atributos**, isto é, aumentando-se a proporção de fraude, não se aumenta proporcionalmente a eficácia da seleção de atributos.
- **Não existe uma proporção de fraude ótima comum para todos os métodos de *resampling***. Cada método de *resampling* se adaptou melhor a uma proporção de fraude.
- O método ***Sampling Outlier***, que criamos nesse trabalho, o qual não tem a necessidade de se informar a taxa de *resampling*, **mostrou-se uma boa alternativa quando o custo de realizar uma classificação para escolher a melhor proporção é alto**.
- Os métodos mais inteligentes e complexos de *resampling* não ofereceram melhoras significativas. O método de *resampling* ***Random Undersampling***, apesar de ser um método randômico simples, **obteve boa performance ao aumentar-se a proporção de fraude** para seleção de atributos.
- **A combinação de atributos dos melhores subconjuntos para detectar fraudes, podem gerar subconjuntos de atributos ainda mais eficazes para alguns cenários de aplicação**. Entretanto, devido a complexidade da solução esse método é

indicado apenas quando o tempo computacional para execução de modelos de classificação para avaliar os modelos não seja alto.

- **É possível construir modelos para detecção de fraude mais eficazes, utilizando técnicas de *resampling* para aumentar a proporção de fraude antes da seleção de atributos**, combinadas com técnicas de classificação.
- **O alto desbalanceamento entre as classes** em um cenário de detecção de anomalias **também impactaram as técnicas consideradas insensíveis ao desbalanceamento entre as classes**.
- A estratégia de **fronteirade Pareto**, utilizada para combinar métricas insensíveis ao desbalanceamento, **obteve melhores resultados** para selecionar atributos, sem alterar a distribuição inicial entre as classes, ou seja, **utilizando a proporção real**.

Podemos concluir que a seleção de atributos para detectar anomalias é um cenário atípico. As observações destacadas acima são essenciais para caracterizar e prover insumos na construção de estratégias adequadas para selecionar atributos nesses cenários. Através das estratégias utilizadas nesse trabalho, construímos modelos para detecção de fraude com ganhos econômicos de até 57% sobre o cenário real.

Alguns dos resultados e análises deste trabalho geraram, até o presente momento 3 publicações científicas, a saber:

- Artigo intitulado Modelos Computacionais baseados em Feature Selection e Undersampling para Detecção de Fraudes Eletrônicas, publicado no XXX Simpósio Brasileiro de Banco de Dados, SBBD 2015 [Lima & Pereira, 2015b].
- Artigo intitulado A fraud detection model based on feature selection and undersampling applied to Web payment systems , publicado no International Workshop on Ensemble of Anomaly Detection Algorithms (EADA), em conjunto com Web Intelligence Conference, (Wi 2015) [Lima & Pereira, 2015a].
- Artigo intitulado Feature Selection Approaches to Fraud Detection in e-Payment Systems, aceito para publicação na 17th International Conference on Electronic Commerce and Web Technologies (EC-Web 2016).

Além das publicações alcançadas, pretende-se enviar, em breve, os resultados finais obtidos e discussões desse trabalho para um periódico internacional. Essa publicação visa consolidar os experimentos, fundamentos e análises abordados nesta dissertação, tornando uma importante referência para seleção de atributos na detecção de fraudes em transações eletrônicas.

Através desta pesquisa confirmou-se a necessidade de prover mecanismos para aumentar a eficácia da seleção de atributos para detectar anomalias. Embora, com as estratégias utilizadas obtivemos bons resultados, acreditamos que é possível oferecer estratégias ainda mais efetivas. Assim, na Seção 8.1 apresentamos alguns possíveis trabalhos futuros, como extensão dessa pesquisa.

## 8.1 Trabalhos Futuros

O trabalho realizado permitiu definir diretrizes para a criação de novas estratégias de seleção de atributos em cenários de detecção de anomalias. Assim, destacamos como possíveis trabalhos futuros:

- Proposta de técnicas de seleção de atributos adequadas especificamente para selecionar atributos em cenários anômalos. Através dos conceitos observados nesse trabalho, pretende-se utilizar os pontos fortes de métodos de seleção de atributos sensíveis ao desbalanceamento entre as classes, alterando o peso dos atributos da classe minoritária, adequando-os para detectar anomalias.

- Utilização de novas estratégias para combinação de métricas insensíveis ao desbalanceamento entre as classes. Nesse trabalho avaliamos a combinação de métricas através da fronteirade Pareto e mostramos que essa pode ser uma boa estratégia. Entretanto, outros métodos podem ser utilizadas. Entre esses pretende-se avaliar técnicas de *Rank Agregation* e Algoritmos Genéticos Multiobjetivos.

- Utilização de métodos de *Learning to Rank* para selecionar atributos. Como os métodos de *ranking* proveram os melhores resultados para seleção de atributos, a transformação do problema em um cenário de *Learning to Rank* pode oferecer bons resultados.

- Avaliação de outros cenários para detecção de anomalias. Embora, cada cenário tenha suas peculiaridades, acreditamos que muitos comportamentos observados nesse trabalho podem ser aplicados em outros cenários de detecção de anomalias.

# Referências Bibliográficas

[Ahmed et al., 2016] Ahmed, M.; Mahmood, A. N. & Islam, M. R. (2016). A survey of anomaly detection techniques in financial domain. *Future Generation Computer Systems*, 55:278--288.

[Aleskerov et al., 1997] Aleskerov, E.; Freisleben, B. & Rao, B. (1997). Cardwatch: a neural network based database mining system for credit card fraud detection. Em *Computational Intelligence for Financial Engineering (CIFEr), 1997., Proceedings of the IEEE/IAFE 1997*, pp. 220–226.

[Alibeigi et al., 2012] Alibeigi, M.; Hashemi, S. & Hamzeh, A. (2012). Dbfs: An effective density based feature selection scheme for small sample size and high dimensional imbalanced data sets. *Data & Knowledge Engineering*, 81:67--103.

[Almendra, 2013] Almendra, V. (2013). Finding the needle: A risk-based ranking of product listings at online auction sites for non-delivery fraud prediction. *Expert Systems with Applications*, 40(12):4805 – 4811. ISSN 0957-4174.

[Arif et al., 2015] Arif, M.; Alam, K. A. & Hussain, M. (2015). Crime mining: A comprehensive survey. *International Journal of u-and e-Service, Science and Technology*, 8(2):357--364.

[Batuwita & Palade, 2012] Batuwita, R. & Palade, V. (2012). Adjusted geometric-mean: a novel performance measure for imbalanced bioinformatics datasets learning. *Journal of bioinformatics and computational biology*, 10(04):1250003.

[Benjamini & Hochberg, 1995] Benjamini, Y. & Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 289--300.

[Bhattacharyya et al., 2011] Bhattacharyya, S.; Jha, S.; Tharakunnel, K. & Westland, J. C. (2011). Data mining for creditcard fraud: A comparative study. *Journal Decision Support Systems.*, 50(3):602–613.

[Bittner, 1962] Bittner, L. (1962). R. bellman, adaptive control processes. a guided tour. *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik*, 42(7-8):364--365. ISSN 1521-4001.

[Bolton et al., 2001] Bolton, R. J.; Hand, D. J. & H, D. J. (2001). Unsupervised profiling methods for fraud detection. Em *Proc. Credit Scoring and Credit Control VII*, pp. 5--7.

[Börzsönyi et al., 2001] Börzsönyi, S.; Kossmann, D. & Stocker, K. (2001). The skyline operator. Em *Proceedings of the 17th International Conference on Data Engineering*, pp. 421--430, Washington, DC, USA. IEEE Computer Society.

[Brause et al., 1999] Brause, R.; Langsdorf, T. & Hepp, M. (1999). Neural data mining for credit card fraud detection. Em *Proceedings of the 11th IEEE International Conference on Tools with Artificial Intelligence*, ICTAI '99, pp. 103--, Washington, DC, USA. IEEE Computer Society.

[Cameron, 1997] Cameron, D. (1997). D. electronic commerce: The new business platform of the internet. *Charleston: Computer Technology Research Corp.*

[Cerqueira, 2010] Cerqueira, P. H. R. (2010). Um estudo sobre reconhecimento de padrões: um aprendizado supervisionado com classificador bayesiano. Dissertação de mestrado, Escola Superior de Agricultura Luiz de Queiroz, Universidade de São Paulo.

[Chandola et al., 2009] Chandola, V.; Banerjee, A. & Kumar, V. (2009). Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3):15:1--15:58. ISSN 0360-0300.

[Chawla, 2005] Chawla, N. V. (2005). Data mining for imbalanced datasets: An overview. Em *Data mining and knowledge discovery handbook*, pp. 853--867. Springer.

[Chawla et al., 2002] Chawla, N. V.; Bowyer, K. W.; Hall, L. O. & Kegelmeyer, W. P. (2002). Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, pp. 321--357.

[Chawla et al., 2004] Chawla, N. V.; Japkowicz, N. & Kotcz, A. (2004). Editorial: Special issue on learning from imbalanced data sets. *SIGKDD Explor. Newsl.*, 6(1):1--6. ISSN 1931-0145.

[Chen & Wasikowski, 2008] Chen, X.-w. & Wasikowski, M. (2008). Fast: a roc-based feature selection metric for small samples and imbalanced data classification pro-

blems. Em *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 124--132. ACM.

[Chiu et al., 2011] Chiu, C.; Ku, Y.; Lie, T. & Chen, Y. (2011). Internet auction fraud detection using social network analysis and classification tree approaches. *International Journal of Electronic Commerce.*, 15(3):123–147.

[Cieslak & Chawla, 2008] Cieslak, D. A. & Chawla, N. V. (2008). Learning decision trees for unbalanced data. Em *Machine learning and knowledge discovery in databases*, pp. 241--256. Springer.

[Cuaya et al., 2011] Cuaya, G.; Munoz-Meléndez, A. & Morales, E. F. (2011). A minority class feature selection method. Em *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, pp. 417--424. Springer.

[Dal Pozzolo et al., 2014] Dal Pozzolo, A.; Caelen, O.; Le Borgne, Y.-A.; Waterschoot, S. & Bontempi, G. (2014). Learned lessons in credit card fraud detection from a practitioner perspective. *Expert systems with applications*, 41(10):4915--4928.

[Demšar, 2006] Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, 7:1--30.

[Dobson, 1990] Dobson, A. J. (1990). *An Introduction to Generalized Linear Models.* London:Chapman and Hall.

[Drummond et al., 2003] Drummond, C.; Holte, R. C. et al. (2003). C4. 5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling. Em *Workshop on learning from imbalanced datasets II*, volume 11. Citeseer.

[Enderlein, 1987] Enderlein, G. (1987). Hawkins, d. m.: Identification of outliers. chapman and hall, london ,new york 1980, 188 s., 14, 50. *Biometrical Journal*, 29(2):198--198. ISSN 1521-4036.

[Forman, 2003] Forman, G. (2003). An extensive empirical study of feature selection metrics for text classification. *The Journal of machine learning research*, 3:1289--1305.

[Friedman, 1940] Friedman, M. (1940). A comparison of alternative tests of significance for the problem of m rankings. *The Annals of Mathematical Statistics*, 11(1):86--92.

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

[Godfrey et al., 2007] Godfrey, P.; Shipley, R. & Gryz, J. (2007). Algorithms and analyses for maximal vector computation. *The VLDB Journal—The International Journal on Very Large Data Bases*, 16(1):5--28.

[Goswami & Chakrabarti, 2014] Goswami, S. & Chakrabarti, A. (2014). Feature selection: A practitioner view. *International Journal of Information Technology and Computer Science (IJITCS)*, 6(11):66.

[Guyon & Elisseeff, 2003] Guyon, I. & Elisseeff, A. (2003). An introduction to variable and feature selection. *J. Mach. Learn. Res.*, 3:1157--1182. ISSN 1532-4435.

[Guyon et al., 2006] Guyon, I.; Gunn, S.; Nikravesh, M. & Zadeh, L. A. (2006). *Feature Extraction: Foundations and Applications (Studies in Fuzziness and Soft Computing)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA. ISBN 3540354875.

[Hall, 2000] Hall, M. A. (2000). Correlation-based feature selection for discrete and numeric class machine learning. Em *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, pp. 359--366, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[Hosmer, 2000] Hosmer, D. W. (2000). *Applied Logistic Regression*. Wiley, New York, 2nd edição.

[Jaccard, 1901] Jaccard, P. (1901). Étude comparative de la distribution florale dans une portion des Alpes et des Jura. *Bulletin del la Société Vaudoise des Sciences Naturelles*, 37:547--579.

[Kailath, 1967] Kailath, T. (1967). The divergence and bhattacharyya distance measures in signal selection. *Communication Technology, IEEE Transactions on*, 15(1):52--60.

[Kamal et al., 2010] Kamal, A. H. M.; Zhu, X.; Pandya, A.; Hsu, S. & Narayanan, R. (2010). Feature selection for datasets with imbalanced class distributions. *International Journal of Software Engineering and Knowledge Engineering*, 20(02):113–137.

[Keele, 2007] Keele, S. (2007). Guidelines for performing systematic literature reviews in software engineering. Em *Technical report, Ver. 2.3 EBSE Technical Report. EBSE*.

[Kim et al., 2013] Kim, K.; Choi, Y. & Park, J. (2013). Pricing fraud detection in online shopping malls using a finite mixture model. *Electronic Commerce Research and Applications*, 12(3):195 – 207. ISSN 1567-4223.

[Koller & Sahami, 1996] Koller, D. & Sahami, M. (1996). Toward optimal feature selection. Em Saitta, L., editor, *Proceedings of the Thirteenth International Conference on Machine Learning (ICML)*, pp. 284--292. Morgan Kaufmann Publishers.

[Lima & Pereira, 2012] Lima, R. A. F. & Pereira, A. C. M. (2012). Fraud detection in web transactions. Em *Proceedings of the 18th Brazilian symposium on Multimedia and the web*, pp. 273--280. ACM.

[Lima & Pereira, 2015a] Lima, R. F. & Pereira, A. C. (2015a). A fraud detection model based on feature selection and undersampling applied to web payment systems. Em *2015 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, volume 3, pp. 219--222. IEEE.

[Lima & Pereira, 2015b] Lima, R. F. & Pereira, A. C. (2015b). Modelos computacionais baseados em feature selection e undersampling para detecção de fraudes eletrônicas. Em *Anais do XXX Brazilian Simpósio Brasileiro de Banco de Dados (SBBD, 2015*, volume 30, pp. 87--92. ISSN 2316-5170.

[Liu & Motoda, 1998a] Liu, H. & Motoda, H. (1998a). *Feature Extraction, Construction and Selection: A Data Mining Perspective*. Kluwer Academic Publishers, Norwell, MA, USA. ISBN 0792381963.

[Liu & Motoda, 1998b] Liu, H. & Motoda, H. (1998b). *Feature Selection for Knowledge Discovery and Data Mining*. Kluwer Academic Publishers, Norwell, MA, USA. ISBN 079238198X.

[Liu & Motoda, 2007] Liu, H. & Motoda, H. (2007). *Computational methods of feature selection*. CRC Press.

[Liu et al., 2009] Liu, X.-Y.; Wu, J. & Zhou, Z.-H. (2009). Exploratory undersampling for class-imbalance learning. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 39(2):539--550.

[Maes et al., 2001] Maes, S.; karl Tuyls; Vanschoenwinkel, B. & Manderick, B. (2001). Credit card fraud detection using bayesian and neural networks. *Vrije Universiteir Brussel*.

[Mani & Zhang, 2003] Mani, I. & Zhang, I. (2003). knn approach to unbalanced data distributions: a case study involving information extraction. Em *Proceedings of workshop on learning from imbalanced datasets*.

[Ngai et al., 2011a] Ngai, E.; Hu, Y.; Wong, Y.; Chen, Y. & Sun, X. (2011a). The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature. *Decision Support Systems*, 50(3):559--569.

[Ngai et al., 2011b] Ngai, E. W. T.; Hu, Y.; Wong, Y. H.; Chen, Y. & Sun, X. (2011b). The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature. *Decis. Support Syst.*, 50(3):559--569. ISSN 0167-9236.

[Pant & Srivasta, 2015] Pant, H. & Srivasta, R. (2015). A survey on feature selection methods for imbalanced datasets. *International Journal of Computer Engineering and Applications*, IX(2).

[Pawar et al., 2014] Pawar, M. A. D.; Kalavadekar, P. N. & Tambe, M. S. N. (2014). A survey on outlier detection techniques for credit card fraud detection. *IOSR Journal of Computer Engineering*, 16(2):44--48.

[Piramuthu, 2004] Piramuthu, S. (2004). Evaluating feature selection methods for learning in data mining applications. *European Journal of Operational Research*, 156(2):483 – 494. ISSN 0377-2217.

[Quinlan, 1993] Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. ISBN 1-55860-238-0.

[Ravisankar et al., 2011] Ravisankar, P.; Ravi, V.; Rao, G. R. & Bose, I. (2011). Detection of financial statement fraud and feature selection using data mining techniques. *Decision Support Systems*, 50(2):491 – 500. ISSN 0167-9236.

[Richhariya & Singh, 2012] Richhariya, P. & Singh, P. K. (2012). Article: A survey on financial fraud detection methodologies. *International Journal of Computer Applications*, 45(22):15–22.

[Salzberg, 1994] Salzberg, S. (1994). C4.5: Programs for machine learning by j. ross quinlan. morgan kaufmann publishers, inc., 1993. *Machine Learning*, 16(3):235–240. ISSN 0885-6125.

[Shannon & Weaver, 1949] Shannon, C. E. & Weaver, W. (1949). The mathematical theory of communication (urbana, il).

[Tseng & Fogg, 1999] Tseng, S. & Fogg, B. J. (1999). Credibility and computing technology. *Communications of the ACM*, 42:41–44.

[Van Hulse et al., 2012] Van Hulse, J.; Khoshgoftaar, T. M.; Napolitano, A. & Wald, R. (2012). Threshold-based feature selection techniques for high-dimensional bioinformatics data. *Network modeling analysis in health informatics and bioinformatics*, 1(1-2):47--61.

[Yang & King, 2009] Yang, H. & King, I. (2009). Ensemble learning for imbalanced e-commerce transaction anomaly classification. Em Leung, C.-S.; Lee, M. & Chan, J. H., editores, *ICONIP (1)*, volume 5863 of *Lecture Notes in Computer Science*, pp. 866–874. Springer.

[Yang & Pedersen, 1997] Yang, Y. & Pedersen, J. O. (1997). A comparative study on feature selection in text categorization. Em *ICML*, volume 97, pp. 412--420.

[Yin et al., 2013] Yin, L.; Ge, Y.; Xiao, K.; Wang, X. & Quan, X. (2013). Feature selection for high-dimensional imbalanced data. *Neurocomputing*, 105:3--11.

[Zhang et al., 2013] Zhang, Y.; Bian, J. & Zhu, W. (2013). Trust fraud: A crucial challenge for china e-commerce market. *Electronic Commerce Research and Applications*, 12(5):299 – 308. ISSN 1567-4223.