Universidade Estadual de Londrina

RODRIGO AUGUSTO IGAWA

MINERAÇÃO DE TEXTOS E WAVELETS APLICADAS NA CLASSIFICAÇÃO DE CONTAS EM REDES SOCIAIS DIGITAIS

Londrina
2016

RODRIGO AUGUSTO IGAWA

MINERAÇÃO DE TEXTOS E WAVELETS APLICADAS NA CLASSIFICAÇÃO DE CONTAS EM REDES SOCIAIS DIGITAIS

Dissertação apresentada ao Programa de Mestrado em Ciências da Computação da Universidade Estadual de Londrina para obtenção do título de Mestre em Ciência da Computação.

Orientador: Prof. Dr. Sylvio Barbon Jr

Londrina
2016

Ficha de identificação da obra elaborada pelo autor, através do Programa de Geração Automática do Sistema de Bibliotecas da UEL

Igawa, Rodrigo Augusto.
Mineração de Texto e Wavelets Aplicadas na Classificação de Contas em Redes Sociais Digitais / Rodrigo Augusto Igawa. - Londrina, 2016.
85 f. : il.

Orientador: Sylvio Barbon Jr.
Dissertação (Mestrado em Ciência da Computação) - Universidade Estadual de Londrina, Centro de Ciências Exatas, Programa de Pós-Graduação em Ciência da Computação, 2016.
Inclui bibliografia.

1. Redes Sociais Digitais - Teses. 2. Processamento Digital de Sinais - Teses. 3. Bots - Teses. I. Barbon Jr, Sylvio. II. Universidade Estadual de Londrina. Centro de Ciências Exatas. Programa de Pós-Graduação em Ciência da Computação. III. Título.

RODRIGO AUGUSTO IGAWA

MINERAÇÃO DE TEXTOS E WAVELETS APLICADAS NA CLASSIFICAÇÃO DE CONTAS EM REDES SOCIAIS DIGITAIS

Dissertação apresentada ao Programa de Mestrado em Ciências da Computação da Universidade Estadual de Londrina para obtenção do título de Mestre em Ciência da Computação.

BANCA EXAMINADORA

Orientador: Prof. Dr. Sylvio Barbon Jr
Universidade Estadual de Londrina – UEL

Prof. Dr. Bruno Bogaz Zarpelão
Universidade Estadual de Londrina – UEL

Prof. Dr. Alan Salvany Felinto
Universidade Estadual de Londrina – UEL

Prof. Dr. Emerson Cabrera Paraiso
Pontifícia Universidade Católica do Paraná – PUCPR

Londrina, 07 de abril de 2016.

AGRADECIMENTOS

Em primeiro lugar, agradeço a Deus pela própria vida e todo o resto que me proporciona.

Agradeço especialmente aos membros da minha família que, durante todo o tempo, não mediram esforçoes para me incentivar a cumprir esta etapa.

Agradeço minha namorada pela atenção e carinho.

Ao meu orientador, pela orientação, incentivo, atenção e dedicação.

Aos meus colegas de jornada que são inúmeros: colegas de laboratório, de disciplinas e de corredor. Agradeço a cada um por cada momento de descontração e alegria.

Ao Departamento de Computação (DC) da Universidade Estadual de Londrina (UEL), por sediar e prover a infraestrutura necessária para o desenvolvimento deste trabalho.

Agradeço à CAPES e a Fundação Araucária do Paraná que ajudaram a financiar meus estudos durante o tempo do programa.

IGAWA, R. A.. **Mineração de Texto e Wavelets Aplicadas na Classificação de Contas em Redes Sociais Digitais**. 85 p. Dissertação de Mestrado (Mestrado em Ciência da Computação) – Universidade Estadual de Londrina, Londrina–PR, 2016.

RESUMO

Para auxiliar a descoberta de fraudes em Redes Sociais Digitais (RSD), este trabalho propõe um modelo para classificação de contas de usuários baseada na Transformada Discreta Wavelet para caracterização do conteúdo textual. O principal objetivo da classificação é distinguir os padrões das classes em: Humanos, *Cyborgs* ou *Bots*. A abordagem proposta analisa a distribuição dos termos chaves da base de dados utilizada e visa manter o custo computacional adequado para RSD. Para a caracterização dos documentos em um processo de discretização, este trabalho propõe um novo esquema de peso chamado *Lexicon Based Coefficient Attenuation* (LBCA). A etapa de classificação é efetuada por meio do uso dos classificadores *Random Forests* e Perceptron Multi-Camadas. Os experimentos foram realizados com um conjunto de postagem obtidas durante os jogos da Copa do Mundo da FIFA de 2014. Os resultados mostraram que o modelo proposto obteve acurácias variando entre 94% e 100% na classificação entre as classes, permitindo a validação da discretização por meio das wavelets e com a contribuição do novo esquema de pesagem, adequado ao cenário das RSD.

Palavras-chave: Transformada Discreta Wavelet. Perceptron Multi-camadas. Random Forests. LBCA

IGAWA, R. A.. **Text Mining and Wavelets on Accounts Classification of Digital Social Networks**. 85 p. Master's Thesis (Master in Science in Computer Science) – State University of Londrina, Londrina–PR, 2016.

ABSTRACT

This work proposes a model to accounts classification according to its textual content and is grounded on Discrete Wavelets Transforms. Classification Task is mainly focused to match Online Social Networks accounts as Humans, Cyborgs or Bots. The proposed approach takes on consideration key terms frequency distribution and aims to maintain computational costs suitable. In order to aid the binning process, this work also proposes a new weighting scheme called Lexicon Based Coefficient Attenuation (LBCA). Classification step is carried out by using Multilayer Perceptrons and Random Forests. Experiments were performed on dataset crawled during 2014 FIFA World Cup Brazil. Results showed that proposed model achieved accuracies ranging from 94% to 100% which allows the validation of Discrete Wavelets Transforms along the proposed weighting scheme, suitable to Online Social Networks.

Keywords: Discrete Wavelet Transform, Multi-layer Perceptron. Random Forests. LBCA

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

LBCA	*Length-Based Coefficient Attenuation*
KDD	*Knowledge Discovery from Databases*
MD	Mineração de Dados
MT	Mineração de Texto
PDI	Processamento Digital de Imagem
PDS	Processamento Digital de Sinais
PMC	Perceptron Multi-Camadas
RI	Recuperação de Informação
RF	*Random Forests*
RNA	Redes Neural Artificial
RSO	Redes Sociais Online
RSD	Redes Sociais Digitais
TDC	Transformada Discreta de Cosseno
TDF	Transformada Discreta de Fourier
TDW	Transformada Discreta Wavelet

SUMÁRIO

1 INTRODUÇÃO

As Redes Sociais Digitais (RSD), também conhecidas como Redes Sociais *Online* (RSO), são consideradas ambientes digitais nos quais as pessoas discutem ideias e expressam opiniões sobre qualquer assunto [1, 2, 3]. Atualmente, as RSD representam uma fonte relevante de informações a serem exploradas em áreas como avaliação de opiniões e pesquisas de marketing [4, 5, 6].

Ao considerar que um montante de declarações textuais informais e subjetivas são criadas diariamente, o conhecimento minerado a partir desses textos poderia ser empregado por pesquisas científicas com fins financeiros e/ou políticos. Por exemplo, organizações que desejam melhorar seus produtos e serviços poderiam se beneficiar dos comentários feitos sobre os mesmos em blogs, sites ou redes sociais [4]. Por outro lado, clientes poderiam se beneficiar da mesma forma. De acordo com o que for desejado, seria possível estudar antes da compra se o produto ou serviço é bem avaliado em revisões de outros clientes e fazer um balanço sobre as opiniões disponíveis até então [7].

Para fins políticos, Sistemas Baseados em Conhecimento podem se aproveitar da instantaneidade do Twitter[1] que já foi reportado como uma forma de propagação de informações mais rápida que muitas mídias de notícias. Usando tal fonte, seria possível avaliar a opinião pública com relação a algum evento antes mesmo que qualquer outro concorrente o fizesse [8].

Entretanto, se por um lado as pessoas conseguem se beneficiar do conteúdo disponível nas RSD, é possível também causar danos. A grande popularidade e facilidade de acesso às RSD também resultou na presença de usuários não desejados. Juntamente com problemas relacionados à privacidade de seus próprios usuários, as RSD apresentam problemas para identificar casos que envolvam usuários desempenhando atividades maliciosas, por exemplo, o *spam*, que define o envio repetido e excessivo de conteúdo não solicitado, sendo a atividade mais comum [9].

Para ilustrar o cenário do problema, em 2012, foi divulgado no trabalho de Fong *et al.* [10] que o Facebook[2] apresentava 8,7% de contas falsas (83 milhões) no mundo todo. A mesma situação ocorre no Twitter. Uma vez que a relação entre usuários se dá principalmente pelo compartilhamento de interesses, consequentemente os usuários do Twitter também se tornaram alvos de campanhas de marketing, notícias tendenciosas para manipulação social e campanhas políticas [11].

Contas falsas, são em grande parte, representadas por contas com comportamento

[1] https://twitter.com/
[2] http://www.facebook.com

automático denominadas robôs e popularmente conhecidas como *bots* (do inglês *robots*). O objetivo dessas contas varia entre a postagem de *spam* de produtos, links maliciosos para prática de *phishing*, uma tentativa de fraude em que se tenta obter dados sensíveis como senhas da vítima, ou simplesmente fazem volume para aparentar que uma dada entidade é mais popular do que realmente é. No Twitter, *bots* tem como objetivo se passar por humanos para assim ganharem seguidores e serem capazes de disseminar suas atividades em grande escala [12].

Fraudes em RSD, como mencionadas, poderiam resultar na disseminação descontrolada de informações falsas, postagens promocionais disfarçadas e *phishing*. Dessa forma, o objetivo das RSD de disseminar conteúdo legítimo, notícias e experiências agradáveis estaria comprometido. Os usuários seriam frequentemente incomodados pela presença de conteúdo malicioso causando declínio do serviço [9, 13].

Como mecanismo de defesa, as RSD disponibilizam uma forma de ação contra qualquer usuário que cause alguma forma de dano como uso incorreto de logomarcas, *spam* ou mesmo publicação de conteúdos ofensivos. No Facebook, o processo para denunciar uma conta que apresente algum comportamento indevido tem origem nos próprios usuários. É necessário que a conta suspeita seja denunciada para, somente então, uma investigação ser iniciada. Normalmente, o processo é longo e propicia tempo para que o usuário da conta acusada se defenda [10]. O problema é que, se tratando de uma grande quantidade de *bots*, no momento em que alguma medida por parte da RSD seja tomada, danos irreparáveis já podem ter sido realizados [14].

Portanto, considerando que as defesas nativas das RSD não são suficientes, Fong *et al.* [10] propuseram fazer uso de árvores de decisão para identificar contas com comportamento malicioso. Experimentos foram bem sucedidos em termos de acurácia para o Facebook.

No Twitter, a preocupação com políticas de proteção contra atividades maliciosas também têm sido objeto de estudo. Considerando *spams*, uma diversidade de estratégias foram exploradas. De 2009 à 2014, foram realizados estudos, em que os primeiros utilizam listas de links maliciosos como base [15], e o mais recente, baseia-se em uma sistema de detecção de anomalias (Mineração de Fluxo) [13].

Ainda em relação ao Twitter, Chu *et al.* [11, 16], propuseram a classificação de contas do Twitter em três classes distintas: humanos, *cyborgs* (um intermediário entre uma conta que posta conteúdos automaticamente e um humano) e *bots*. Para tanto, o autor fez uso de diversas informações, por exemplo, plataforma usada para postagem (mobile ou Web), análise de *links* postados pela conta, quantidade de relacionamentos, entre outras. Juntamente com a proposta de classificação, os autores também descreveram as diferenças entre as três contas.

Com o objetivo de contribuir com a segurança nas RSD e vialibilizar a classificação de contas de RSD sem o uso de muitas informações distintas, o foco principal deste trabalho foi propor um modelo baseado em Mineração de Texto (MT) que proporcione a classificação de contas de usuários das RSD para evitar a disseminação descontrolada de informação fraudulenta e permitir que usuários ainda possam utilizar as RSD para tomada de decisões. O modelo proposto foi avaliado em duas situações: a) classificar as contas como *bots* ou humanos e b) assim como Chu *et al.* [11, 16], classificar as contas como *bots*, humanos ou *cyborgs*.

É importante destacar que justamente por se tratar de um trabalho de MT, a maior contribuição é viabilizar a classificação fazendo somente o uso de texto, ou seja, nenhum outro tipo de informação como hora, frequência ou características do perfil do usuário são necessárias. Em relação ao modelo proposto, esta abordagem faz uso de Transformada Discreta Wavelet (TDW) e MT, cuja combinação já foi proposta anteriormente na literatura para diversos fins [17, 18]. A principal vantagem de fazer uso de TDW é a possibilidade de observar a distribuição de frequência dos principais termos de uma base para usá-los como descritores. Para a etapa de classificação, Perceptron Multi-Camadas (PMC) e *Random Forests* (RF) foram testados devido ao seus frequentes usos na literatura.

Por fim, com experimentos, buscou-se efetuar testes que analisem parâmetros de configuração da própria abordagem como: a) como as medidas de Discretização afetam o modelo proposto?; b) qual dos esquemas de peso mais contribui para a acurácia do modelo?; c) qual a família de TDW melhor se adequa ao problema? e d) quais resultados PMC e RF apresentam nos experimentos?. O restante deste trabalho está organizado da seguinte forma:

- O Capítulo 2 apresenta os conceitos e métodos nos quais este trabalho foi baseado. Inicialmente são apresentados alguns conceitos em relação ao Twitter, a RSD usada neste trabalho para experimentos. Posteriormente, são discutidos os conceitos que foram usados no modelo proposto. Entre tais estão: Mineração de Dados (MD), MT, TDW, PMC e RF.

- O Capítulo 3 apresenta os trabalhos relacionados, exemplificando como soluções relacionadas a este trabalho foram propostas. Primeiramente, são apresentadas as abordagens disponíveis na literatura para solucionar diversas fraudes acerca de RSD e, em seguida, trabalhos relacionados a TDW e MT.

- O Capítulo 4 descreve o modelo proposto apresentado neste trabalho. O modelo é dividido em 5 etapas: Concatenação, Discretização, Pesagem, TDW, seleção de atributos e Classificadores. O objetivo de cada etapa é conseguir transformar os

textos produzidos por usuários de RSD em descritores de sinais para, então, poder classificá-los.

- O Capítulo 5 aborda experimentos e resultados. Primeiramente, considerações em relação as configurações dos experimentos são apresentadas juntamente com detalhes da base usada. Posteriormente, são feitas discussões sobre os resultados de cada camada do modelo proposto para, enfim, discutir configurações praticáveis.
- O Capítulo 6 apresenta a conclusão. São feitas discussões finais sobre as limitações do trabalho realizado e diretrizes para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Este Capítulo apresenta os conceitos e métodos nos quais esta dissertação foi baseada. A Seção 2.1 apresenta os conceitos relacionados às RSD com o foco no Twitter, uma vez que os experimentos realizados neste trabalho utilizaram uma base de dados provenientes dessa RSD. Em seguida, os conceitos sobre MD e MT são apresentados na Seção 2.2. Apresentar os conceitos em relação a MD e MT é importante devido ao modelo proposto ser baseado em estratégias de mineração. Posteriormente, para apresentar as vantagens obtidas ao utilizar os sinais baseados em termos apresentados na Seção 4.2, conceitos fundamentais de TDW são descritos na Seção 2.3. Por fim, para a etapa de classificação, são apresentados os conceitos relacionados ao Aprendizado de Máquina na Seção 2.4.

2.1 Redes Sociais Digitais e Twitter

A noção que se tem por RSD é mostrada em [1]. Basicamente, entende-se por uma RSD qualquer agregação social que acontece por meio virtual em que os indivíduos trocam ideias e expressam opiniões sobre qualquer assunto e representam recursos de considerável relevância para diversas atividades [5].

Um exemplo de RSD é o Twitter. Em [1], boa parte das funcionalidades do Twitter são descritas. Por exemplo, uma postagem (*tweet*) não deve exceder 140 caracteres, *links* e imagens são permitidos. Tais características fazem do mesmo um exemplo de microblog.

Em contraste com grande maioria das outras RSD disponíveis, no Twitter, não é esperada uma conversa em estilo de *chats*. A interação entre usuários é feita através do ato de seguir e permitir que alguém te siga.

Ao seguir alguém, o usuário seguidor será notificado sempre que o seguido escrever ou sempre que o seguido for citado. Da mesma forma, ao permitir que alguma pessoa o siga, o usuário permitirá que o mesmo seja notificado em relação as suas postagens.

A forma de diálogo é feita através do uso do caractere @ representando uma citação de usuário. Por exemplo:

"@usuario1, qual foi o resultado do jogo de ontem?"

Todos os usuários que seguem o "usuario1" podem visualizar a postagem (*tweet*) acima. O "usuário1" pode, da mesma forma, citar quem o citou e permitir que todos os seguidores visualizem a postagem. É importante notar como o caráter de discussão pública é incentivada no Twitter.

Outro exemplo de funcionalidade também descrita em [1], é a possibilidade de

republicar uma postagem.

"RT @usuario1 que frango."

Nesse caso, a sigla "RT" é usada para indicar uma republicação e a citação seguinte a sigla é usada para informar quem escreveu a postagem original. Essa atividade é conhecida popularmente como *retweetar* e a postagem como *retweet*.

Um terceiro exemplo de funcionalidade é o uso das *hashtags*. Denotadas pelo carácter #, elas possuem um objetivo diferente da citação e do *retweet*. O objetivo do uso das *hashtags* é fornecer uma forma de organização para postagens. Portanto, são usualmente empregadas para marcar assuntos, eventos e afins. Atualmente, é possível visualizar os assuntos discutidos ao redor do mundo nos últimos instantes[1] baseados nas *hashtags*.

2.2 Mineração de Dados e Mineração de Texto

Como apresentado por Zhang *et al.* [19], é chamado de MD a ação de extrair informação a partir de grandes volumes de dados. *Knowledge Discovery from Databases* (KDD), por sua vez é o nome dado a todo o processo desde aquisição dos dados até avaliação do modelo. Na prática, a partir do banco de dados de uma organização, empresários podem descobrir padrões e comportamentos não explícitos sobre seus produtos ou clientes. Por exemplo, seria possível descobrir de forma automática quais os tipos de clientes mais adequados para comprar cada produto da organização. Dessa forma, seria possível facilitar e auxiliar empresários em tomadas de decisões.

A Figura 2.1 ilustra um precesso de KDD. Em geral, a execução sequencial das etapas consiste em [19, 20]:

- **Aquisição:** Esta etapa consiste na obtenção de todos os dados necessários. O objetivo principal desta etapa é a junção de todas as bases necessárias para a formação de uma única fonte de dados.

- **Preprocessamento:** Neste passo do processo, a confiabilidade dos dados coletados é enfatizada. Uma vez que dados colhidos de bancos de dados de sistemas em cenários reais apresentam grandes quantidades de ruído, torna-se importante que os dados adquiridos passem por uma etapa de limpeza para correção de ocorrências de dados não presentes ou fora de um padrão (*outliers*). Por exemplo, para atributos numéricos, medidas estatísticas como média, desvio padrão e intervalo de confiança podem ser usadas para a detecção de *outliers*. Uma vez encontrados, é recomendado a remoção *outliers* devido a não contribuirem nas etapas posteriores.

[1] http://trendsmap.com/

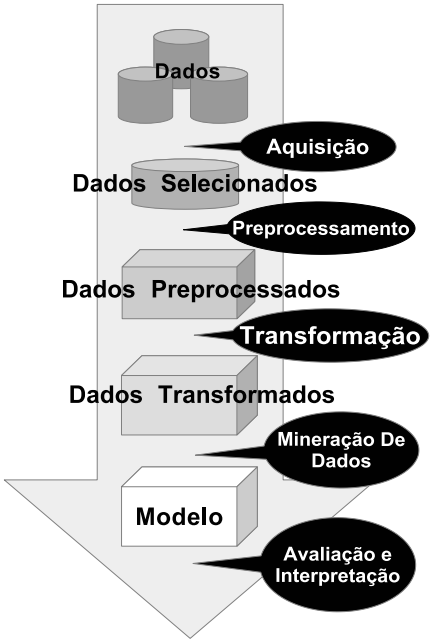


Figura 2.1 – Processo de KDD retirado e adaptado de Maimon e Rokach [20].

- **Transformação:** Seleção de atributos e extração de atributos são as principais tarefas efetuadas nesta etapa. Transformação de atributos como discretização de atributos também pode ser efetuada quando necessário. A etapa de transformação de dados é considerada como fundamental para um processo de KDD, pois em diversos cenários pode ser importante extrair valores a partir atributos existentes ao invés de considerá-los individualmente. Usar apenas parte dos atributos existentes pode ser relevante com relação a performance em casos que milhares de atributos estão envolvidos.

- **MD:** Esta etapa é conhecida por ser o momento da escolha de um algoritmo de MD. Embora não exista uma regra definida para a escolha do melhor algoritmo para cada situação, existem algumas diretrizes para auxiliar esta etapa. Se for considerado que a compreensão de como os atributos influenciam a decisão do algoritmo, árvores de decisão são recomendadas devido a sua fácil legibilidade. Se considerado que a acurácia do modelo é mais importante, PMC é recomendada. Ainda, quando possível, é recomendado que um experimento envolvendo mais de um algoritmo sempre seja efetuado para a escolha do melhor modelo.

- **Avaliação e interpretação:** Por fim, nesse estágio, são avaliados alguns aspectos relacionados ao algoritmo. Avalia-se sua confiabilidade, legibilidade e precisão. Quanto a confiabilidade, é considerado se o mesmo sucesso obtido pelo algoritmo no conjunto de treinamento será também obtido quando colocado em um cenário real. Quanto a legibilidade, se o algoritmo é capaz de auxiliar seus usuários na compreensão do problema. Quanto a precisão, se o algoritmo é eficaz o suficiente para se

manter em funcionamento. Caso algum dos critérios não seja satisfeito, é possível voltar para qualquer uma das etapas anteriores para a melhora do modelo.

A MT, inspirada em KDD e MD [21], segue um modelo como mostrado na Figura 2.2. Ao observar a Figura 2.1, é possível perceber algumas semelhanças entre MT e KDD. Ao considerar a parte superior da Figura 2.2, é possível notar os seguintes passos para um processo de MT [21]:

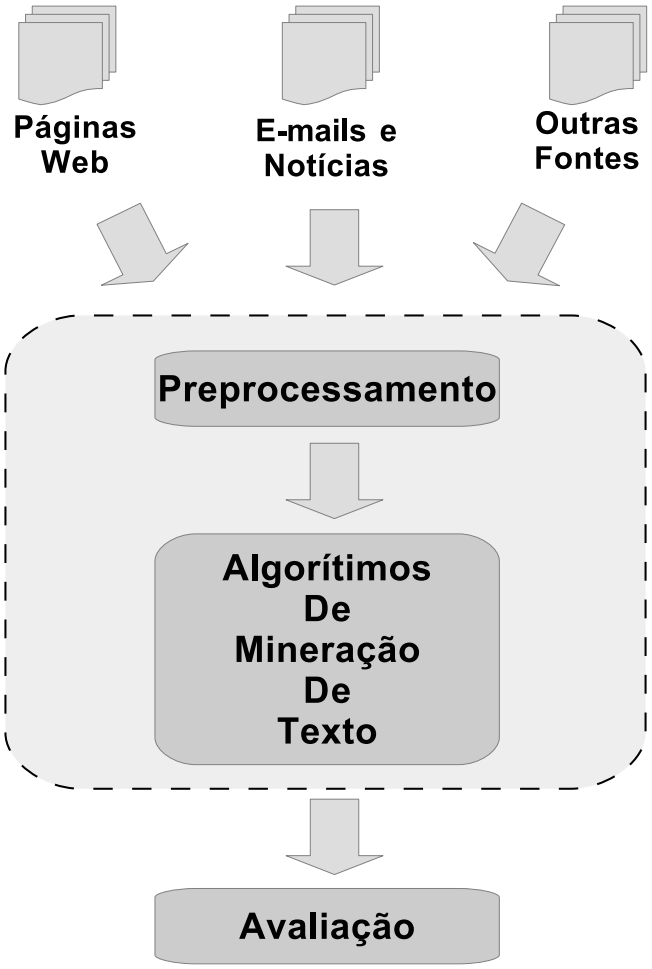


Figura 2.2 – Processo de MT adaptado de [21].

- **Aquisição:** Diferentemente de um processo de KDD, a aquisição dos dados não é sempre feita baseada em um banco de dados. Na verdade, como é visto na parte superior da Figura 2.2, é muito comum que o conjunto de dados a ser minerado seja uma coleção de documentos de uma ou mais fontes de texto. Por exemplo, pode ser a junção de e-mails, noticiários em páginas Web e blogs. Ainda, como é caso da abordagem proposta nesse trabalho, é possível que o conjunto de dados seja obtido em outra fonte: *microblogs*. O objetivo final da etapa de aquisição é a obtenção de

uma **coleção de documentos**, onde cada **documento** representa uma unidade textual do cenário real como um e-mail, um relatório, uma redação ou um artigo.

- **Preprocessamento:** Analogamente ao KDD, o conjunto de dados adquiridos, neste caso coleção de documentos, não é diretamente usado por um algoritmo de MD sem ser preprocessado antes. O objetivo desta etapa é preparar os dados adquiridos para que sejam apropriadamente usados por um algoritmo de mineração de texto. Existe um número considerável de tarefas inclusas na etapa de preprocessamento. Cada caso pode exigir a execução de uma ou mais tarefas. Alguns exemplos incluem:

  - *Case Convertion:* uma tarefa simples que consiste na normalização entre caracteres escritos em caixa alta e caixa baixa. Naturalmente, pode ser que em alguns casos, como no caso do reconhecimento de códigos fonte, não seja possível efetuar essa tarefa. Em casos normais, "Não quero" se tornaria "não quero". Desta forma, "Não quero" e "NÃO quero" seriam considerados os mesmos termos.
  - *Tokenization:* com esta tarefa, busca-se definir dentro de um **documento** o que será considerado como a menor unidade de texto chamada de **termo**. Por exemplo, em casos simples, uma palavra pode ser equivalente a um termo. Porém, em algumas tarefas, como no caso de análise de sentimento, é muito frequente agrupar algumas palavras em único termo. As duas palavras em "Não gosto" poderiam então ser representado como "Não_gosto" em único termo. *Tokenization* pode ser importante como em casos do idioma inglês onde é interessante considerar "do_not_like" um único termo para representar negação.
  - *Stopwords Removal:* consiste na remoção de todos os termos encontrados em uma lista. Normalmente considera-se na lista um conjunto de palavras que não carregam significado, como é o caso de preposições e artigos. Links no caso de páginas Web também podem ser acrescentados à lista de *stopwords*. Em geral, pode-se acrescentar termos necessários de acordo com cada caso. A principal contribuição desta tarefa seria a redução de termos no documento.
  - *Stemming:* essa tarefa tem como objetivo reduzir palavras em radicais. Por exemplo, no caso dos seguintes termos em inglês: "fisher" e "fishing". Após a execução dessa tarefa, ambos os termos serão apenas "fish". É uma técnica normalmente baseada na remoção de sufixos muito presente em casos de Recuperação da informação (RI). Assim como o caso de *stopwords removal* a principal contribuição é a redução dos dados.

  Tarefas de KDD como seleção e extração de características são consideradas em MT dentro da etapa de preprocessamento. Um exemplo de característica que pode ser extraída em MT seria o número de substantivos e verbos, número médio de palavras

antes e após a etapa de *stopwords removal* ou ainda a frequência de ocorrência de cada palavra-chave presente no texto.

- **algoritmos de Mineração de Texto:** assim como no caso de KDD, em MT também é possível a aplicação de algoritmos de aprendizado de máquina. Hipoteticamente, para classificar documentos em categorias como romance, comédia, suspense e mistério é possível a aplicação de árvores de decisão, PMC entre outros [21]. algoritmos de aprendizado de máquina não supervisionados também possibilitam o agrupamento de notícias relacionadas à um tema.

- **Avaliação:** Analogamente ao caso de KDD, é fundamental avaliar se o modelo de MT elaborado é adequado. É viável efetuar esta etapa da mesma forma que se efetuaria no caso de KDD. Por exemplo, no caso da elaboração de um buscador como seria um típico caso de RI, para avaliar o modelo, medidas como *precision* & *recall*, descritas detalhamente em [22] poderiam ser usadas para verificar o sucesso do buscador.

2.3 Transformada Discreta Wavelet

A TDW consiste em uma alternativa eficiente para análise de dados no domínio da frequência, semelhante à análise obtida através da Transformada Discreta de Fourier (TDF). Diversos problemas como compressão de sinal, reconhecimento de voz, detecção de objetos e eliminação de ruído em imagens podem ser resolvidos com o auxílio da TDW [23]. Como pode ser visto detalhadamente em [24], a TDW pode ser vista como um par de filtros, sendo um passa-baixas ($h[\ ]$) e um passa-altas ($g[\ ]$).

Dado um sinal f, onde f pode ser representado como $[f_1, f_2, ..., f_N]$, discreto e de comprimento N, f é submetido a ambos os filtros. A sua primeira metade (ou seja, $[f_1, f_2, ..., f_{\frac{N}{2}}]$) é submetido ao filtro passa-baixas e o restante ($[f_{\frac{N}{2}+1}, ..., f_N]$) ao filtro passa-altas. Tendo aplicado ambos os filtros pela primeira vez, um novo sinal é obtido. A primeira metade do sinal representa o resultado da aplicação do filtro passa-baixas e é um vetor de comprimento $\frac{N}{2}$ chamado de Aproximação (A). A segunda metade representa a aplicação do filtro passa-altas também de comprimento $\frac{N}{2}$ chamado de Detalhe (D). Uma vez aplicado ambos os filtros, a primeira metade do sinal resultante é submetida a ambos os filtros novamente. Este processo é chamado de decomposição do sinal e é ilustrado na Figura 2.3 que mostra a decomposição do sinal em até 3 níveis.

É importante notar que, a cada nível, o comprimento do sinal passa por um processo de *downsampling* por 2, o que reduz pela metade o comprimento do sinal a ser filtrado. Por exemplo, caso o sinal f tenha comprimento $N = 8$, ao final do primeiro nível de decomposição obtêm-se um sinal com duas partes compostas por 4 amostras. As primeiras 4 amostras representam o resultado do filtro passa-baixas e as 4 amostras restantes

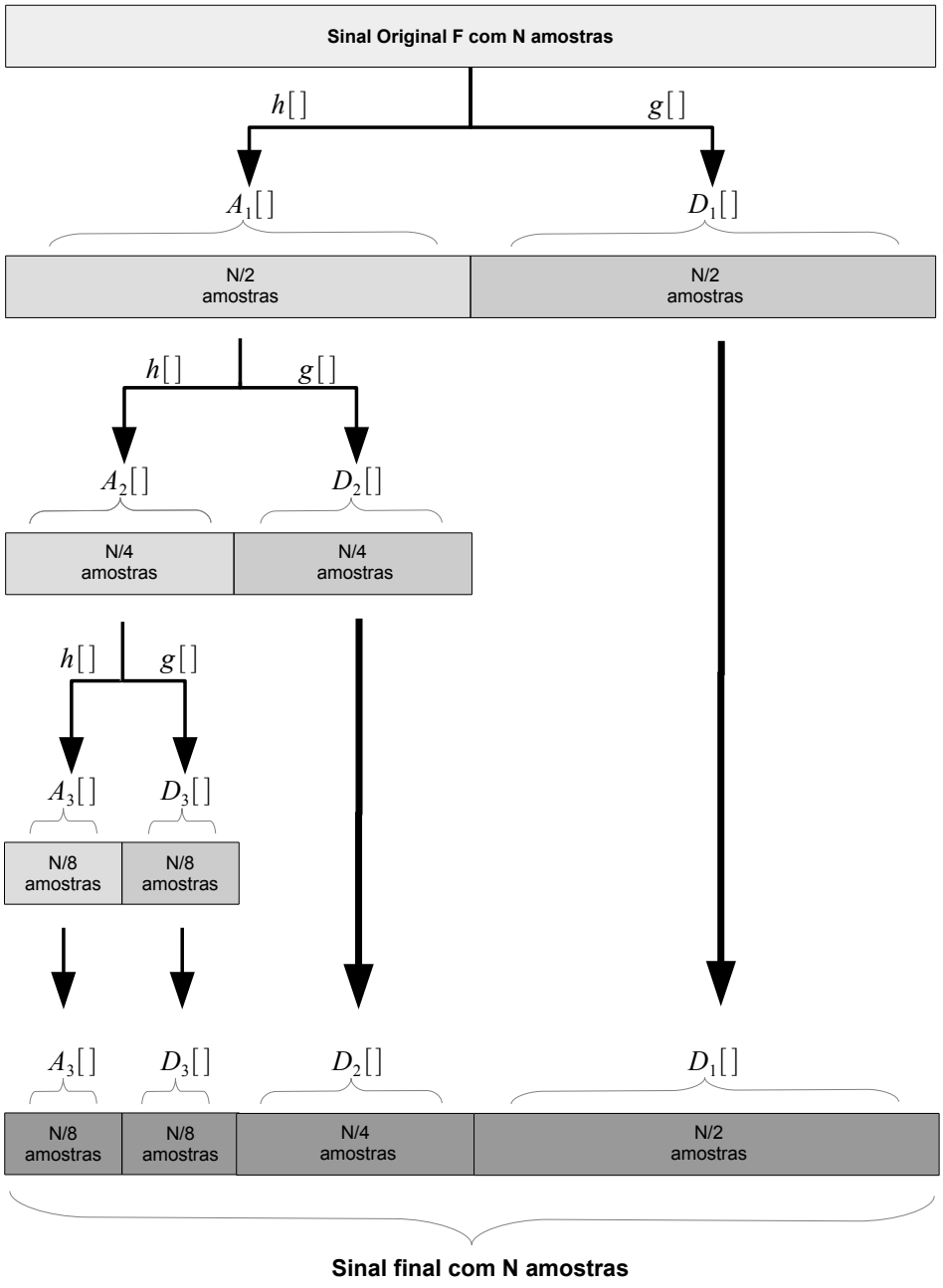


Figura 2.3 – Processo de decomposição da TDW.

representam o resultado do filtro passa-altas. Para o segundo nível, o mesmo processo é efetuado usando somente a primeira metade do sinal, ou seja as 4 primeiras amostras. Como pode ser visto na Figura 2.3, as amostras do sinal resultante são representados pelos quadros em cinza. Ainda, é possível perceber que o sinal resultante tem o mesmo número de amostras que o sinal original, sendo composta pelos coeficientes provenientes da aplicação do filtro passa-baixas do último nível seguido dos coeficientes provenientes da aplicação do filtro passa-altas nos demais níveis de composição. Por fim, o número máximo de níveis a serem aplicados é $log_2(N)$, onde N é o comprimento do sinal.

Outra característica interessante da TDW é sua divisão em famílias. Por exem-

plo, filtros com características semelhantes, usualmente do mesmo criador, são agrupadas formando famílias. Dentro de uma mesma família, filtros de diferentes suportes são encontrados. A TDW Haar é encontrada no MATLAB[2] como membro da família Daubechies e seu respectivo suporte é 2, ou seja, seus filtro passa-baixas e passa-altas são ambos de tamanho 2. Um outro exemplo de membro da família Daubechies é a de suporte 4, chamada Daubechies-4. A seguir, na próxima Seção, um exemplo de cálculo da TDW é apresentado.

2.3.1 Cálculo da Transformada Discreta Wavelet

Para o cáculo da TDW de um sinal, são necessários apenas os filtros $h[\ ]$ e $g[\ ]$ [24, 17]. O cálculo da TDW pode ser realizado pela multiplicação de duas matrizes $(W * f^t)$, onde uma matriz corresponde aos filtros (W) e a segunda matriz ao sinal a ser transformado (f).

$$W = \begin{bmatrix}
h_1 & h_2 & \cdots & h_n & 0 & 0 & \cdots & \cdots & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & h_1 & h_2 & \cdots & h_n & \cdots & \cdots & 0 & 0 & 0 & 0 & 0 & 0 \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
0 & 0 & 0 & 0 & 0 & 0 & \cdots & \cdots & h_1 & h_2 & \cdots & h_n & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \cdots & \cdots & 0 & 0 & h_1 & h_2 & \cdots & h_n \\
g_1 & g_2 & \cdots & g_n & 0 & 0 & \cdots & \cdots & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & g_1 & g_2 & \cdots & g_n & \cdots & \cdots & 0 & 0 & 0 & 0 & 0 & 0 \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
0 & 0 & 0 & 0 & 0 & 0 & \cdots & \cdots & g_1 & g_2 & \cdots & g_n & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \cdots & \cdots & 0 & 0 & g_1 & g_2 & \cdots & g_n
\end{bmatrix}$$

$$f = \begin{bmatrix} f_1 & f_2 & f_3 & \cdots & \cdots & \cdots & f_{N-1} & f_N \end{bmatrix}$$

Um detalhe importante para o cálculo da TDW é o posicionamento dos filtros $h[\ ]$ e $g[\ ]$ na matriz W. Se o sinal a ser transformado possui tamanho N, então as $\frac{N}{2}$ primeiras linhas da matriz W são preenchidas com o filtro passa-baixas $h[\ ]$. As demais linhas da matriz W são preenchidas com o filtro passa-altas $g[\ ]$. Porém, como mostrado na matriz W, os filtros não ocupam todas as colunas de uma determinada linha da matriz W. Na prática os n coeficientes dos filtros $h[\ ]$ e $g[\ ]$ começam a ser posicionados na primeira coluna da matriz e para as demais linhas avançam 2 colunas sucessivamente. Para possibilitar a multiplicação de matrizes $(W * f^t)$, a matriz W possui N linhas e N colunas, de acordo com o tamanho N de f.

[2] http://www.mathworks.com/products/matlab/

Para ilustrar os passos descritos acima, supondo que f_1 seja o seguinte sinal a ser transformado:

$$f_1 = \begin{bmatrix} 5 & 3 & 5 & 2 & 0 & 1 & 0 & 2 \end{bmatrix}$$

Usando a família Wavelet de Haar, em algumas ferramentas como MATLAB pertencente à família de Daubechies, cujo filtros (de suporte 2) passa-baixas e passa-altas são respectivamente: $h = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}$ e $g = \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}$. Neste caso, a matriz de filtros, agora W_1 de tamanho 8×8 de acordo com o comprimento do sinal f_1, é:

$$W_1 = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}$$

Como descrito anteriormente, as $\frac{N}{2}$ primeiras linhas, no caso 4 primeiras, são preenchidas com o filtro passa-baixas, e as demais linhas com o filtro passa-altas. Assim como no exemplo genérico, os filtros são posicionados 2 colunas a frente em relação à linha anterior.

Portanto, o resultado da decomposição em primeiro nível é:

$$W_1 * f_1^{\mathsf{T}} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} * \begin{bmatrix} 5 \\ 3 \\ 5 \\ 2 \\ 0 \\ 1 \\ 0 \\ 2 \end{bmatrix} = \begin{bmatrix} \frac{8}{\sqrt{2}} \\ \frac{7}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ \frac{2}{\sqrt{2}} \\ \frac{2}{\sqrt{2}} \\ \frac{3}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \\ -\frac{2}{\sqrt{2}} \end{bmatrix} \tag{2.1}$$

Assim, $\begin{bmatrix} \frac{8}{\sqrt{2}} & \frac{7}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{2}{\sqrt{2}} \end{bmatrix}$ (A_1) e $\begin{bmatrix} \frac{2}{\sqrt{2}} & \frac{3}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & -\frac{2}{\sqrt{2}} \end{bmatrix}$ (D_1) são os sinais provenientes do filtro passa-baixas e passa-altas, respectivamente. Para iniciar o segundo nível da decomposição, usa-se apenas a primeira metade do sinal resultante, ou seja, (A_1). A matriz de filtros (W_2) e a função a ser transformada f_2, no caso $f_2 = A_1$, são:

$$W_2 = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} \text{ e } f_2 = \begin{bmatrix} \frac{8}{\sqrt{2}} \\ \frac{7}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ \frac{2}{\sqrt{2}} \end{bmatrix}$$

Assim como na etapa anterior para W_1, a primeira metade das linhas de W_2 são preenchidas usando os filtros passa-baixas enquanto a segunda metade das linhas são preenchidas usando o filtro passa-altas. A posição dos filtros sempre avançando 2 colunas a cada linha também é mantida. O resultado da multiplicação no segundo nível fica:

$$W_2 * f2 = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} * \begin{bmatrix} \frac{8}{\sqrt{2}} \\ \frac{7}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ \frac{2}{\sqrt{2}} \end{bmatrix} = \begin{bmatrix} \frac{15}{\sqrt{4}} \\ \frac{3}{\sqrt{4}} \\ \frac{1}{\sqrt{4}} \\ -\frac{1}{\sqrt{4}} \end{bmatrix} \tag{2.2}$$

Analogamente ao passo anterior, $\begin{bmatrix} \frac{15}{\sqrt{4}} & \frac{3}{\sqrt{4}} \end{bmatrix}$ é a parte do sinal proveniente do filtro passa-baixas (A_2) e $\begin{bmatrix} \frac{1}{\sqrt{4}} & -\frac{1}{\sqrt{4}} \end{bmatrix}$ proveniente do passa-altas (D_2). Para o terceiro nível, mais uma vez, apenas a parte proveniente do filtro passa-baixas da etapa anterior (A_2) é usada, de forma de que a matriz de filtros W_3 e a função a ser filtrada $f_3 = A_2$ são:

$$W_3 = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} \text{ e } f_3 = \begin{bmatrix} \frac{15}{\sqrt{4}} \\ \frac{3}{\sqrt{4}} \end{bmatrix}$$

Mantendo a ordem do posicionamento dos filtros passa-baixas e passa-altas. A única diferença é, que desta vez, não há mais espaço para avançar 2 colunas a cada nível, afinal o tamanho do sinal a ser transformado é apenas 2. Por fim, o resultado da multiplicação em terceiro nível é:

$$W_3 * f_3 = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} * \begin{bmatrix} \frac{15}{\sqrt{4}} \\ \frac{3}{\sqrt{4}} \end{bmatrix} = \begin{bmatrix} \frac{18}{\sqrt{8}} \\ \frac{12}{\sqrt{8}} \end{bmatrix} \tag{2.3}$$

Onde $\frac{18}{\sqrt{8}}$ é o resultado proveniente do filtro passa-baixas (A_3) e $\frac{12}{\sqrt{8}}$ é o resultado proveniente do filtro passa-altas D_3. Uma vez que o sinal original f_1 possui tamanho $N = 8$ não é possível decompor o sinal mais vezes. Como descrito na Seção anterior, é possível decompor um número máximo igual a $log_2(N)$. No caso, foi possível decompor o sinal em até três niveis.

Ainda como descrito na Seção anterior, o sinal resultante da TDW é composto pelo resultado do filtro passa-baixas do último nível, concatenado com os resultados dos

filtros passa-altas do nível atual juntamente com os níveis anteriores. Em outras palavras o sinal resultante é descrito como:

$$f_{final} = \begin{bmatrix} A_3 \\ D_3 \\ D_2 \\ D_1 \end{bmatrix} = \begin{bmatrix} \frac{18}{\sqrt{8}} \\ \frac{12}{\sqrt{8}} \\ \frac{1}{\sqrt{4}} \\ -\frac{1}{\sqrt{4}} \\ \frac{2}{\sqrt{2}} \\ \frac{3}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \\ -\frac{2}{\sqrt{2}} \end{bmatrix} \tag{2.4}$$

A Figura 2.4 ilustra um resumo do processo descrito.

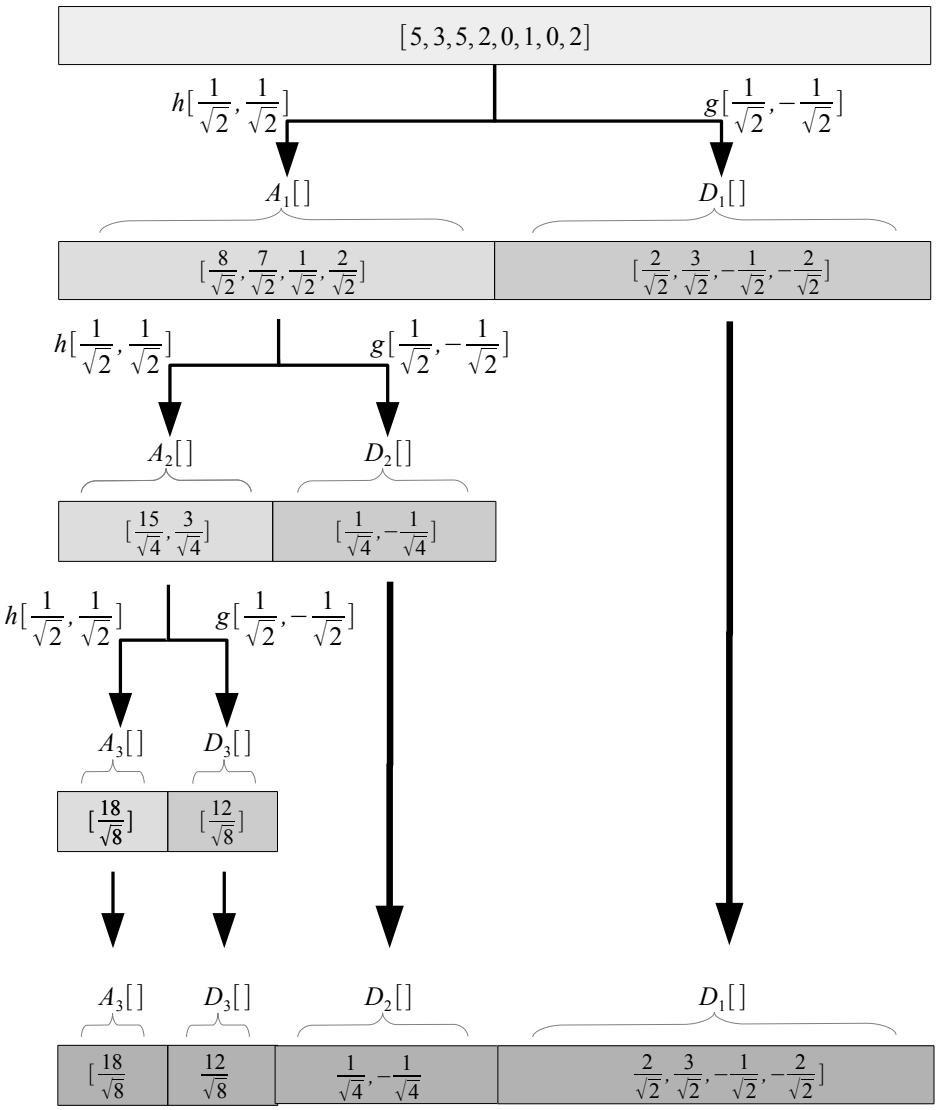


Figura 2.4 – Exemplo de cálculo da TDW.

Por fim, a maior contribuição da TDW para este trabalho foi demonstrada por Park [17] e é sobre o significado de cada componente de f_{final}. Por exemplo, o primeiro componente de f_{final} ($\frac{18}{\sqrt{8}}$) representa, em seu numerador (18), a soma de todos os valores de f_1 ($[5, 3, 5, 2, 0, 1, 0, 2]$). O segundo componente de f_{final} ($\frac{12}{\sqrt{8}}$), também em seu numerador (12), representa a diferença entre a soma da primeira metade ($[5, 3, 5, 2]$) e a segunda metade de f_1 ($[0, 1, 0, 2]$). O terceiro componente de f_{final} ($\frac{1}{\sqrt{4}}$), em seu numerador (1), representa a diferença do primeiro quarto ($[5, 3]$) e o segundo quarto ($[5, 2]$) de f_1. O quarto componente de f_{final} ($-\frac{1}{\sqrt{4}}$), em seu numerador (-1), representa a diferença do terceiro quarto ($[0, 1]$) e o quarto quarto ($[0, 2]$) do sinal de f_1. Por fim, os 4 últimos componentes de f_1, representam a diferença entre os pares de oitavos do sinal de f_1. Por exemplo, o quinto componente de f_{final} ($\frac{2}{\sqrt{2}}$), em seu numerador (2), representa a diferença entre o primeiro oitavo do sinal (5) e o segundo oitavo (3) do sinal de f_1. Analogamente, o sexto componente de f_{final} ($\frac{3}{\sqrt{2}}$), em seu numerador 3, representa a diferença do terceiro oitavo 5 e o quarto oitavo 2 do sinal de f_1.

Portanto, ao invés de usar apenas a soma dos valores de f_1 ou mesmo usar f_1 completamente, este trabalho faz uso de TDW para aproveitar a vantagem de analisar a distribuição de frequência dos termos para descrever o comportamento do uso dos termos chaves.

2.4 Aprendizado de Máquina

Aprendizado de máquina é um campo de pesquisa que tem como objetivo ensinar o computador a reconhecer padrões. Por exemplo, segundo Dougherty [25], para um ser humano normal é uma tarefa fácil distinguir um número ao visualizá-lo. Da mesma forma, é fácil distinguir faces. Para o autor, aprendizado de máquina é o campo que busca algoritmos que permitam o computador a reconhecer tais padrões. O rosto de cada pessoa possui alguns padrões como posição do olho, boca e nariz. Com o auxilio do aprendizado de máquina, um computador pode então ser capaz de indentificar esses padrões específicos de determinado rosto (classe). Podem haver vários rostos (classes) diferentes e ao encontrar uma nova imagem de uma face, busca-se identificá-lo a quem pertence (classe).

Aplicações citadas por Dougherty [25] incluem: biometria (reconhecimento de impressões digitais), reconhecimento de caracteres, classificação de documentos entre maliciosos ou legítimos, reconhecimento de voz, entre outos. Para este trabalho, aprendizado de máquina é usado para que contas de usuários sejam automaticamente classificadas de acordo com sua produção textual.

Como será apresentado posteriormente, este trabalho faz uso de um conjunto de dados previamente classificados. Portanto, os algoritmos usados são conhecidos por um sub campo conhecido como aprendizado de máquina supervisionado.

2.4.1 Abordagem Supervisionada

Algumas vezes, pode ser que o conjunto de dados não contenha classificações previamente definidas. Nestas situações, algoritmos de aprendizado de máquina não supervisionado devem ser capazes de identificar agrupamentos entre exemplos semelhantes [26].

Nos experimentos realizados neste trabalho, o conjunto de dados foi previamente classificado. Portanto, o conjunto de dados é processado pelo algoritmo que possui marcações com a resposta correta para cada exemplo. A partir destes exemplos, um modelo é gerado pelo algoritmo que passa a estar apto para classificar exemplos futuros [26].

Foram efetuados experimentos com 2 métodos distintos de aprendizado de máquina supervisionado. Portanto, a Seção 2.4.1.1 descreve os principais conceitos envolvendo o funcionamento da PMC e a Seção 2.4.1.2 destaca detalhes considerando o uso da RF.

2.4.1.1 Perceptron Multi-Camadas

Uma Perceptron Multi-Camadas (MLP) é um grafo de unidades conectadas de forma a representar um modelo matemático dos neurônios biológicos. Essas unidades são usualmente chamadas de unidades de processamento, nós ou simplesmente neurônios. As unidades são conectadas por meio de arestas com peso que representam a força da conexão entre cada uma delas. É um modelo inspirado no modelo biológico para representar as sinapses entre os neurônios [27].

Em geral, um conjunto de dados é processado nas unidades de entrada, intermediárias e a saída do classificador, tendo a resposta apresentada pelas unidades de saída. As unidades são representadas por modelos matemáticos inspirado no neurônio biológico.

No caso da PMC, os neurônios são organizados como na Figura 2.5, onde x_{11}, x_{12} e x_{13} representam as unidades de entrada, ou seja, unidades que representam a entrada do conjunto de dados na RNA que formam a camada de entrada. As unidades x_{21}, x_{22}, x_{23} e x_{24} são unidades de processamento que formam a camada intermediária ou oculta. A camada de saída corresponde às unidades x_{31}, x_{32} e x_{33} que também são unidades de processamento. Y_1, Y_2 e Y_3 representam a saída do processamento da RNA. Por fim, as unidades x_{10} e x_{20} representam unidades de auxílio chamadas *bias* usadas para acelerar a fase de treinamento da RNA.

Ainda se tratando da PMC, é necessário também definir o modelo de funcionamento das unidades de preprocessamento. Na maioria das vezes, a PMC usa dois tipos de unidades de processamento. Ambas têm como objetivo transformar um conjunto de entradas, que chegam por meio das ligações, em uma única saída. Usualmente, é feita uma soma ponderada das entradas com o peso de suas respectivas arestas. O valor da soma é então usado em função de ativação. A Figura 2.6 apresenta as duas funções de ativação que são usadas na PMC. Nas camadas ocultas uma função sigmoide ilustrada

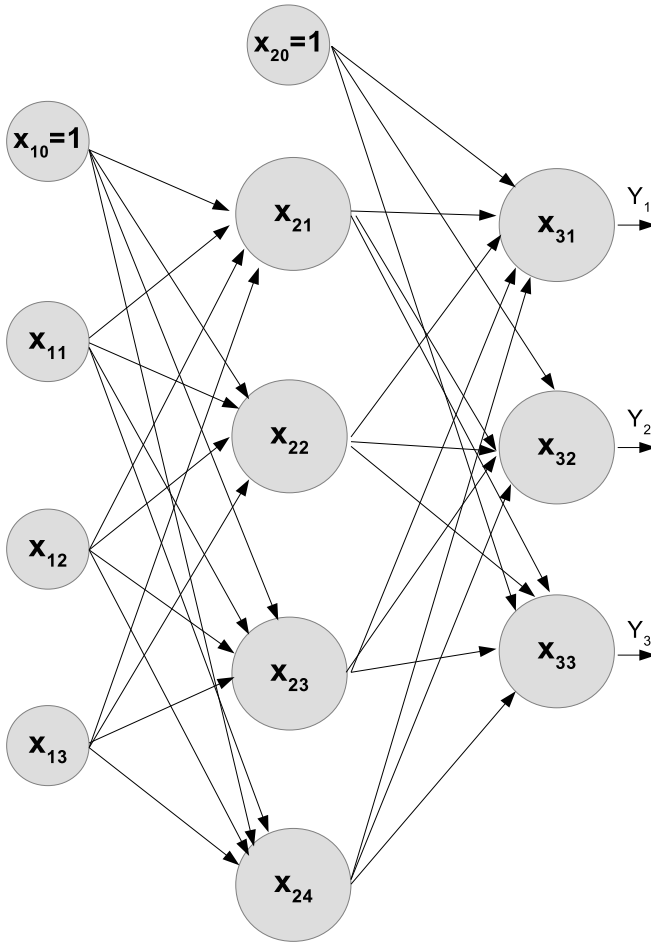


Figura 2.5 – Arquitetura da Perceptron Multi-Camadas, adaptado de Aggarwal [27].

na Figura 2.6a é empregada. Na camada de saída, por sua vez, é frequente o uso de uma função de ativação linear ilustrada na Figura 2.6b [22].

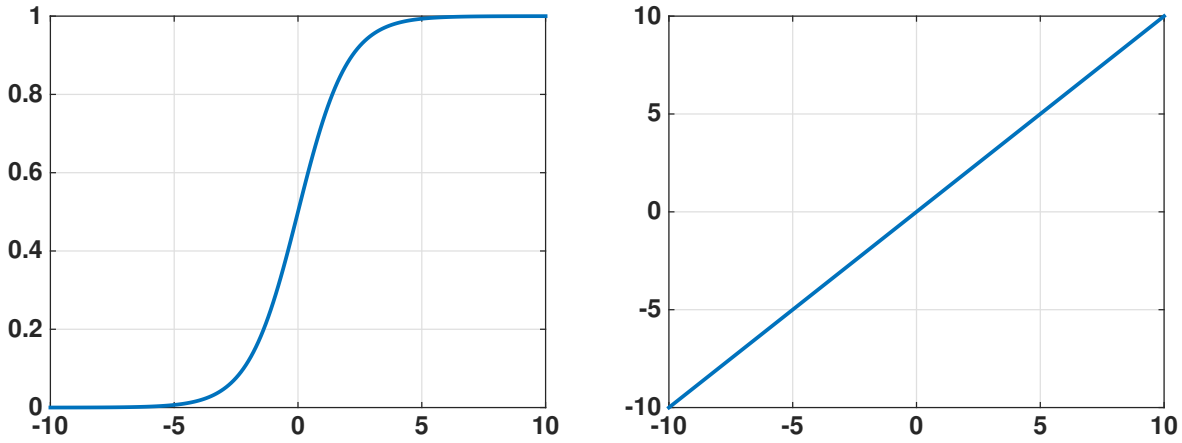
O cálculo da função sigmoide e da função linear é mostrado nas Equações 2.6 e 2.7, respectivamente, onde v é a soma ponderada de todas as entradas com seus respectivos pesos (Equação 2.5) e n é número de entradas na unidade de processamento.

$$v = \sum_{i=1}^{n} x_i * w_i \tag{2.5}$$

$$out(v) = v \tag{2.6}$$

$$out(v) = \frac{1}{1 + e^{-v}} \tag{2.7}$$

Para a fase de treinamento, o algoritmo para ajustes de pesos usado é a Retro-propagação (*backpropagation*). O objetivo do algoritmo é ajustar os pesos das arestas de



(a) Sigmoide. (b) Linear.

Figura 2.6 – Funções de ativação.

forma que o erro cometido na saída da RNA seja aceitável ou zero. Para tanto, o algoritmo de Retro-propagação é baseado no Método Gradiente Descendente juntamente com contador de épocas para encerrar o treinamento quando critérios como erro satisfatório for satisfeito [22, 27].

2.4.1.2 *Random Forests*

Random Forests é um método que consiste no uso de vários classificadores individuais, especificamente árvores, para alcançar melhor estabilidade. O método, por meio de um conjunto de árvores treinadas, efetua uma votação entre todas as árvores para exibir uma resposta. Inicialmente, todos as árvores disponíveis apresentam o mesmo peso para votação, embora a alteração para votação ponderada também seja possível. Em geral o processo de votação é feito como ilustrado na Figura 2.7 [28].

O modelo de treinamento para *Random Forests* é ilustrado na Figura 2.8 e é completamente descrito em [28]. Em geral, o conjunto de dados, assim como na maioria dos métodos de aprendizado máquina, é separado em duas partes: uma parte dos dados para o treinamento e uma outra parte para teste.

Porém, o conjunto formado pelos dados de treino é novamente repartido em duas partes: o conjunto *inBag* e o conjunto *outOfBag*. Um detalhe importante é que se o conjunto de treinamento possuir 1000 amostras de treino, o conjunto *inBag* deverá também possuir 1000 amostras de treino e aproximadamente 1/3 das amostras do conjunto de treinamento não deverão estar presentes no conjunto *inBag*. Basicamente, o conjunto *inBag* terá 1000 amostras baseada em duplicações e o 1/3 remanescente do conjunto de treinamento estará presente no conjunto *outOfBag*.

O objetivo dessa separação no conjunto de treinamento é para que seja possível

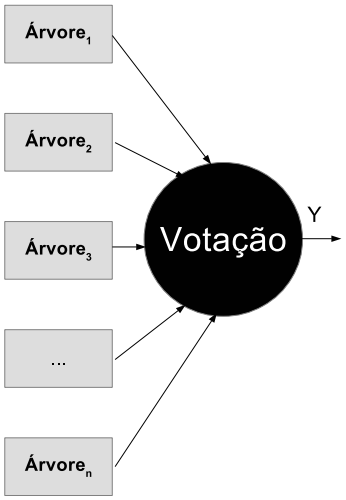


Figura 2.7 – Organização das árvores de uma floresta, adaptado de Livingston [28].

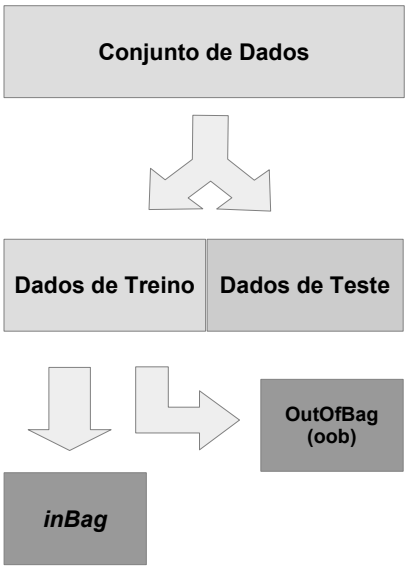


Figura 2.8 – Organização dos dados para treino da RF, adaptado de Livingston [28].

treinar uma árvore e efetuar testes sem que seja usado o conjunto de testes. Dessa forma, o conjunto *inBag* é usado para treinar cada uma das árvores individualmente enquanto o conjunto *outOfBag* é usado para testá-las sem fazer uso do conjunto teste. O processo de como é efetuada a separação do conjunto de dados é mostrado na Figura 2.8.

Um outro detalhe importante em relação ao treinamento das árvores da floresta são os atributos usados. Se os mesmos atributos e instâncias forem usados para treinar todas árvores, as árvores da floresta apontariam a mesma resposta sempre, perdendo a necessidade de voto. Para garantir respostas distintas entre grande parte das árvores, o conjunto *inBag* é dividido em B partes de mesmo tamanho. Cada árvore a ser treinada usará somente um dos B conjuntos (Figura 2.9).

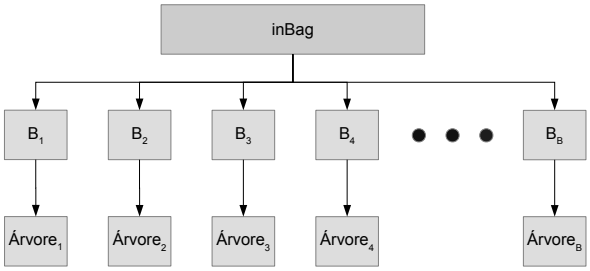


Figura 2.9 – Organização individual do treino de cada árvore.

Ainda em relação ao treino individual de cada árvore, um número m de descritores é selecionado aleatoriamente. Cada árvore irá considerar somente os m descritores. Desta forma, além de serem treinadas com exemplos possivelmente diferentes, descritores diferentes podem ser usados por diferentes árvores. Liaw e Wiener [29] apresentam os seguinte passos para o treino de uma floresta:

1. Para $b = 1$ até B:

   a) Retire um subconjunto amostras do conjunto *inBag*.

   b) Treine uma árvore sem efetuar podas com o conjunto de amostras retiradas seguindo os passos recursivamente:

      i. Selecione m descritores aleatóriamente do conjunto de descritores.

      ii. Selecione o melhor descritor entre os m selecionados para o nó.

      iii. Separe o nó.

Ao final do processo de treinamento das árvores individuais, o método segue como já discutido na Figura 2.7. O conjunto de testes é usado para avaliar o desempenho da floresta perante os dados, até então, não conhecidos. Cada árvore dispõe de uma resposta, a decisão final do classificador é o resultado de uma votação entre todas as árvores.

2.4.2 Matriz de Confusão

Segundo Aggarwal [27], uma das ferramentas mais usadas para avaliação de classificadores é chamada de matriz de confusão. Na prática, trata-se de uma matriz quadrada $M_{i,j}$ de dimensão K, onde K é número de classes. Uma entrada $M_{i,j}$, denota a quantidade de vezes que uma instância c_i foi classificada como c_j. Em um caso binário, é chamado de verdadeiro positivo (VP) e verdadeiro negativo (VN) o número de instâncias classificadas corretamente como positivo e negativo, respectivamente. Falso negativo (FN) denota o número de instâncias que foram classificadas como negativas, mas são positivas. Por sua vez, falso positivo (FP) denota o número de instâncias que foram classificadas como postivos, mas são negativas.

Tabela 2.1 – Exemplo dos resultados de predição de um classificador para um problema binário, retirado e adaptado de Aggarwal [27].

x	**r(x)**	**c(x)**
x_1	P	P
x_2	P	P
x_3	N	P
x_4	P	P
x_5	N	N
x_6	P	P
x_7	P	N
x_8	N	N
x_9	P	P
x_{10}	N	P

Tabela 2.2 – Exemplo de matriz de confusão, retirado e adaptado de Aggarwal [27].

	P	N
P	5(VP)	1(FN)
N	2(FP)	2(VN)

Por exemplo, a Tabela 2.1, apresenta os resultados obtidos de um classificador c para um problema binário. A primeira coluna representa as instâncias, a segunda coluna representa a classe real de cada instância e, por fim, a terceira coluna representa a predição feita pelo classificador. É possível observar que o valor de VP é 5, pois, houve 5 vezes em que uma instância de classe positiva foi predita como classe positiva. Da mesma forma, houve 2 vezes em que uma instância negativa foi predita como negativa, portanto VN é 2. A Tabela 2.2 ilustra os valores obtidos através das predições. As entradas que não pertencem a diagonal principal representam os erros cometidos pelo classificador.

Embora diversas medidas para avaliação de um classificador possam ser obtidas a partir da matriz de confusão, a principal estatística usada neste trabalho para avaliação de um classificador é a acurácia. Na prática, a acurácia representa a proporção de instâncias preditas corretamente em relação ao total de instâncias preditas. Para o exemplo das Tabela 2.1 e 2.2 o valor da acurácia é 0,7%. A Equação 2.8 mostra como obter o valor da acurácia.

$$\text{Acurácia} = \frac{VP + VN}{VP + VN + FN + FP} \tag{2.8}$$

2.4.3 Validação Cruzada

Dougherty [25] descreve a validação cruzada como um método de avaliação criado para situações em que há um número limitado de exemplos para treinamento disponível. Portanto, pesquisadores seriam tentados a usar o conjunto de dados inteiro para treino.

Entretanto, o resultado seria um modelo que não generaliza corretamente os dados e os resultados satisfatórios, na prática, não seriam obtidos. Neste trabalho a utilização da validação cruzada é apresentada na Seção 5.1.

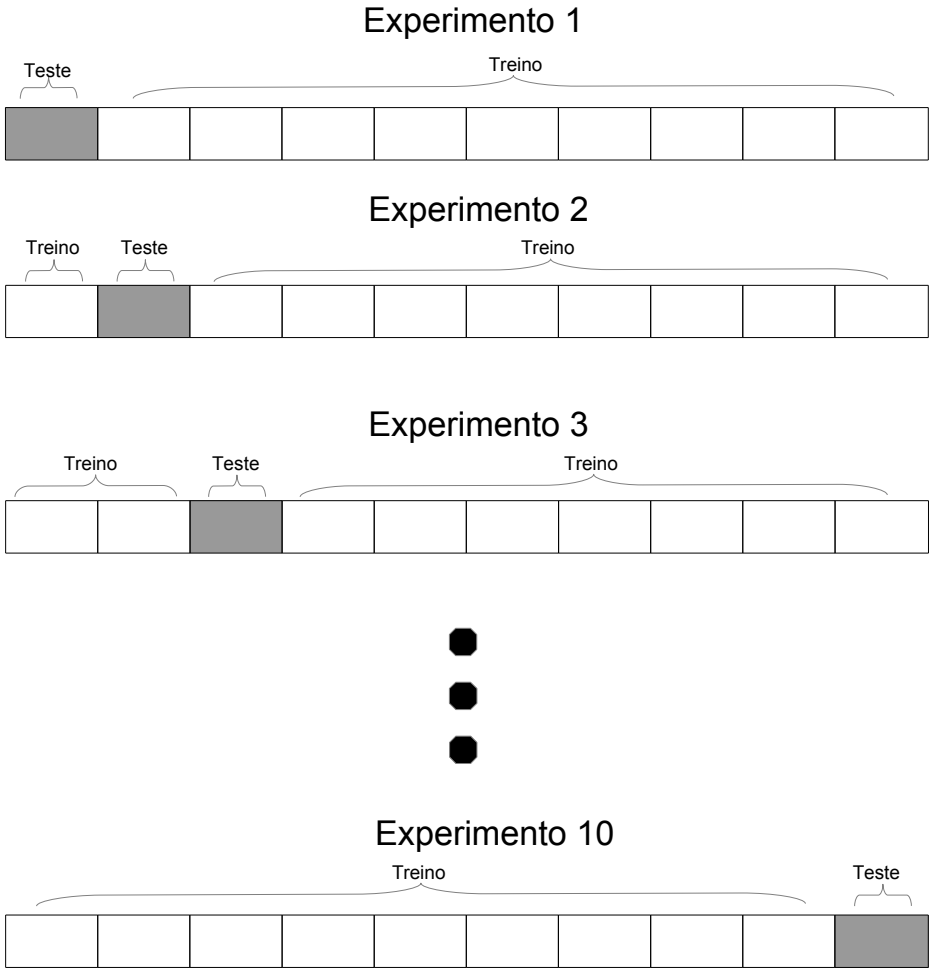


Figura 2.10 – Exemplo de validação cruzada para K=10, retirado e adaptado de Dougherty [25].

Validação cruzada é o método de avaliação onde parte do conjunto de dados é removido antes do treinamento. O restante do conjunto de dados é usado para treino e a parte inicialmente removida é usada para testes.

Uma variação da validação cruzada é o método *k-fold* (Figura 2.10). Nesta variação o conjunto de dados é dividido em K partes de mesmo tamanho em exemplos. Uma das partes é usada para treino e o restante para testes. Esse procedimento é repetido K vezes. Para cada uma das vezes a acurácia é calculada. A acurácia final passa a ser a média das acurácias obtidas.

3 TRABALHOS RELACIONADOS

Este capítulo apresenta os trabalhos relacionados encontrados na literatura. A Seção 3.1 apresenta trabalhos que abordaram os problemas acerca de diversas fraudes em RSD. Entre tais problemas foram abordadas: detecção de *spammers*, contas falsas de paródia e *bots*. A Seção 3.2 apresenta brevemente trabalhos que abordaram a aplicação de TDW em imagens e, posteriormente, trabalhos que abordaram TDW e MT.

3.1 Fraudes em Redes Sociais Digitais

Conforme descrito brevemente no Capítulo 1, as RSD apresentam dificuldades para solucionar diversos tipos de fraudes e cibercrimes. Uma vez que a definição usada na literatura para cibercrime abrange qualquer atividade maliciosa em que o computador ou rede seja um meio ou alvo, diversos cibercrimes estariam presentes nas RSD. Entre eles, temos: assuntos relacionados a proteção da privacidade dos usuários, propagandas indesejadas como *spams*, contas falsas relacionadas a celebridades, entre outros. Na maioria dos casos, os usuários dos serviços disponibilizados pelas RSD são os mais prejudicados. Em casos de *phishing*, o usuário pode perder o acesso a própria conta ao clicar em um *link*falso cujo destino é malicioso. Em casos de paródias de figuras públicas ou mesmo contas falsas cujo único objetivo de existência é a divulgação de conteúdo malicioso, o usuário continua sendo o maior prejudicado.

Para ilustrar tais problemas, a Figura 3.1 apresenta exemplos de contas maliciosas e como o mesmo tipo de conta maliciosa pode ser usada para mais de um tipo atividade. Na literatura, as contas maliciosas são divididas nos seguintes grupos [30, 31, 16]:

- *Spammers*: Contas usadas para a divulgação em massa de conteúdo repetitivo.
- Contas de *Phishing*: Contas usadas para a exibição de conteúdo fraudulento com o objetivo de obter informações privadas dos usuários.
- Contas de paródia: Contas falsas que se passam por celebridades.
- Contas falsas: Uma conta que não corresponde a uma pessoa física existente.
- *Bots*: Contas cujo o comportamento é controlada de forma automática.

As áreas de intersecção da Figura 3.1 representam quando uma conta de um determinado tipo pratica atividades de outra. Por exemplo, um *bot* pode ser usado para a prática de *phishing* e *spam*. Uma conta falsa pode ser usada apenas como uma conta falsa manipulando números de seguidores, como também pode ser uma conta falsa e ainda ser

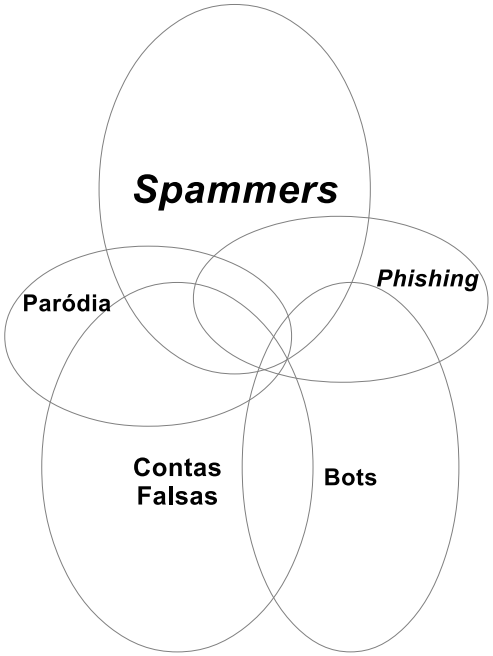


Figura 3.1 – Contas maliciosas e RSD.

usada para a prática de *phishing* e *spam*. Considerando tais problemas de segurança em RSD, trabalhos que abordaram alguns tipos de contas mais presentes foram propostos. As próximas seções descrevem uma revisão da literatura a respeito de soluções desenvolvidas principalmente em relação a MT e MD.

Em relação ao modelo proposto nesta dissertação, a principal contribuição para a literatura que aborda segurança em RSD é a identificação de contas com postagens automáticas descritas na Seção 3.1.3. Como descrito no paragráfo anterior e ilustrado na Figura 3.1, uma conta com comportamento automático (*bot*) pode ser usada para diversos tipos de fraudes.

Ao considerar uma comparação de estratégias encontradas na literatura, o modelo proposto nesta dissertação apresenta um objetivo similar ao proposto por Chu *et al.* [16] que é a classificação das contas em RSD em três classes: Humanos, *Cyborgs* e *Bots*. Porém, a contribuição em relação a estratégias encontradas na literatura é que o modelo proposto precisa somente de texto para classificar contas em RSD. Um detalhe importante é que o modelo proposto, então, pode ser aplicado em diversas RSD diferentes, uma vez que nenhuma informação em relação a uma RDS específica é usada no modelo.

Uma terceira contribuição para os trabalhos relacionados é feita para a literatura através do uso de TDW. O uso de TDW para MT já foi proposto por outros como apresentado na Seção 3.2.2. Porém, neste trabalho apresentamos o uso de TDW para a classificação textual do conteúdo produzido por contas de RSD.

3.1.1 Detecção de *Spams* em RSD

O empenho desprendido para solucionar a presença de *spams* em RSD foi percebido por meio da diversidade de trabalhos encontrados, por exemplo, Song *et al.* [32] e Lumezanu e Feamster [33]. Um exemplo de trabalho mais simples foi proposto por Grier *el al.* [15] que usou apenas uma lista de *links* maliciosos como gabarito para verificar se uma conta postava conteúdos maliciosos. Os autores concluíram que somente usar a lista de *links* maliciosos não é o suficiente para detectar *spammers*. Sob uma perspectiva de análise de perfil, Lee *et al.* [30] propuseram uma análise de estudo dos perfis de *spammers*. Martines-romo e Araujo [34] fizeram a proposta de um sistema completo que faz uso de Aprendizado de Máquina e cujo conjunto de dados é pré-classificado a partir de uma lista de *links* maliciosos. Portanto, ao usar um grande conjunto de postagens que possuíam algum *link*malicioso, o sistema se auto-abastece para extrair características que classificam postagens futuras independente da lista de *links*.

É possível propor uma solução para detecção de *spammers* baseada puramente em MT. Um exemplo dessa abordagem é mostrado por Benevenuto *et al.* [35]. Para detectar *spammers*, os autores levaram em consideração diversos descritores textuais como quantidade de palavras por postagem, relação de URLs por palavras, quantidade de caracteres, *links*, *hashtags* e usuários mencionados por postagem. Juntamente com os descritores textuais, os autores também consideraram características que descrevem o comportamento da conta, por exemplo, relação da quantidade de usuários seguidos por usuários seguidores, idade da conta e quantidade de postagem semanal e diária.

Posteriormente, ao se tratar de uma abordagem que faz uso de Aprendizado de Máquina, cada usuário é representado por um vetor de características (neste caso, numérico) onde cada posição do vetor possui o valor de uma característica. A solução, então, generaliza um modelo de classificação baseado em um conjunto de usuários previamente classificados para depois aplicar o conhecimento obtido na predição de futuros usuários em duas classes: *spammers* e *não-spammers*. Os autores escolheram Máquinas de Vetores Suporte como algoritmo de classificação e conseguiram resultados satisfatórios.

Um modelo puramente baseado em MT para detectar *spams* é mostrado por Santos *et al.* [36]. Os autores propõem um modelo baseado em conteúdo a nível de caracteres. Para tanto, o conteúdo de cada conta é representado por um documento D. Considerando que também se trata de um problema de classificação binária (*spammers* e *não-spammers*), os documentos D são separados em dois conjuntos. Durante o treinamento cada D previamente classificado é utilizado pelo algoritmo para a construção de um modelo de predição em nível de caracteres. O objetivo dessa abordagem é verificar o grau de semelhança de uma futura postagem comparado com as postagens usadas em treinamento para predizer a classe de tal mensagem. Nos experimentos, os autores verificaram que o modelo proposto obteve resultados semelhantes aos modelos tradicionais de classificação de texto

que incluem o uso de algorítimos como *K-Nearest Neighbors*, *Naive Bayes* e Máquinas de Vetores de Suporte.

3.1.2 Detecção de contas falsas

Embora contas falsas possam fazer parte do problema de detecção de *spams* mencionado anteriormente, um outro tipo de dano que pode ser causado por contas ilegítimas é a manipulação de números. Por exemplo, como apresentado anteriormente, RSD representam um meio para que empresas possam divulgar prestação de serviço e produtos. Particularmente, é comum que a popularidade real de uma empresa esteja relacionada à sua popularidade nas RSD. Portanto, celebridades e marcas tem como objetivo relacionar-se nas RSD com a quantidade máxima de usuários possíveis. Nesta situação, é muito frequente a contratação de contas falsas vendidas por terceiros para que se relacionem com um dado perfil comercial [31]. Soluções diversas já foram propostas: modelos baseados em grafos de relacionamentos por Cao *et al.* [37] e até mesmo defesas nativas, como foi apresentado por Stein *et al.* [38] e implementado no Facebook.

Uma proposta apresentada por Fong *et al.* [10] baseada em MD reúne um conjunto de informações sobre o perfil de um usuário para identificá-lo como falso ou legítimo no Facebook. Exemplos de informações usadas são: idade, gênero, educação, avatar do perfil, número de amigos e completude do perfil. Da mesma forma que a maioria das abordagens de MD e MT, cada instância de perfil é representada por um vetor de características. Por meio do experimento realizado no Facebook, os autores concluíram que o J48, um algorítimo de árvore de decisão, foi o algorítimo que atingiu melhores resultados com acurácia próxima aos 80%.

Uma abordagem semelhante, aplicada a uma base de dados do Twitter, é mostrada por Cresci *et al.* [39]. Os autores propõem um vetor de características para o perfil baseada em informações como: presença do nome do perfil; imagem de avatar; endereço físico; sistema operacional utilizado (IOS ou Android); se está sincronizado com uma conta de Facebook, Instagram ou Foursquare; se foi mencionado por outro usuário. Nesse trabalho um resultado interessante é a diferença encontrada entre uma conta falsa e um *spammer*. Uma conta falsa costuma usar uma quantidade de *links* muito menor que a de um *spammer*.

Jiang *et al.* [31] descreveram contas falsas como um grande conjunto de contas ilegítimas vendidas para se relacionarem e promoverem a popularidade da conta de terceiros. Assim sendo, os autores defendem a ideia que contas denominadas "*Zombies*" se comportam da mesma maneira, pois são um grupo de contas vendidas por empresas para promoverem a conta de um cliente. Portanto, um exemplo de característica usada pelo autores é que as contas "*Zombies*" possuem uma quantidade similar de relacionamento entre si (os próprios clientes).

3.1.3 Detecção de contas com postagens automáticas

Contas com postagens automatizadas, popularmente conhecidas como *bots*, também causam problemas para as RSD. Dependendo do objetivo de quem os controla, os *bots* podem estar relacionados a problemas de disseminação de *spams* como também podem ser usados para manipular números, por exemplo, no caso de contas falsas. Entretanto, *bots* nem sempre representaram um problema. *Bots* legítimos disseminam um volume grande de postagens benignas, como no caso de noticiários e outras situações emergenciais. Neste caso, os *bots* cumprem exatamente o propósito de algumas RSD como o Twitter, cuja a ideia é ser uma rede de compartilhamento de informações. *Bots* maliciosos, por outro lado, são ferramentas criadas com o único propósito de praticar ações danosas [16].

Juntamente com a existência de *bots*, RSD como o Twitter apresentam a existência de usuários intermediários entre *bots* e humanos, os *cyborgs*. Devido a ferramentas conectadas as contas das RSD que quando autorizadas pelo usuário postam conteúdo automaticamente, parte dos usuários passam a se comportar como um intermediário entre *bots* e *humanos* [16].

Como um passo inicial, o trabalho de Chu *et al.* [16] é focado no estudo do comportamento entre as três classes: Humanos, *Cyborgs* e *Bots*. Para tanto, os autores estudaram uma diversidade de descritores que poderiam identificar um padrão entre os três. Entre os descritores foram incluídos: plataforma de postagem (móvel ou outra qualquer), quantidade de postagem em um intervalo de tempo entre as três classes, quantidade de relacionamentos e quantidade de tempo entre postagens.

Posteriormente, Chu *et al.* [11], fizeram uma proposta de classificação com o uso de RFs. Usando boa parte dos descritores relatados no trabalho anterior, os autores foram capazes de classificar as contas entre as três classes com acurácia satisfatória e, ao mesmo tempo, conseguiram verificar a existência de postagens maliciosas como *spams*.

Trabalhos dedicados ao comportamento de contas com postagens automáticas também foram realizados. Messias *et al.* [40] demonstraram por meio de experimentos que o uso de *Bots* é uma forma simples para se tornar popular em RSD e, dessa forma, manipular estatísticas em relação a popularidade para fraudar campanhas de propaganda. Basicamente, os autores usaram *Bots* que, em primeiro momento, apenas reproduziam informações disponíveis em sites de notícias. Com o decorrer do tempo, o número de contas relacionadas ao *Bots* era considerável.

De maneira bastante semelhante, Edwards *et al.* [41], realizaram um estudo usando *bots*. Os experimentos também utilizaram contas que apenas reproduziriam conteúdos já disponibilizadas em noticiários. Porém, para a surpresa dos autores, as postagens realizadas pelos *bots* obteve respostas por partes de usuários humanos. Como conclusão, os autores apresentaram que *bots* podem ser facilmente adquirir confiança de usuários huma-

nos. Justamente por adquirir a confiança de demais usuários com facilidades, os autores consideraram o uso de *bots* essencial para divulgação de notícias importantes.

Por fim, um estudo de *bots* realizados no Facebook apontou resultados preocupantes em relação a segurança. Embora o Facebook se trate de uma RSD cuja as informações não são públicas como no caso do Twitter, com o uso de *bots*, Yazan *et al.* [14] demonstraram que seria completamente possível obter informações privadas dos usuários. Isso seria possível porque os usuários interagiram facilmente com os *bots* e, portanto, a disseminação de conteúdo malicioso seria efetuada com facilidade.

3.2 Transformada Discreta Wavelet

Até o presente momento, a TDW foi usada em diversos campos [42]. Portanto, a Seção 3.2.1 apresenta aplicações diversas de TDW como na compressão e redução de ruído. Também são apresentados trabalhos que aplicaram TDW em áudio e imagens. Seção 3.2.2 apresenta trabalhos que relacionaram TDW e MT. São mostrados exemplos de Recuperação de Informação (RI) e classificação de documentos.

3.2.1 Aplicações gerais

Em Processamento Digital de Imagem (PDI), TDW foram usadas para elaborar propostas de proteção autoral em imagens. Para tal, seria necessário inserir dados de forma imperceptíveis ao olho humano, que identifique a legitimidade de uma imagem como feito por Lin e Lin [43] com o uso de TDW. Ao transformar uma imagem para o domínio wavelet é possível efeturar alterações que não são visualmente perceptíveis. Um dos desafios dos autores era identificar o melhor posicionamento no dominio wavelet para efetuar alterações. Baixas frequências são facilmente percebidas visualmente quando alteradas. Altas frequências, por outro lado, são alteradas para compressão. Portanto, os autores decidiram usar as frequências médias e, em experimentos, verificaram que foi uma boa solução.

Um outro exemplo de PDI foi mostrado por Tedmori e Al-Najdawi [44]. Os autores também trabalharam no domínio wavelet. Em prática, a imagem cifrada consiste em embaralhar o domínio da frequência de forma que somente o destinatário e o remetente conheceriam e, portanto, qualquer forma de ataque *man in the middle* não seria bem sucedida. Uma vez alterada no domínio da frequência por um invasor, a imagem ao ser decifrada não ficaria reconhecível. Por meio de experimentos os autores verificaram que o modelo proposto foi bem sucedido.

Redução de ruído, um uso clássico de TDW [45], foi proposto por Han e Chang [46]. Se tratando puramente de PDS, o objetivo dos autores era identificar qualquer tipo comportamento anormal nas médias e altas frequências. Para testes, os autores artificial-

mente inseriram ruído nos sinais. Por fim, o sinal obtido pelo método deveria ser próximo ao sinal original. Os autores foram bem sucedidos em experimentos.

Compressão de imagens, segundo Walker [45] a maior aplicação de TDW em PDI, foi estudada detalhadamente por Wang *et al.* [47]. Segundo os autores, existem duas formas para comprimir imagens na literatura: a) manter apenas as frequências baixas e b) detectar um limiar que indique quais são as frequências que devem ser mantidas. Ambos as abordagens consistem em aplicar a TDW na imagem original. Uma vez obtida a imagem no domínio wavelet as frequências não desejadas são anuladas. Como apresentado nos parágrafos anteriores, as altas frequências em PDI são pouco perceptíveis visualmente.

3.2.2 Aplicações em Mineração de Texto

Para resolver problemas relacionados a texto, a TDW pode ser usada em uma variedade de formas. Na literatura, é possível encontrar trabalhos relacionados à recuperação de informação (RI), classificação de documentos e visualização de texto.

Em RI, os trabalhos foram iniciados por Park *et al.*. Inicialmente, Park *et al.* [48] não fizeram uso de TDWs, mas da Transformada Discreta de Cosseno. Os autores apresentaram a representação de texto através de sinais. Tal abordagem também é usada neste trabalho. Para representar um documento como um conjunto de sinais, é necessário um conjunto de termos. Para cada termo, um sinal é criado. O documento, então, passa a ser representado pelos conjuntos de sinais. A vantagem de usar sinais, vetores numéricos, é que um sinal carrega em cada entrada a frequência do termo sinalizado em uma determinada parte do documento. Em experimentos, os autores verificaram que documentos relavantes em relação a um conjunto de termos apresentam ocorrência dos termos na mesma parte do documento.

Posteriormente, Park *et al.* [49] realizaram testes com Transformada Discreta de Fourier (TDF). Nesta versão, a mesma representação em sinais foi usada. Apenas a transformada a ser usada foi alterada. Os autores concluíram que, em questões de acurácia, não há grande diferença entre usar TDC ou TDF, porém, por critérios de performance a TDC poderia ser usada.

Por último, Park *et al.* [18] fizeram uso de TDW. Também mantendo a mesma abordagem dos dois trabalhos anteriores, os autores verificaram que ao usar TDW os resultados melhoravam em acurácia e performance. Se tratando de uma abordagem de *ranking*, as TDW também possibilitavam várias possibilidades devido a multi-resolução e a quantidade de famílias diponíveis. Em experimentos, os autores testaram Daubechies-4 e Haar. Em acurácia, Daubechies-4 obteve o melhor resultado.

O modelo proposto por Park para representar texto como um conjunto de sinais foi posteriormente usado em RI para páginas Web. Purwitasari *et al.* [50] adaptaram a

representação de documentos em sinais para páginas Web. A abordagem foi muito similar a proposta por Park com TDF, mas desta vez aplicada em páginas Web. Em experimentos efetuados na Wikipedia[1], os autores concluíram que seria necessário considerar também mais aspectos além do *ranking* obtido pela TDF.

Portanto, em um segundo trabalho, Purwitasari *et al.* [51] propuseram um método híbrido. Nesta nova abordagem características adicionais como quantidade de acesso foram levadas em consideração. Tal abordagem foi nomeada de *PageRank Scoring* pelo autores e obteve bons resultados.

Considerando classificação de documentos. Thaicharoen *et al.* [52] também fizeram uso da representação de Park. Ao invés de usar um domínio da frequência ou um domínio wavelet, os autores decidiram por usar o resultado da TDW como descritores para algoritmos de aprendizado de máquina. Os autores usaram SVM e os testes foram realizados em 2 bases diferentes. Os resultados indicaram que a abordagem foi bem sucedida.

Uma abordagem de classifiçação de documentos diferente de Park foi proposta por Xexeo *et al.* [53]. Os autores representaram a base de texto por uma matriz onde as linhas representavam os termos e as colunas os termos. Cada entrada da matriz corresponderia a valor do *Term Frequency-Inverse Document Frequency* (TF-IDF). Posteriormente, cada linha seria tratada como um sinal para aplicar TDW. O resultado da TDW de cada sinal seria então a característica a ser processada por um algoritmo de aprendizado de máquina, no caso *K-Nearest Neighbors* (K-NN).

Weng *et al.* [54] propuseram um método que faz uso de TDW para detecção de eventos no Twitter. Os autores identificam termos chaves sobre as postagens de cada usuário e cada entrada do sinal é representada pela frequência daquele termo em uma janela de tempo. A informação cronológica é obtida através da hora da postagem. Ao agrupar sinais que sinalizam o mesmo termo os autores identificam um evento de acordo com o tamanho do agrupamento. Experimentos foram realizados com o objetivo de analisar a eleição na Singapura em 2011. Os resultados obtidos pelos autores foram satisfatórios e reconhecidos pela mídia do país.

[1] https://en.wikipedia.org/wiki/YIQ

4 MODELO PROPOSTO

Neste Capítulo, é apresentada a abordagem proposta neste trabalho que tem como objetivo solucionar um problema de classificação em RSD para classificação de contas. Um dos aspectos mais importantes da abordagem está relacionado em como descrever as amostras avaliadas de forma adequada por meio de seus atributos.

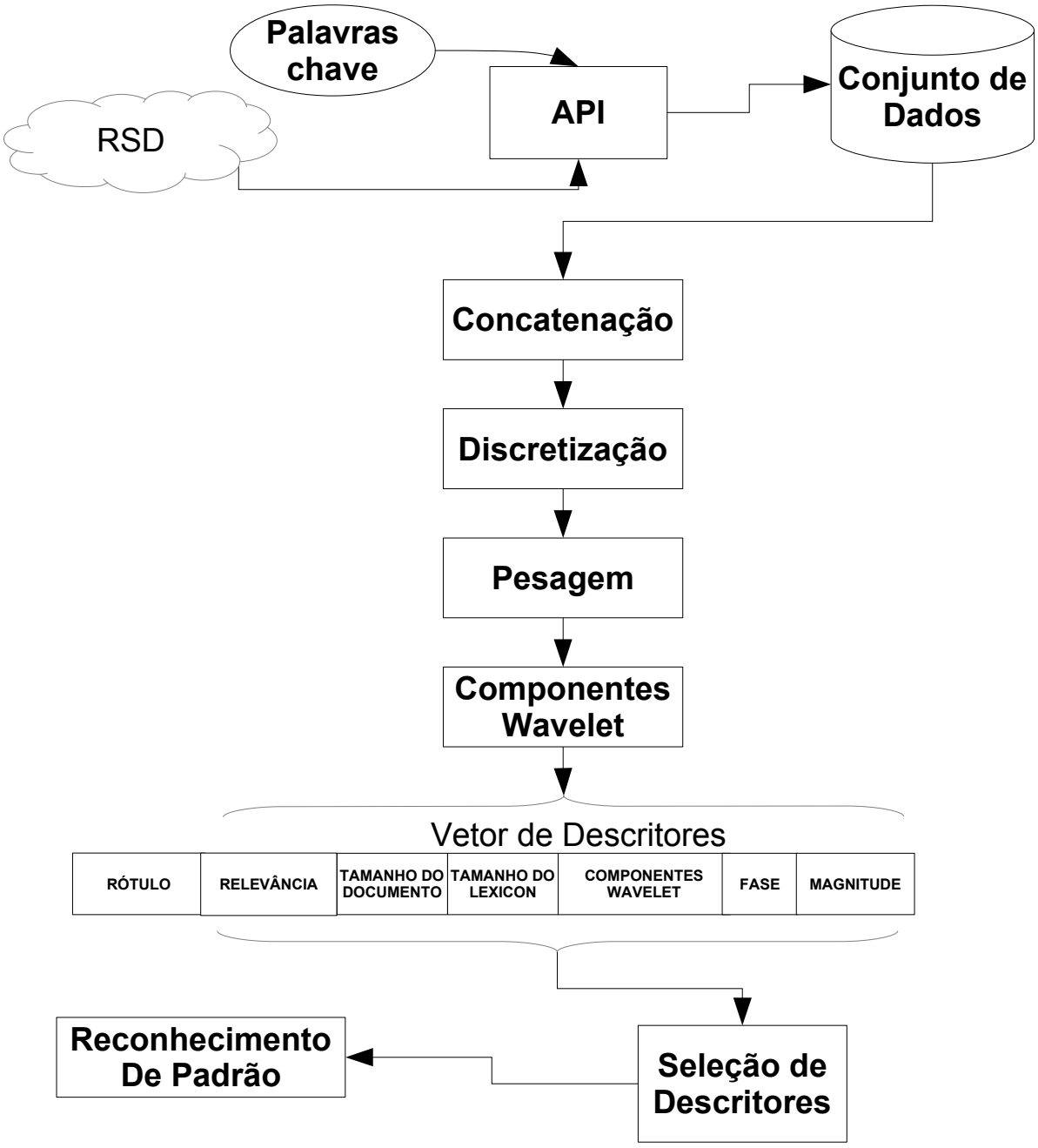


Figura 4.1 – Visão geral do modelo proposto.

Com esse objetivo, a abordagem propõe um modelo composto por cinco etapas,

como apresentado pela Figura 4.1. Estas etapas são: Concatenação, Discretização, Pesagem e Cálculo dos Componentes Wavelets. Após realizadas estas etapas, um vetor com os descritores é obtido.

O vetor resultante pode ter comprimento relativamente extenso, com elementos que para determinado problema ou base não contribuam para a fase de classificação. Em razão disto, uma etapa de Seleção de Descritores é realizada para diminuir a dimensionalidade dos vetores antes do processo de classificação (incluindo treinamento e avaliação).

Na etapa de Reconhecimento de Padrões não é especificado um classificador, o que se propõe é o uso de modelos supervisionados de Aprendizado de Máquina.

É importante destacar a nova proposta do cálculo de pesagem intitulado *Length-Based Coefficient Attenuation* (LBCA), que junto aos demais estabelecidos pela literatura foram avaliados e apresentados no Capítulo 5.

Assim, este Capítulo contempla as etapas do modelo proposto, como ilustrado na Figura 4.1. Um exemplo numérico do vetor de descritores usado é mostrado na Figura 4.4 e discutido ao longo deste capítulo.

4.1 Concatenação

A ideia inicial do modelo proposto é representar os usuários de RSD como uma coleção de documentos. Para obter uma estratégia de MT baseada na TDW que descreva de forma adequada as contas em RSD, o conteúdo textual produzido por um usuário requer uma representação de sinais baseados em termos similar ao proposto por Park [17]. Portanto, no modelo proposto, após representar as contas de RSD como uma coleção de documentos, cada usuário será representado como uma coleção de sinais.

No contexto das RSD, algumas definições são necessárias para explicar como representar cada conteúdo textual produzido por cada usuário como um documento de MT.

Seja $u \in U$ um usuário pertencente a um conjunto de usuários e $p \in P$ a parte textual de uma postagem em um conjunto de postagens. Um documento d, de um usuário u, corresponde ao conjunto de postagens p, onde $p.user == u$ como mostra a Equação 4.1. Portanto, o documento d pode ser visto como uma concatenação de todas as postagens feitas por um único usuário [55]. A Tabela 4.1 ilustra esse processo, onde as três primeiras linhas representam as postagens feitas por um único usuário. Por sua vez, a última linha da tabela representa as três postagens do usuário concatenadas sucessivamente.

$$Documento(d, u) = \{p | p.user == u\} \tag{4.1}$$

Tabela 4.1 – Exemplo de concatenação.

Identificador	Conteúdo
tweet#1	Parabéns pelo jogo, pelo gol, por representar o Brasil, por jogar com raça e amor à camisa, por orgulhar o povo brasileiro @neymarjr #bom_Diiiia #DeusnocomandoSempre #viaçãoUtil #copa2014 #job #utilnaCopa #Util #Brasil
tweet#2	Só por que a copa começa no dia 12, o dia dos namorados tem que ser outro dia, wtf?
tweet#3	Esquece a copa gente tem festa junina
documento#1	Parabéns pelo jogo, pelo gol, por representar o Brasil, por jogar com raça e amor à camisa, por orgulhar o povo brasileiro @neymarjr #bom_Diiiia #DeusnocomandoSempre #viaçãoUtil #copa2014 #job #utilnaCopa #Util #Brasil Só por que a copa começa no dia 12, o dia dos namorados tem que ser outro dia, wtf? Esquece a copa gente tem festa junina

4.2 Discretização

Posteriormente à etapa de concatenação, é efetuada a Discretização. Originalmente proposta por Park *et al.* [56], esta etapa tem como principal objetivo transformar um documento d em um conjunto de vetores discretos. Basicamente, transforma-se d em um conjunto de sinais baseados em termos. Por sua vez, um sinal baseado em termos corresponde a uma sequência de valores que representa a ocorrência de um termo em uma porção do documento. Computacionalmente, o sinal é representado por um vetor numérico. Cada elemento deste vetor, chamado de *bin*, representa uma porção de d. Estes *bins* identificam o número de ocorrências do termo t em cada porção, como é visto na Equação 4.2

$$\widetilde{f}_{d,t} = [f_{d,t,0}, f_{d,t,1}, ..., f_{d,t,b}, ..., f_{d,t,n-1}] \tag{4.2}$$

A Equação 4.3 ilustra um exemplo onde d é representado pelas ocorrências do termo t. Portanto, t, neste caso "copa", apareceu 1 vez na quarta porção (ou seja, *bin*). O termo também apareceu uma vez na quinta e sétima porção do documento. Nas demais porções do documento, o termo não foi encontrado. Nos trabalhos desenvolvidos por seus criadores, a medida foi fixada em 8 porções [56, 17, 18]. Porém, os documentos eram estruturados. Para páginas Web, 16 *bins* foram usados como medida mais adequada [51].

Nesta etapa, tokenização é efetuada. Para dividir um documento em partes de mesmo tamanho em palavras, é necessário trasnformar d em um vetor onde cada entrada corresponde a um único termo. O critério usado para separação de termos foi somente

a presença de espaço em branco. Portanto, como mostra a Figura 4.2, “#bom_Diiiia” é representado como um termo. Caracteres como “,”, “:”, entre outros também foram removidos.

Um detalhe importante é que cada *bin* possui tamanho $\left\lceil \frac{tamanho\_do\_documento}{numero\_de\_bins} \right\rceil$. No caso do exemplo apresentado, o número de bins usado foi 8 e o documento d possui 57 termos ($tamanho\_do\_documento$), como ilustrado na Figura 4.2. O termo sinalizado foi “copa” e está destacado em negrito. Portanto, o tamanho de cada *bin* corresponde a 8 termos ($\left\lceil \frac{57}{8} \right\rceil$), com exceção do último *bin* que pode ser menor. Com o objetivo de facilitar a visualização, a Figura 4.2 apresenta os termos pertencentes a cada *bin* distinto em uma célula diferente.

Para o modelo proposto neste trabalho, a quantidade de *bins* proposta é de 32. Os detalhes sobre o processo de avaliação e definição de tal valor pode ser encontrado no Capítulo 5.2.

Parabens	pelo	jogo	pelo
gol	por	representar	o
Brasil	por	jogar	com
raça	e	amor	a
camisa	por	orgulhar	o
povo	brasileiro	@neymarjr	#bom_Diiiia
#DeusnocomandoSempre	#viacaoUtil	**#copa**	#job
#utilnaCopa	#Util	#Brasil	So
por	que	a	**copa**
comeca	no	dia	12
o	dia	dos	namorados
tem	que	ser	outro
dia	wtf?	Esquece	a
copa	gente	tem	festa
junina			

Figura 4.2 – Exemplo de tokenização.

$$\widetilde{f}_{d,t} = [0, 0, 0, 1, 1, 0, 1, 0] \tag{4.3}$$

4.3 Pesagem

Esta etapa, chamada de Pesagem, é uma normalização dos valores que complementam a representação em sinais baseados em termos [51, 17, 57].

Entretanto, os esquemas de pesos encontrados na literatura foram criados com diferentes objetivos. Por exemplo, o esquema de peso criado por Arru *et al.* [57] foi criado para auxiliar na precisão de tarefas de MT com o objetivo de recomendação de conteúdo. Esquemas descritos nos trabalhos de Park *et al.* [18] foram todos criados e usados em tarefas de RI. Nenhum dos pesos encontrados na literatura foi criado com o objetivo de identificar a diferença de textos provenientes de humanos e máquinas.

Portanto, para buscar uma descrição do problema mais efetiva, um novo esquema chamado LBCA (*Length-Based Coefficient Attenuation*) foi proposto. Sua formula se encontra na Equação 4.4, onde τ_d é o tamanho do vocabulário do documento (como mostra a Seção 4.5) d e $\bar{\tau}_d$ é a média dos tamanhos de todos os vocabulários. Como pode ser percebido, o LBCA foi elaborado com um objetivo em particular: distinguir conteúdos produzidos entre humanos e máquinas. O esquema proposto aposta em uma diferença específica: o número de palavras distintas usadas em um texto humano é maior do que em um texto produzido por *bot*, ainda que se tratem do mesmo assunto. Isto é consequência do próprio uso informal da linguagem e, portanto, enfatizar o número de palavras distintas pode ser um auxílio mais efetivo na distinção de textos produzidos por *bots*. Após esta etapa, o vetor numérico contendo frequências de termos deve se tornar um vetor com frequências alteradas por pesos, como visto na Equação 4.5.

$$w_{d,t,b} = f_{d,t,b} \cdot \tau_d / \bar{\tau}_d \tag{4.4}$$

$$\widetilde{w}_{d,t} = [w_{d,t,0}, w_{d,t,1}, ..., w_{d,t,b}, ..., w_{d,t,n-1}] \tag{4.5}$$

Segundo Park [17], não é possível exemplificar numericamente a pesagem. Uma vez que pesagem pode ser interpretada como uma forma de normalização, são necessários diversos exemplos para ilustrar a necessidade da pesagem. Porém, pesagem é uma etapa fundamental para obter resultados satisfatórios. A Figura 4.3 ilustra a etapa de pesagem.

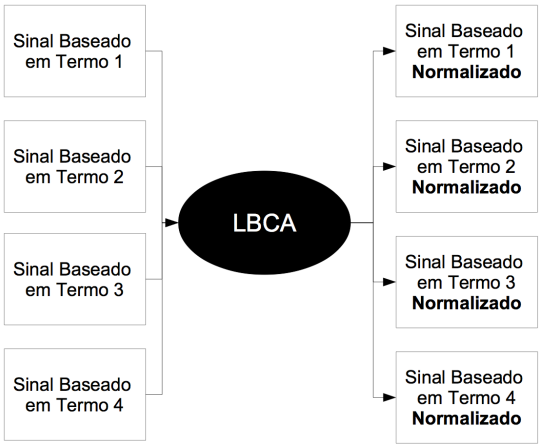


Figura 4.3 – Visão geral da etapa de pesagem.

4.4 Wavelets

As famílias da TDW são funções matemáticas que decompõe os dados em diferentes componentes interpretáveis como frequência. Comparáveis com a tradicional TDF, a TDW representa uma escolha preferencial para analisar sinais não-estacionários que contenham descontinuidade e picos [17]. Neste trabalho, os sinais provenientes da etapa anterior de pesagem, $\widetilde{w}_{d,t}$, são decompostos usando a TDW a fim de obter $\widetilde{\zeta}_{d,t}$. A partir desta etapa, um documento, ou seja, um conjunto de *tweets* em d, produzidos pelo usuário u é representado por um conjunto de sinais do espectro contendo componentes Wavelets. A Equação 4.6 ilustra tal etapa.

$$\begin{aligned}\widetilde{\zeta}_{d,t} &= TDW(\widetilde{w}_{d,t}) \\ &= [\zeta_{d,t,0}, \zeta_{d,t,1}, ..., \zeta_{d,t,b}, ..., \zeta_{d,t,n-1}]\end{aligned} \tag{4.6}$$

Se considerar o exemplo das seções anteriores sem a pesagem, o resultado desta etapa é ilustrado pela Equação 4.7, onde $d = 1$ e $t = copa$. Como apresentado na Seção 2.3.1, cada um dos componentes wavelets obtidos através da transformação do sinal possui um significado relevante com relação a distribuição de frequência. Por exemplo, o primeiro componente ($\frac{3}{\sqrt{8}}$) que, em seu numerador (3), apresenta a soma da ocorrência do termo "copa" no documento 1. Os demais componentes da TDW apresentam a distribuição de frequência do termo em alguma parte do sinal. O segundo componente ($\frac{-1}{\sqrt{8}}$) apresenta, em seu numerador (-1), a diferença entre os 4 primeiros termos e os 4 últimos termos do sinal baseado em termo original. O terceiro e o quarto componentes representam as diferenças entre os quartos dos sinais e os 4 últimos componentes wavelets apresentam as diferenças entre os oitavos dos sinais, respectivamente.

Para o modelo proposto neste trabalho, focado em detectar automação de texto no Twitter, é proposto o uso da Daubechies 4. A análise completa de tal recomendação é encontrada no Capítulo 5.2.

$$\widetilde{\zeta}_{d=1,t=copa} = [\frac{3}{\sqrt{8}}, \frac{-1}{\sqrt{8}}, \frac{-1}{\sqrt{4}}, \frac{0}{\sqrt{4}}, \frac{0}{\sqrt{2}}, \frac{-1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}] \tag{4.7}$$

4.4.1 Componentes de Magnitude

Como apresentado na Seção 4.4, os componentes wavelets apresentam informações sobre a distribuição de frequência de um termo no decorrer do documento. Essa diferença, pode ser representada por qualquer valor positivo ou negativo.

Um complemento para os componentes wavelets são os componentes de magnitude. Como apresentando por Park [17], é interessante guardar também apenas o módulo dos componentes wavelets. As magnitudes podem ser vistas como uma versão mais simples

dos componentes wavelets, pois, não apresentam informações sobre a diferença da distribuição de frequência como no caso dos componentes que apresentam valores positivos ou negativos.

$$H_{d,t,b} = |\zeta_{d,t,b}| \tag{4.8}$$

Exatamente como proposto por Park *et al.*[17] e mostrado por meio da Equação 4.8, os componentes de magnitude são obtidos por meio de uma simples aplicação da função módulo em cada componente wavelet. Para o exemplo usado no decorrer deste Capítulo, os componentes de magnitudes obtidos são mostrados na Equação 4.9.

$$H_{d=1,t=copa,b} = [\frac{3}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{4}}, \frac{0}{\sqrt{4}}, \frac{0}{\sqrt{2}}, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}] \tag{4.9}$$

4.4.2 Componentes de Fase

Os componentes de fase, da mesma forma que os componentes de magnitude, complementam os componentes wavelets. O objetivo dos componentes de fase é indicar apenas em que parte do documento a diferença de frequência foi positiva ou negativa. Mesmo com normalizações, componentes de magnitude podem apresentar diferenças relevantes entre documentos diferentes. Nestes casos, os componentes de fase são relevantes pois apresentam valores somente entre 0 e 1.

$$\phi_{d,t,b} = \frac{\zeta_{d,t,b}}{H_{d,t,b}} \tag{4.10}$$

Também proposto inicialmente por Park [17], os componentes de fase são obtidos por meio da checagem do sinal de cada componente wavelet, como mostrado na Equação 4.10.

$$\phi_{d=1,t=copa,b} = [+1, -1, -1, 0, 0, -1, +1, +1] \tag{4.11}$$

4.4.3 Relevância do Documento

A relevância de um documento com relação ao seus termos foi proposta em trabalhos de Park *et al.* [17, 18, 49]. Se trata de um valor numérico que busca atribuir uma nota para um documento que informa a relevância do mesmo em relação aos termos desejados para RI.

A relevância do documento obtêm valores maiores quando os termos desejados se encontram no mesmo *bin*. Portanto, os documentos relevantes não seriam encontrados

apenas ao considerar a frequência dos termos procurados. Para ser considerado um documento relevante, segundo este atributo, é necessário que os termos desejados se encontrem próximos.

Portanto, contas com comportamento maliciosos, semelhantes a *spammers*, que tem como objetivo manipular o uso e a popularidade de termos seriam representados por valores maiores.

Para calcular a sua relevância, são necessários os componentes de fase e de magnitude. Por exemplo, a Equação 4.12, onde $\#(T)$ é o número de termos usados, mostra um coeficiente usado para adaptar as magnitudes. Se os termos principais aparecem em *bins* diferentes, o valor $\bar{\Phi}_{d,b}$ descresce. Posteriormente, o valor de $\bar{\Phi}_{d,b}$, por sua vez, é usado na Equação 4.13 que apresenta um valor parcial do quão relevante um documento é em relação aos termos chave. Porém, justamente como apresentado em [17, 18, 49], o valor final da relevância do documento é mostrado na Equação 4.14.

$$\bar{\Phi}_{d,b} = |\frac{\sum_{t \in T} \phi_{d,t,b}}{\#(T)}| \tag{4.12}$$

$$s_{d,b} = \bar{\Phi}_{d,b} \cdot \sum_{t \in T} H_{d,t,b} \tag{4.13}$$

$$S_d = \sum_{i \in b} s^2_{d,b_i} \tag{4.14}$$

Para ilustrar as equações descritas acima, os próximos paragráfos apresentam uma adaptação do mesmo exemplo apresentado em Park [17].

$$\begin{aligned} F_1 &= [0,0,0,1,0,1,0,0] \\ F_2 &= [0,0,0,0,0,0,1,0] \end{aligned} \tag{4.15}$$

Onde, F_1 e F_2 são 2 sinais baseados em termos criados aleatoriamente. Os resultados de ambos os sinais após efetuada a TDW de Haar como descrito na Seção 4.4 são:

$$\begin{aligned} \widetilde{\zeta}_{d,t_1} &= [\tfrac{2}{\sqrt{8}}, \tfrac{0}{\sqrt{8}}, \tfrac{-1}{\sqrt{4}}, \tfrac{1}{\sqrt{4}}, \tfrac{0}{\sqrt{2}}, \tfrac{-1}{\sqrt{2}}, \tfrac{-1}{\sqrt{2}}, \tfrac{0}{\sqrt{2}}] \\ \widetilde{\zeta}_{d,t_2} &= [\tfrac{1}{\sqrt{8}}, \tfrac{-1}{\sqrt{8}}, \tfrac{0}{\sqrt{4}}, \tfrac{-1}{\sqrt{4}}, \tfrac{0}{\sqrt{2}}, \tfrac{0}{\sqrt{2}}, \tfrac{0}{\sqrt{2}}, \tfrac{1}{\sqrt{2}}] \end{aligned} \tag{4.16}$$

Para calcular os componentes de magnitde, aplica-se a Equação 4.8 em ambos os sinais $\widetilde{\zeta}_{d,t_1}$ e $\widetilde{\zeta}_{d,t_2}$:

$$\begin{aligned} H_{d,t_1} &= [\tfrac{2}{\sqrt{8}}, \tfrac{0}{\sqrt{8}}, \tfrac{1}{\sqrt{4}}, \tfrac{1}{\sqrt{4}}, \tfrac{0}{\sqrt{2}}, \tfrac{1}{\sqrt{2}}, \tfrac{1}{\sqrt{2}}, \tfrac{0}{\sqrt{2}}] \\ H_{d,t_2} &= [\tfrac{1}{\sqrt{8}}, \tfrac{1}{\sqrt{8}}, \tfrac{0}{\sqrt{4}}, \tfrac{1}{\sqrt{4}}, \tfrac{0}{\sqrt{2}}, \tfrac{0}{\sqrt{2}}, \tfrac{0}{\sqrt{2}}, \tfrac{1}{\sqrt{2}}] \end{aligned} \tag{4.17}$$

Para calcular os componentes de fase, aplica-se a Equação 4.10 em ambos os sinais $\widetilde{\zeta}_{d,t_1}$ e $\widetilde{\zeta}_{d,t_2}$:

$$\begin{aligned} \phi_{d,t_1} &= [+1, 0, -1, +1, 0, -1, -1, 0] \\ \phi_{d,t_2} &= [+1, -1, 0, -1, 0, 0, 0, +1] \end{aligned} \tag{4.18}$$

Por fim, para iniciar o cálculo da relevância de um documento, são necessários ϕ_{d,t_1} e ϕ_{d,t_2} para analisar se houve ocorrência dos termos desejados no mesmo bin. Substituindo ambos os componentes de fase na Equação 4.12, obtemos:

$$\begin{aligned} \bar{\Phi}_d =& \left| \frac{\sum_{t \in T} \phi_{d,t,b}}{\#(T)} \right| \\ =& \left| \frac{\phi_{d,t_1} + \phi_{d,t_2}}{2} \right| \\ =& \left| \frac{[+1, 0, -1, +1, 0, -1, -1, 0] + [+1, -1, 0, -1, 0, 0, 0, +1]}{2} \right| \\ =& \left| \frac{[+2, -1, -1, 0, 0, -1, -1, +1]}{2} \right| \\ =& \left| 1, -\frac{1}{2}, -\frac{1}{2}, 0, 0, -\frac{1}{2}, -\frac{1}{2}, \frac{1}{2} \right| \\ =& [1, \frac{1}{2}, \frac{1}{2}, 0, 0, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}] \end{aligned} \tag{4.19}$$

É importante notar que, sempre que existe ocorrência de frequência dos termos desejados no mesmo *bin*, o valor da entrada $\bar{\Phi}_d$ é 1. Do contrário, o seu valor decresce como é o caso da segunda e terceira entrada. Em ambos os casos, o valor da entrada é $\frac{1}{2}$ pois somente 1 dos termos ocorreu no respectivo *bin*. Para calcular a segunda etapa da relevância do documento como descrito na Equação 4.13, é necessário calcular a soma das magnitudes:

$$\begin{aligned} \sum_{t \in T} H_{d,t,b} =& H_{d,t_1} + H_{d,t_2} \\ =& [\frac{2}{\sqrt{8}}, \frac{0}{\sqrt{8}}, \frac{1}{\sqrt{4}}, \frac{1}{\sqrt{4}}, \frac{0}{\sqrt{2}}, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, \frac{0}{\sqrt{2}}] + [\frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{0}{\sqrt{4}}, \frac{1}{\sqrt{4}}, \frac{0}{\sqrt{2}}, \frac{0}{\sqrt{2}}, \frac{0}{\sqrt{2}}, \frac{1}{\sqrt{2}}] \\ =& [\frac{3}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{4}}, \frac{2}{\sqrt{4}}, \frac{0}{\sqrt{2}}, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}] \end{aligned} \tag{4.20}$$

Posteriormente, é efetuado o produto escalar de $\bar{\Phi}_d$ com $\sum_{t \in T} H_{d,t,b}$:

$$\begin{aligned} s_d =& \bar{\Phi}_d \cdot \sum_{t \in T} H_{d,t,b} \\ =& [1, \frac{1}{2}, \frac{1}{2}, 0, 0, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}] \cdot [\frac{3}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{4}}, \frac{2}{\sqrt{4}}, \frac{0}{\sqrt{2}}, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}] \\ =& [\frac{3}{\sqrt{8}}, \frac{1}{2\sqrt{8}}, \frac{1}{2\sqrt{4}}, 0, 0, \frac{1}{2\sqrt{2}}, \frac{1}{2\sqrt{2}}, \frac{1}{2\sqrt{2}}] \end{aligned} \tag{4.21}$$

Enfim, como mostrado na Equação 4.14, a relevância do documento é obtida ao calcular a soma dos quadrados de s_d:

$$\begin{aligned} s_d =& \sum_{i \in b} s_{d,b_i}^2 \\ =& [(\frac{3}{\sqrt{8}})^2, (\frac{1}{2\sqrt{8}})^2, (\frac{1}{2\sqrt{4}})^2, 0^2, 0^2, (\frac{1}{2\sqrt{2}})^2, (\frac{1}{2\sqrt{2}})^2, (\frac{1}{2\sqrt{2}})^2] = 1,59 \end{aligned} \tag{4.22}$$

Após as etapas acima descritas, o valor de s_d também é usado no vetor de descritores. O valor obtido no exemplo representa um valor mediano. Contas que postem os termos sinalizados muitas vezes sempre de forma similar podem alcançar maiores valores de s_d.

4.5 Tamanho do *lexicon*

O *lexicon*, também chamado por outro autores de vocabulário, é o conjunto formado pelos termos únicos presentes em um documento. Portanto, o tamanho do *lexicon* consiste apenas da quantidade de termos únicos presentes em um documento [58]. Neste trabalho, o tamanho do vocabulário também é usada no cálculo do LBCA, como visto na Equação 4.4. Para o exemplo usado até o momento, o tamanho do lexicon é 44. Portanto, existem 44 termos distintos em $d = 1$, como pode ser visto na Figura 4.2

4.6 Tamanho do documento

Ao considerar que, neste trabalho, um documento d corresponde a acumulação de várias postagens de um único usuário. O tamanho do documento d, corresponde ao número de termos presentes em d. Para o exemplo usado, o tamanho de d é 57, ou seja, a concatenação dos 3 *tweets* somaram 57 termos no total.

4.7 Seleção de atributos

O primeiro passo para garantir uma etapa de classificação com performance eficiente é o uso dos descritores corretos para a construção de um classificador. Neste trabalho, o algorítimo empregado é o *Correlation-based Feature Selection* proposto em [59]. Em geral, o algorítimo busca medidas que informam o quão relevante cada descritor, característica, seria para garantir a separabilidade entre as classes. Os autores fazem uso de medidas de mérito e confiança.

4.8 Reconhecimento de Padrões

Por fim, o vetor de descritores usado nesta proposta possui: componentes wavelets, componentes de fase, componentes de magnitude, relevância do documento, tamanho do

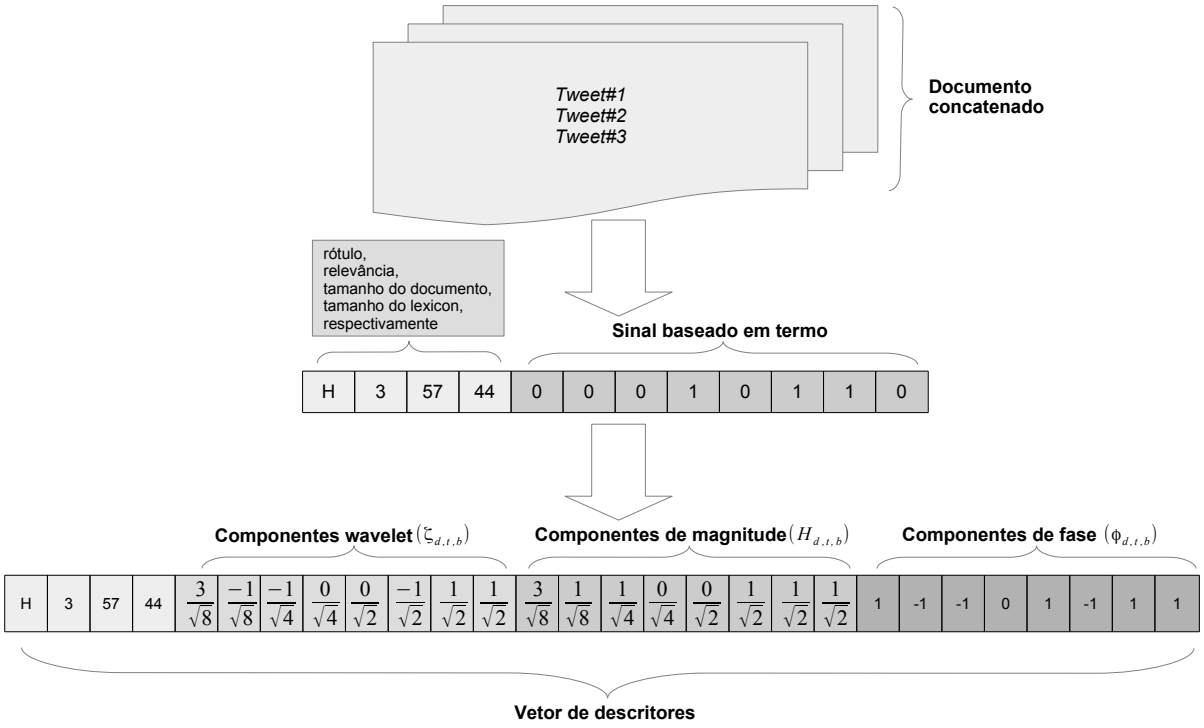


Figura 4.4 – Exemplo ilustrativo do vetor de descritores.

documento e tamanho do *lexicon*. Como especificado anteriormente, um algorítimo de seleção de atributos é empregado para melhorar a performance da etapa de classificação.

Em relação aos algorítimos de classificação, a literatura recomenda utilizar PMC e RF para ambientes de RSD. Por se tratar de uma proposta de MT em RSD, é necessário fazer uso de métodos e algorítimos que satisfaçam a necessidade de performance computacional. Por isso, na etapa de extração de descritores, TDWs foram empregadas no lugar de TDFs. As RFs têm sido muito usadas nos últimos anos em situações onde tempo de execução e acurácia precisam ser eficientes [60, 61]. Em situações nas quais o classificador deve se adaptar a novas instâncias e modelos não vistos na etapa de treinamento, PMCs são as recomendadas na literatura no passar dos recentes anos [62, 63].

5 EXPERIMENTOS E RESULTADOS

Este Capítulo apresenta a metodologia escolhida para realizar os testes do modelo proposto. Inicialmente, na Seção 5.1, são apresentadas informações gerais sobre a aquisição do conjunto de dados e detalhes sobre o tamanho da produção textual dos usuários em palavras e *tweets*. Posteriormente, são apresentados os parâmetros usados para cada camada do modelo proposto, incluindo: discretização, pesagem, TDW e classificadores. A segunda parte do Capítulo, a Seção 5.2, apresenta os resultados obtidos. Os resultados são discutidos isoladamente considerando as camadas do modelo proposto individualmente. Em seguida, resultados obtidos considerando uma discussão geral são apresentados.

5.1 Conjunto de dados e configuração dos experimentos

O Twitter, a RSD usada neste trabalho, diferente de outras RSD é conhecido pelo fato de seus usuários publicarem seus pensamentos por meio de pequenos textos, ou seja, não mais que 140 caracteres, usando a Web ou dispositivos móveis [1]. Esses pequenos textos, chamados de *tweets*, podem ser visualizados publicamente e são imediatamente disseminados entre os usuários relacionados com seu autor [64].

O grupo de desenvolvimento do Twitter oferece uma API para acesso dos dados relacionados aos *tweets*. Os dados usados como amostras nos experimentos foram coletados usando tal serviço[1]. Os dados coletados com o uso da API do Twitter contêm vários campos, por exemplo, número de identificação da mensagem, o ID do autor do *tweet*, o campo contendo o texto curto publicado e alguns outros campos de meta-dados [64]. Para este trabalho, o campo mais importante é o campo contendo o conteúdo da postagem que é usado para analisar a distribuição de frequência dos termos. Todos os dados recebidos da API do Twitter são armazenados em um banco de dados MySQL.

O conjunto de dados foi coletado durante a Copa do Mundo de 2014 para conter apenas *tweets* relacionados à Copa do Mundo. As palavras-chave usadas para a coleta foram: “BRASIL”, “COPA” e “COPA2014”. O conjunto de dados de *tweets* coletados foram escritos no idioma português (Brasil) e, uma vez que o evento é muito popular, foi possível encontrar os três tipos de classes: Humanos, *Bots* e *Cyborgs*. É possível verificar as postagens feitas por *bots* e humanos e ainda um comportamento híbrido dos *Cyborgs* na Tabela 5.1.

A identificação das instâncias de cada classe foram feitas de forma manual. Trabalhos encontradados na literatura [11, 65, 66, 67] que descreveram o comportamento das três classes foram usados como guia para fazer a classificação manual de cada instância.

[1] http://twitter4j.org/en/index.html

Tabela 5.1 – Exemplo de *tweets* coletados.

#	Conteúdo	Classe
1	**Copa** completa 13 dias e 7 seleções da América já estão garantidas na próxima fase http://glo.bo/1jfPIDf #G1naCopa pic.twitter.com/9DdXWs05IH	Bot
2	Trânsito no Distrito Federal é alterado para o jogo decisivo do **Brasil**. http://bit.ly/V5rXIW #**copa2014**	Bot
3	Quando eu falei, a Alemanha vai ganhar a **copa**, os cara deram risada kkkkkkkkkkkk	Humano
4	Pensei que o **brasil** ia fazer mais gols mais td bem kkk	Humano
5	I'm at Boulevard Londrina Shopping (Londrina, PR) w/ 3 others http://4sq.com/1qXI6dw **Brasil**, Londrina	Cyborg
6	tive a impressão de que estava impedido... impressão ASIODHAOSIDHIOASH #**Brasil**	Cyborg

Na prática, uma conta que não apresenta nenhuma postagem realizada de forma automática, como postagens oriundas de aplicativos como Foursquare[2], e que também não apresentam postagens repetidas ou muito semelhantes entre si, foram manualmente classificadas como Humano.

Por outro lado, uma conta que apresenta postagens com conteúdo muito similares entre si, ou que apresenta postagens cujo exato conteúdo foi encontrado em algum site, foi classificada como *bot*. Essas 2 características se referem aos *bots* que postam mensagens repetitivas ou *bots* que divulgavam notícias.

O terceiro caso, *cyborgs*, é classificado toda vez que uma conta com postagens similares as produzidas por humanos apresenta também uma ou mais ocorrências de postagens de aplicativos como o Foursquare.

Outro detalhe sobre o conjunto de dados coletado é o montante de texto produzido por usuário. Uma vez que cada usuário escreve quantidades diferentes de palavras por postagem e tal informação poderia ser relevante para a detecção da classe avaliada, nenhuma normalização na quantidade de palavras foi realizada. Os experimentos foram realizados com diferentes tamanhos de documentos entre usuários. Dados sobre o tamanho das amostras do conjunto de dados são encontrados na Tabela 5.2.

No total, o conjunto de dados usado é composto por 100 usuários. Os mesmos 100 usuários, ou seja, o mesmo conjunto de dados é usado em dois experimentos. Para o Experimento 1, os usuários encontrados foram pré classificados em três classes: 36 humanos, 36 *Cyborgs* e 28 *Bots*. Para o experimento 2, os usuários foram pré-classificados em duas classes: Humanos e Não-Humanos. Neste segundo caso, os usuários da classe Não-Humanos são os mesmo usuários do Experimento 1 que foram classificados como *Cyborgs* ou *Bots*. Portanto, no Experimento 2, 64 são *Não-Humanos* e 36 são Humanos.

[2] https://pt.foursquare.com/

Tabela 5.2 – Dados gerais sobre o conjunto de dados.

Número total de usuários	100
Quantidade média de palavras por usuário	746,49
Quantidade de palavras no conjunto de dados	74469
Maior documento do conjunto de dados	2155
Menor documento do conjunto de dados	81
Média de *tweets* por usuário	19,36
Total de *tweets* no conjunto de dados	1936

O objetivo de elaborar dois experimentos se deve principalmente ao auxílio em casos de abordagens que utilizam a Análise de Sentimento, que tem como objetivo identificar emoções em texto. Por exemplo, o Experimento 2, ao considerar *bots* e *cyborgs* como a classe Não-Humano, separa os usuários cuja a produção de textos não é automatizada. Esta separação seria de grande auxílio de preprocessamento, pois não seria de interesse analisar sentimentos de *bots* ou mesmo *cyborgs*. Garantiria-se que a análise foi feita apenas em usuários sem comportamento automatizado. Porém, em caso que as três classes distintas devessem ser separadas, os resultados do Experimento 1 atenderiam o requisito ao distinguir os usuários em três classes.

Embora o montante de usuários pareça pequeno para a realidade de milhões de usuários ativos no Twitter, um trabalho que também usou um montante de usuários similar é encontrado em [68]. Neste trabalho, também foi possível usar um montante pequeno uma vez que o número de comportamentos distintos não é muito grande.

Tabela 5.3 – Esquemas de pesos para sinais baseados em termos.

Nome (Descrito por)	Cálculo
BD-ACI-BCA (Park [17])	$w_{d,t,b} = \frac{1+log(f_{d,t,b})}{(1-s)+s \cdot W_d/\bar{W}_d}$
OKAPI	$w_{d,t,b} = \frac{f_{d,t,b}}{f_{d,t,b}+\tau_d/\bar{\tau}_d}$
SMART	$w_{d,t,b} = \frac{(1+log(f_{d,t,b}))/(1+log(\bar{f}_{d,t}))}{(1-s)+s \cdot \tau_d/\bar{\tau}_d}$
ARRU (Arru *et al.* [57])	$w_{d,t,b} = IPF_{t,b} \cdot CF_{d,b,t}$
LBCA	$w_{d,t,b} = f_{d,t,b} \cdot \tau_d/\bar{\tau}_d$

Tabela 5.4 – Famílias wavelets e seus suportes.

Família	Suporte
Haar	2
Daubechies	$4, 8, 16, 32$ *e* 64
Coyflets	$6, 12, 18, 24$ *e* 30
Symmlets	8 *e* 16
Beylkin	24
Vaidyanathan	18

As configurações de cada experimento foram formadas por diversas opções. Por exemplo, para discretização foram combinadas as seguintes quantidades de divisões (*bin*): 8, 16, 32, 64, 128, 256. Seria possível descobrir a melhor medida de discretização já que *tweets* são diferentes de páginas Web e documentos estruturados. Segundo a literatura, a quantidade de divisões utilizadas para páginas Web é de 16 *bins* [51] e para documentos estruturados 8 *bins* [56].

Com relação a pesagem, 5 esquemas de pesos encontrados na literatura juntamente com o proposto LBCA foram avaliados. Detalhes dos pesos são mostrados na Tabela 5.3. Todos os esquemas de peso já foram relatados na literatura, exceto um, que é o esquema de pesagem proposto pela abordagem. Diferentes famílias de TDWs foram exploradas como Haar, Daubechies, Coyflets, Symlets, Beylkin e Vaidyanathan com o objetivo de descobrir a família mais adequada e seu respectivo suporte para a descrição dos padrões de escrita. Na Tabela 5.4 é mostrada cada família com os respectivos suportes testados. Todas as famílias usadas nos experimentos foram escolhidas devido o sua reputação em diversos trabalhos como de Barbon *et al.* [69, 70]. As palavras sinalizadas foram as mesmas que as usadas para coletar os dados por meio da API do Twitter.

Antes da etapa de treinamento, o *Correlation-based Feature Subset Selection* proposto por Hall [59] foi adotado para a redução de dimensionalidade. Quatro arquiteturas de PMC e a RF foram testadas para obter a melhor classificação. As diferentes arquiteturas de PMC testadas são referentes as variações das quantidades de neurônios na camada oculta em relação à camada de entrada. Nos experimentos, as variações testadas incluem: a) o dobro da quantidade de neurônios na camada oculta; b) 1,5 da quantidade de neurônios na camada oculta; c) mesmo número de neurônios na camada oculta e d) metade da quantidade de neurônios na camada oculta. A etapa de treinamento foi então precedida uma vez para cada possibilidade de combinação entre discretização, pesagem, família wavelet e classificador.

Para a avaliação, foi adotada a validação cruzada. Desta forma, é garantido que cada configuração será testada diversas vezes usando diferentes partes do conjunto a cada rodada. As medidas da configuração são calculadas ao tirar a média do total de rodadas no teste. Um exemplo completo das configurações usadas é mostrado na Figura 5.1, onde

o LBCA é destacado entre os demais esquemas de pesos.

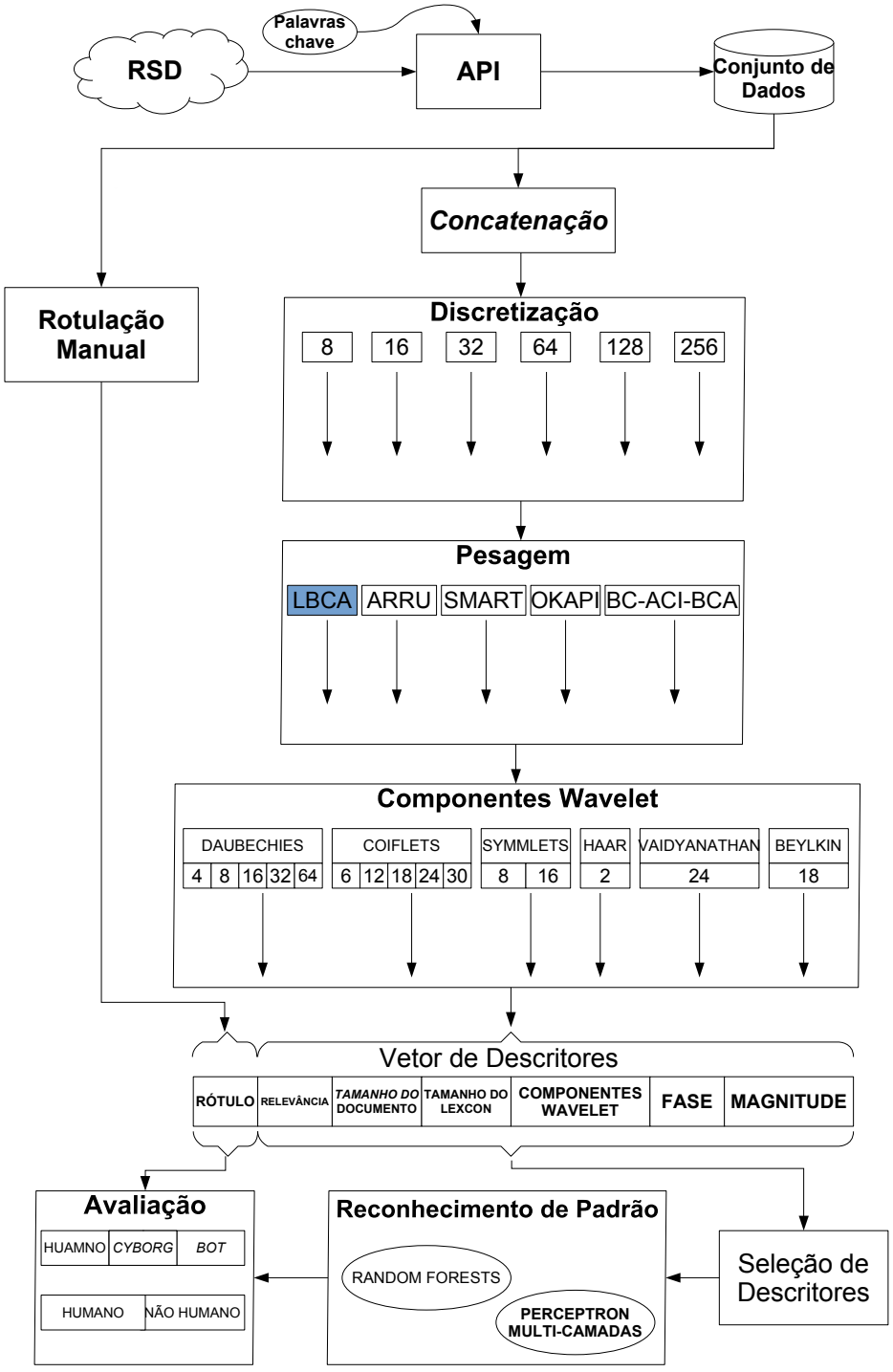


Figura 5.1 – Visão geral das configurações de experimento.

No total, 2250 testes foram realizados para ambos os experimentos. Esses testes cobriram todas as possíveis combinações entre 6 medidas de discretização (8, 16, 32, 64, 128 e 256), 5 esquemas de pesos diferentes (LBCA, ARRU, SMART, OKAPI, BC-ACI-BCA), 6 famílias de TDW em diferentes resoluções (15 possibilidades) e 2 classificadores (4 arquiteturas de PMC e RF).

Os resultados finais são computados em uma matriz de confusão para avaliar os classificadores. Por isso, um exemplo de uma configuração para avaliação do modelo seria: discretização em 8 partes, LBCA para pesagem, Transformada Wavelet Haar e *Random Forests* como classificador.

5.2 Análise e discussão

Esta Seção apresenta os resultados obtidos referentes aos Experimentos 1 e 2. Para cada um dos experimentos, 2250 testes foram efetuados variando cada combinação de Discretização, Pesagem, TDW e Classificadores.

Como apresentado inicialmente ao final do Capítulo 1, por meio de experimentos buscou-se efetuar testes que analizassem cada camada do modelo proposto. Portanto, primeiramente, são discutidos os resultados em relação a cada camada do modelo proposto de forma independente. Em seguida, são discutidos os resultados dos experimentos de forma geral.

5.2.1 Discretização

Dos 2250 testes efetuados para o Experimento 1, cada uma das 6 medidas de Discretização foram testadas 375 vezes variando entre as demais possíveis combinações entre Pesagem, TDW e Classificador. Em seguida a média da acurácia dos 375 testes foi calculada. Analogamente, a acurácia média dos 375 também foi calculada para o Experimento 2.

Os resultados dos experimentos em variação da quantidade usada na Discretização apresentam indícios baseados em acurácia média e desvio padrão que 8, 16 e 32 partes são escolhas factíveis para um cenário real. A Figura 5.2 mostra a média da acurácia, juntamente com o desvio padrão em preto, de cada quantidade usada. Os números entre parênteses na Figura 5.2 representam a média entre as acurácias dos Experimentos 1 e 2.

Este resultado é independente da família wavelet usada, esquema de peso e classificador. Discretização em 8, 16 e 32 partes apresentaram médias de acurácias sensivelmente maiores, com 91,22%, 92,00% 91,30%, respectivamente. Ainda, as 3 quantidades apresentaram baixo valor de desvio padrão, que é outro indício significante em relação a homogeneidade dos resultados, independente dos demais parâmetros. Outra vantagem em relação aos valores de Discretização menores é o impacto no tamanho do vetor de descritores. Quanto menor o valor de N, menor o tamanho do vetor, consequentemente, menor espaço de armazenamento e melhor performance.

Para 64, 128 e 256 partes, a acurácia média apresentou um gradiente descendente. Tal fato é justificável considerando que quanto mais partições, mais descritores são apre-

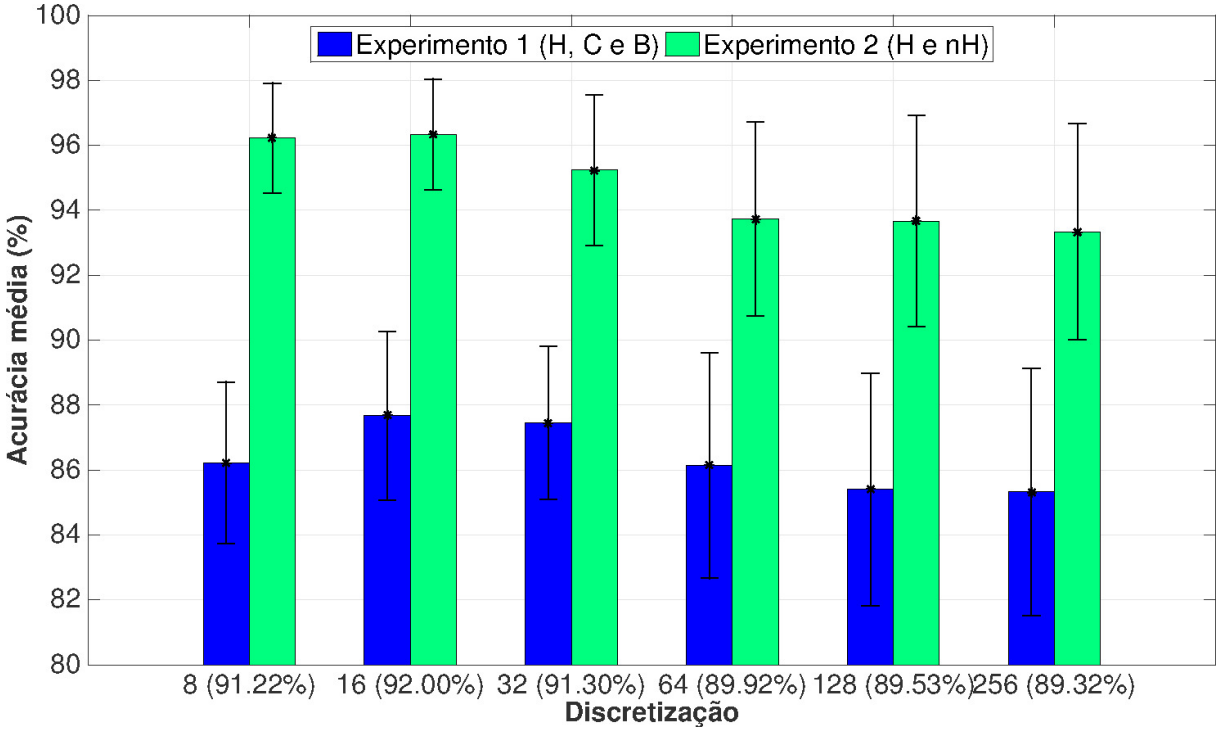


Figura 5.2 – Discretização e acurácia média para Experimentos 1 e 2.

sentados. Mais descritores que o necessário, entretanto, podem comprometer a capacidade de generalização dos classificadores.

5.2.2 Pesagem

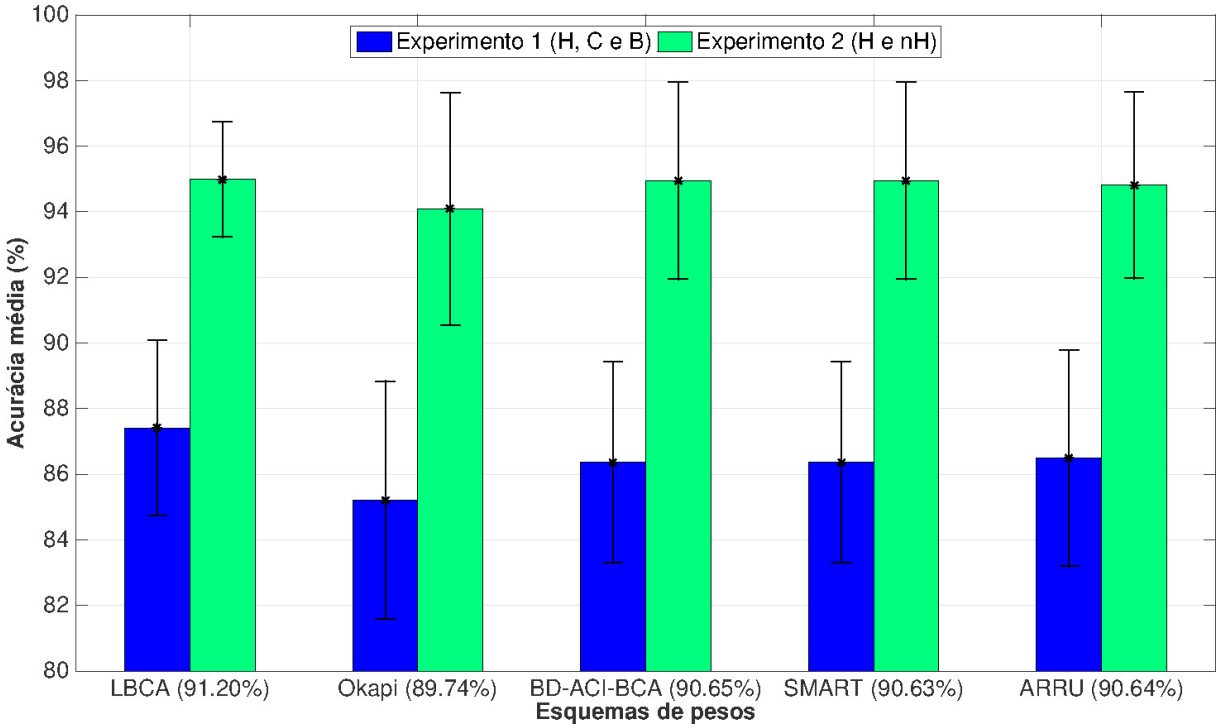


Figura 5.3 – Esquemas de pesos e acurácia média para Experimentos 1 e 2.

Os resultados com relação a pesagem e os esquemas de pesos são ilustrados por meio da Figura 5.3. Analogamente ao caso da discretização, cada um dos 5 esquemas de pesos foi testado 450 vezes por experimento. Os resultados também são ilustrados em acurácia média e desvio padrão.

Embora todos os esquemas de pesos testados apresentaram resultados semelhantes em acurácia média, o esquema de peso proposto neste trabalho, o LBCA, apresentou acurácia média de 91,20%. Tal acurácia, é sensivelmente maior que os demais esquemas de pesos, por exemplo, BD-ACI-BCA (90,65%) ,SMART (90,63%) e ARRU (90,64%).

Os resultados levemente superiores do LBCA também apontam indícios que o tamanho do *lexicon* e o tamanho do documento podem ser relevantes como descritores. Esse indício se dá ao fato do LBCA fazer uso de ambos os descritores em sua fórmula, como descrito anteriormente na Tabela 5.3.

5.2.3 Wavelets

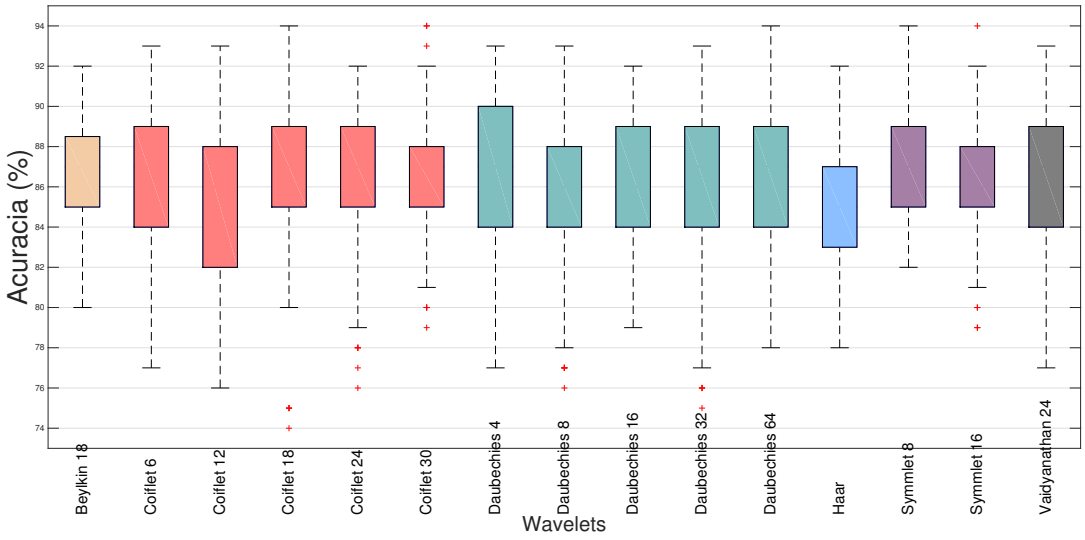
Na Figura 5.4, boxplots foram usados para mostrar os resultados sob outra perspectiva: wavelets. Membros de mesma família são ilustrados na mesma cor e a presença de *outliers* é vista como sinais de cruz vermelha. Nas abordagens de MT, os trabalhos relacionados usaram apenas Haar e Daubechies [18, 51]. Por isso, neste trabalho, foram levadas em consideração outras famílias.

Para calcular o boxplot de cada uma das 15 possibilidades de escolha de TDW foi testada 150 vezes. Portanto, os boxplot apresentam as acurácias alcançadas por cada TDW individualmente.

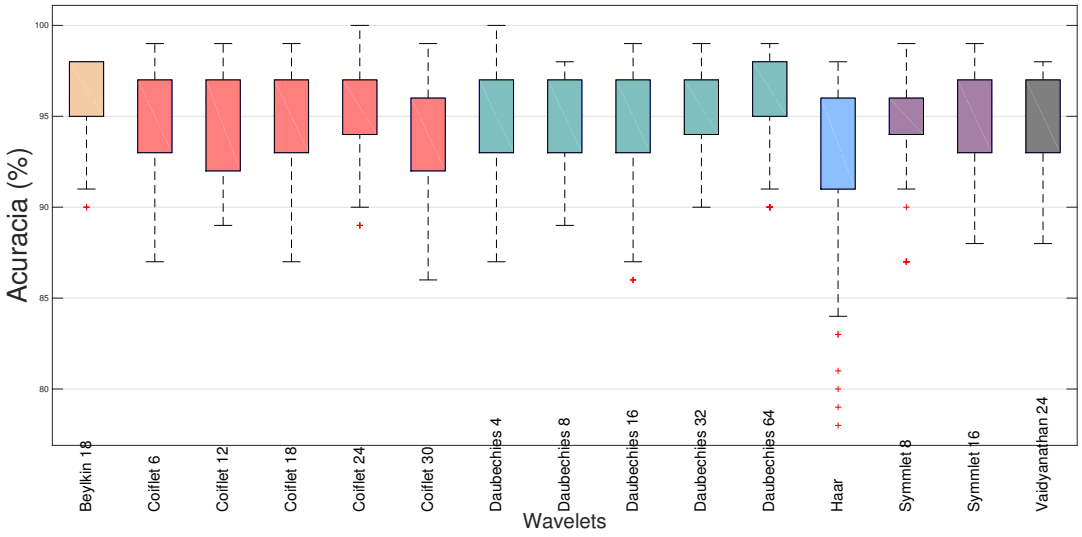
Sobre os resultados, Coiflets e Daubechies mostram maiores acurácias. Em uma configuração específica, alcançaram 94% de acurácia para o experimento 1 (Figura 5.4a) e 100% para o experimento 2 (Figura 5.4b). Entretanto, ao levar em conta o tamanho simétrico de quartis, que indica estabilidade em relação a acurácia, Daubechies 4 mostra um bom resultado em relação a ambos os experimentos.

Considerando a família Coiflet, os resultados mostram que suporte 6 e 12 alcançaram bons resultados enquanto suportes maiores resultaram na ocorrência de *outliers* como visto na Figura 5.4a. Um outro caso de bom resultados é a Vaidyanathan.

As Symmlets apresentaram bons resultados máximos em ambos os experimentos, embora tenham apresentado *outliers* em ambos experimentos. Vaidyanathan, Haar, e Beylkin apresentaram resultados simétricos para o experimento 1, mas não para o experimento 2. No geral, os resultados mostram que, em um experimento entre humanos e não humanos, Haar, conhecida pela sua simplicidade [69], pode não garantir a apropriada descrição do problema. Beylkin é específica para áudio [69], e assim, os resultados não significativos para texto são justificáveis.



(a) Experimento 1: Humano, *Cyborg* e *Bot*.



(b) Experimento 2: Humano e Não-humano.

Figura 5.4 – Wavelets e acurácia

Este é um resultado importante, uma vez que trabalhos da literatura usaram somente Haar e Daubechies para MT. Os resultados mostram que além de Daubechies 4, Coiflets 6 e Coiflets 12 alcançaram bons resultados de mediana, quartis quase simétricos e sem ocorrência de *outliers*

5.2.4 Seleção de descritores

Na maioria dos casos, os descritores selecionados após a aplicação do *Correlation-based Feature Subset Selection* proposto por Hall [59] variava de acordo com as demais escolhas como TDW, discretização e pesagem. Entretanto, 2 descritores estavam sempre presentes após a seleção de atributos: o tamanho do *Lexicon* e o tamanho do documento. Tal presença ilustra um boa escolha ao considerar o tamanho do *Lexicon* no esquema de

peso LBCA. Juntamente com as TDWs, ambos atributos se mantiveram importantes para a representação do problema.

5.2.5 Classificadores

Em relação aos classificadores, os resultados são apresentados na Tabela 5.5 para o experimento 1 e na Tabela 5.6 para o experimento 2. Ambas as tabelas apresentam os resultados dos 5 maiores em acurácia e dos 5 classificadores com menor acurácia.

É possível observar que ambos os classificadores PMC e RF obtiveram bons resultados em ambas as tabelas. As quatro possibilidades de configurações para a camada oculta alcançaram resultados satisfatórios.

Em relação aos resultados, uma acurácia maior que 87,0% foi alcançada quando distinguindo entre humanos, *cyborgs* e *bots*. Por exemplo, a RF obteve 88,7% de média acurácia juntamente com mediana de 89,0% e um máximo de 94,0%. Juntamente com 88,0% de acurácia média, as PMC apresentaram bons resultados. Ainda, ao considerar que as PMCs de menores arquiteturas alcançaram resultados melhores que as PMC de arquiteturas maiores, a perspectiva de manter o custo computacional sob controle se manteve.

Tabela 5.5 – Resultado dos classificadores para o Experimento 1.

Classif.	Média	Des. Pad.	Mediana	Máx.	Min.
RF	88,7%	2,13%	89,0%	94,0%	82,0%
PMC[9-5-1]	88,1%	2,34%	88,0%	92,0%	83,0%
PMC[13-20-1]	87,4%	2,40%	87,0%	91,0%	83,0%
PMC[9-9-1]	87,1%	2,06%	87,0%	92,0%	83,0%
PMC[13-7-1]	87,0%	2,01%	87,0%	90,0%	83,0%
PMC[46-46-1]	82,1%	2,63%	83,0%	87,0%	74,0%
PMC[57-57-1]	82,1%	2,43%	83,0%	87,0%	74,0%
PMC[57-81-1]	82,1%	2,43%	83,0%	87,0%	74,0%
PMC[34-34-1]	82,1%	1,94%	82,0%	85,0%	78,0%
PMC[41-62-1]	82,1%	2,35%	83,0%	85,0%	74,0%

Para a Tabela 5.6, no experimento 2, RF não foi presente nos resultados com 5 maiores acurácias (média: 96,6%, desvio padrão: 1,19%, mediana: 97%, máximo: 100%, mínimo: 93%), mas obteve acurácia máxima com uma configuração específica (discretização: 32, pesagem: LBCA, Wavelet: Daubechies 4). Arquiteturas simples, como no Experimento 1, obtiveram resultados melhores em termos de média de acurácia e desvio padrão. Portanto, assim no experimento 1, a perspectiva de baixo custo computacional se manteve.

O Experimento 2, justamente por ser mais simples e por se tratar de menos classes, alcançou resultados melhores que o experimento 1. Para trabalhos de Análise de Senti-

Tabela 5.6 – Resultado dos classificadores para o Experimento 2.

	Classif.	Média	Des. Pad.	Mediana	Máx.	Min.
	PMC[7-11-1]	97,0%	1,31%	97,5%	98,0%	94,0%
	PMC[8-4-1]	97,0%	1,69%	97,0%	99,0%	93,0%
	PMC[7-7-1]	96,9%	1,32%	97,5%	98,0%	94,0%
	PMC[8-16-1]	96,9%	1,81%	97,0%	99,0%	93,0%
	PMC[7-4-1]	96,9%	1,29%	97,0%	98,0%	94,0%
(*)	Random Forests	96,6%	1,19%	97,0%	100,0%	93,0%
	PMC[35-35-1]	90,8%	2,41%	91,0%	95,0%	87,0%
	PMC[35-35-1]	90,8%	2,41%	91,0%	95,0%	87,0%
	PMC[35-70-1]	90,7%	2,31%	91,0%	95,0%	87,0%
	PMC[35-53-1]	90,7%	2,37%	91,0%	95,0%	87,0%
	PMC[38-38-1]	90,7%	2,94%	90,0%	96,0%	86,0%

(*): Caso excepcional para a RF

mento, este resultado seria o mais importante. Principalmente, como no caso excepcional da RF que alcançou de acurácia média de 96,6% e máxima de 100,0% para classificar humanos e não humanos, mesmo não fazendo parte da parte superior da Tabela 5.6.

5.2.6 Melhores configurações

Tabela 5.7 – Melhores configurações em acurácia.

	Bins	Weight	Wavelet	Class.	Acur.	FP
(Exp. I)	16	LBCA	Coiflet 6	RF	94%	6,3%
	8	LBCA	Daubechies 4	RF	93%	7,4%
	16	ARRU	Daubechies 16	PMC[9-9-1]	92%	8,6%
	16	ARRU	Daubechies 8	PMC[9-5-1]	92%	8,6%
(Exp. II)	32	LBCA	Daubechies 4	Random F.	100%	0%
	32	ARRU	Daubechies 32	PMC[8-8-1]	99%	1,2%
	8	ARRU	Coiflet 6	PMC[8-16-1]	99%	1,2%
	16	LBCA	Beylkin 18	PMC[7-7-1]	97%	2,5%

Ao considerar todos os resultados discutidos até então, configurações mais factíveis em termos de acurácia são apresentados na Tabela 5.7. Tal conjunto de configurações requer que cada uma das partes tenha bons resultados nas discussões anteriores e não somente um único *outlier* maior como bom resultado. Por isso, para a Tabela 5.7, embora todos esquemas de pesos apresentaram bons resultados, apenas os esquemas de peso ARRU e LBCA foram usados (ambos com desvio padrão e acurácia sensivelmente melhores que os demais); as famílias wavelets usadas são, em grande maioria, Daubechies e Coiflets; os classificadores presentes na Tabela 5.7 também estão presentes nas Tabelas 5.5 e 5.6, na parte superior. Portanto, as configurações propostas como melhores apresentam bons resultados em questão de estabilidade.

Em casos que ambos os experimentos sejam necessários, uma recomendação praticável seria: [discretização: 32, pesagem: LBCA, wavelet: Daubechies 4, classificador: Random Forests]. No experimento 1, esta configuração obteve 91,0% e 100,0% no experimento 2, uma média de 95,0% de acurácia. Cada uma das partes desta recomendação aparece na Tabela 5.7. A Figura 5.5 ilustra tal proposta.

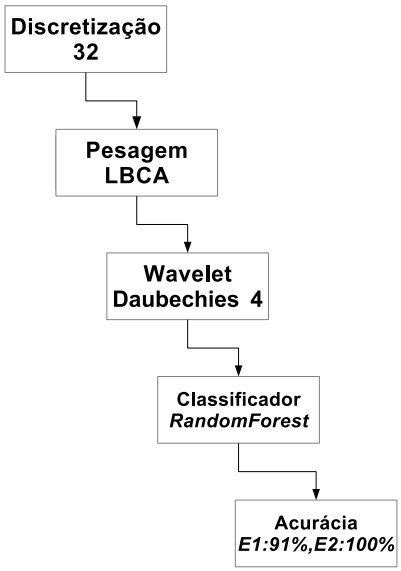


Figura 5.5 – Proposta de configuração para ambos os experimentos.

A última discussão é sobre a complexidade computacional. Para ilustrar o custo do modelo proposto, a seguir é listada a complexidade de cada etapa:

- Discretização: linear;
- Pesagem: lineares, pois, só dependem de contagem;
- TDW: linear, segundo Park [17];
- RF: $O(T*K*N*log(N))$ onde T é o número de árvores, K é o número de descritores, e N é o número de amostras [71];
- PMC: $O(W^3)$ para treinar, $O(W)$ para aplicar [72].

É importante destacar a importância de manter as três primeiras camadas do modelo proposto com complexidade linear. Discretização, Pesagem e TDW são dependentes do tamanho do vetor de descritores. Adicionalmente, medidas maiores de discretização aumentam o tamanho do sinal baseado em termos de forma exponencial. Embora o tamanho do sinal baseado em termos não altere a complexidade assintótica da pesagem e da TDW,

na prática, o cálculo da pesagem e da TDW também aumenta com sinais baseados em termos maiores. Em relação aos classificadores, a preocupação com a complexidade é menor, uma vez que a seleção de descritores mantêm o número de descritores por amostras com tamanho factível. Outro detalhe positivo em relação aos classificadores é que uma vez treinada, a aplicação de um classificador não é custosa.

6 CONCLUSÃO

Como apresentado no Capítulo 3, trabalhos foram desenvolvidos para solucionar algumas fraudes em RSD. Entretanto, este trabalho é a primeira solução proposta baseada somente em MT de forma a ser aplicável a qualquer RSD para detecção de fraudes.

Como proposta de uma solução puramente baseada em MT para um problema recorrente nas RSD atuais, foi desenvolvido um modelo. Este modelo foi validado com uma diversidade de configurações em dois experimentos, como visto no Capítulo 5. O Experimento 1 é focado em um caso generalizado de três classes: Humanos, *Cyborgs* e, *Bots*. Caso seja necessária a identificação de uma amostra em uma classe em específico, por exemplo *Cyborgs*, o Experimento 1 tem resultados que validam as configurações ideais. Porém, em casos que somente a produção humana de texto, ou seja, sem a presença de *bots* ou mesmo *cyborgs* seja requisitada, o Experimento 2 é o indicado.

Para alcançar melhores resultados em termos de acurácia média e desvio padrão, ambos os experimentos tiveram os resultados separados em camadas de configurações para análise, como mostrado no Capítulo 5.2.

A grande preocupação juntamente com a acurácia do modelo proposto foi seu custo computacional. Por isso, o fato de valores de discretização iguais a 8, 16 e 32 partes alcançarem bons resultados de acurácia média manteve a perspectiva de melhor desempenho, principalmente, ao considerar que quanto maior o valor de discretização, maior o vetor numérico que seria processado.

Juntamente com o modelo proposto, este trabalho apresentou um novo esquema de pesagem - um recurso essencial em diversos trabalhos de MT - que foi comparado à outros esquemas de pesagem da literatura no Capítulo 5.2. Tal esquema de pesagem, LBCA , contribuiu significantemente para a distinção de contas em todos os experimentos, obtendo o melhor resultado quando comparado aos da literatura atual.

Por meio do uso de diferentes famílias wavelets, um estudo adicional em relação à atual literatura foi feito considerando TDW. Coeficientes obtidos pela família Daubechies, como previamente destacado na literatura, obtiveram bons resultados, enquanto as demais famílias como Vaidyanathan e Beylkin que são famílias criadas especificamente para problemas de áudio não obtiveram resultados de destaque.

Ao considerar a seleção de descritores para a redução de dimensionalidade, dois descritores se mantiveram presentes na maioria das configurações: tamanho do *Lexicon* e o tamanho do *Corpus*. Tal fato enfatiza a importância do uso dessas características para o cáculo do LBCA. Justamente ao levar em consideração o *Lexicon* o LBCA foi capaz de obter bons resultados de acurácia média, e desta forma, foi consolidado que a quantidade

de palavras distintas é relevante para identificar se o usuário de uma conta é um ser humano ou não.

Em relação aos classificadores, foi observado que as menores arquiteturas obtiveram os melhores resultados. Este fato reforça a expectativa de manter adequado o desempenho do modelo proposto para as RSD. A PMC é conhecida na literatura pela possibilidade do aprendizado *on-line*, ou seja, uma vez treinada, é possível que novas amostras sejam generalizadas sem a necessidade de reajustar o modelo de indução para todas as amostras anteriores. A RF, por sua vez, apresenta estabilidade em termos de acurácia média. Isto se deve principalmente por ser um classificador que utiliza várias árvores, escolhendo um subconjunto mais adequado à amostra corrente. A criação da RF não é considerado custoso no contexto das RSD, pois é possível a realização da etapa de treinamento (criação das árvores) em ambientes distribuídos.

A acurácia do modelo proposto atingiu níveis satisfatórios. A precisão de 94% foi alcançada para a classificação de contas, distinguindo-as entre Humanos, *Cyborgs* e *Bots*. O resultado foi ilustrado na Tabela 5.5. Outro resultado importante foi que o Experimento 2 obteve acurácia de 100% como visto na Tabela 5.6, para a melhor configuração. Uma terceira constatação em relação ao modelo proposto é encontrado na Figura 5.5, onde uma mesma configuração seria praticável para ambos os experimentos, contribuindo para uma solução em um cenário mais generalista.

Ao considerar a preocupação a respeito do custo computacional, um modelo factível em desempenho, como apresenta a Figura 5.5. Esta configuração mantem a solução viável para o uso nas RSD, principalmente em relação a discretização, pesagem e TDW que dependem do tamanho do sinal baseado em termos. Ainda em relação a discretização, a medida usada pode aumentar o tamanho do sinal baseado em termos de forma exponencial.

Assim, por meio do modelo e dos resultados, considera-se válida a contribuição deste trabalho na redução de fraudes em RSD, auxiliando na diminuição do número de vítimas de fraudes ao detectar contas que não são pertencentes a um humano.

Em relação a trabalhos futuros, a principal preocupação deste trabalho foi as RSD e seus propósitos com a sociedade em termos de segurança. Desta forma, *bots* que são fraudulentos precisam ser encontrados para que a ordem nas RSD seja preservada. Por isso, seria de grande interesse analisar o comportamento textual especificamente de *bots* para detectar se o mesmo é fraudulento ou legítimo (não fraudulento).

REFERÊNCIAS

[1] ZAPPAVIGNA, M. Ambient affiliation: A linguistic perspective on twitter. *New Media & Society*, SAGE Publications, v. 13, n. 5, p. 788–806, 2011.

[2] HSIEH, L.-C. et al. Live semantic sport highlight detection based on analyzing tweets of twitter. In: IEEE. *Multimedia and Expo (ICME), 2012 IEEE International Conference on*. [S.l.], 2012. p. 949–954.

[3] HASSAN, A.; ABBASI, A.; ZENG, D. Twitter sentiment analysis: A bootstrap ensemble framework. In: IEEE. *Social Computing (SocialCom), 2013 International Conference on*. [S.l.], 2013. p. 357–364.

[4] BAHRAINIAN, S.-A.; DENGEL, A. Sentiment analysis and summarization of twitter data. In: IEEE. *Computational Science and Engineering (CSE), 2013 IEEE 16th International Conference on*. [S.l.], 2013. p. 227–234.

[5] MOSTAFA, M. M. More than words: Social networks' text mining for consumer brand sentiments. *Expert Systems with Applications*, Elsevier, v. 40, n. 10, p. 4241–4251, 2013.

[6] YU, S. J. The dynamic competitive recommendation algorithm in social network services. *Information Sciences*, Elsevier, v. 187, p. 1–14, 2012.

[7] SERRANO-GUERRERO, J. et al. Sentiment analysis: A review and comparative analysis of web services. *Information Sciences*, Elsevier, v. 311, p. 18–38, 2015.

[8] RILL, S. et al. Politwi: Early detection of emerging political topics on twitter and the impact on concept-level sentiment analysis. *Knowledge-Based Systems*, Elsevier, v. 69, p. 24–33, 2014.

[9] BHAT, S. Y.; ABULAISH, M. Community-based features for identifying spammers in online social networks. In: ACM. *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. [S.l.], 2013. p. 100–107.

[10] FONG, S.; ZHUANG, Y.; HE, J. Not every friend on a social network can be trusted: Classifying imposters using decision trees. In: *Future Generation Communication Technology (FGCT), 2012 International Conference on*. [S.l.: s.n.], 2012. p. 58–63.

[11] CHU, Z. et al. Detecting automation of twitter accounts: Are you a human, bot, or cyborg? *Dependable and Secure Computing, IEEE Transactions on*, IEEE, v. 9, n. 6, p. 811–824, 2012.

[12] WALD, R. et al. Predicting susceptibility to social bots on twitter. In: IEEE. *Information Reuse and Integration (IRI), 2013 IEEE 14th International Conference on*. [S.l.], 2013. p. 6–13.

[13] MILLER, Z. et al. Twitter spammer detection using data stream clustering. *Information Sciences*, Elsevier, v. 260, p. 64–73, 2014.

[14] BOSHMAF, Y. et al. The socialbot network: when bots socialize for fame and money. In: ACM. *Proceedings of the 27th Annual Computer Security Applications Conference*. [S.l.], 2011. p. 93–102.

[15] GRIER, C. et al. spam: the underground on 140 characters or less. In: ACM. *Proceedings of the 17th ACM conference on Computer and communications security*. [S.l.], 2010. p. 27–37.

[16] CHU, Z. et al. Who is tweeting on twitter: human, bot, or cyborg? In: ACM. *Proceedings of the 26th annual computer security applications conference*. [S.l.], 2010. p. 21–30.

[17] PARK, L. A. *Spectral Based Information Retrieval*. Tese (Doutorado) — The University of Melbourne, 2003.

[18] PARK, L. A.; RAMAMOHANARAO, K.; PALANISWAMI, M. A novel document retrieval method using the discrete wavelet transform. *ACM Transactions on Information Systems (TOIS)*, ACM, v. 23, n. 3, p. 267–298, 2005.

[19] ZHANG, S.; ZHANG, C.; WU, X. *Knowledge discovery in multiple databases*. [S.l.]: Springer Science & Business Media, 2004.

[20] MAIMON, O.; ROKACH, L. *Data mining and knowledge discovery handbook*. [S.l.]: Springer, 2005. v. 2.

[21] FELDMAN, R.; SANGER, J. *The text mining handbook: advanced approaches in analyzing unstructured data*. [S.l.]: Cambridge University Press, 2007.

[22] WITTEN, I. H.; FRANK, E. *Data Mining: Practical machine learning tools and techniques*. [S.l.]: Morgan Kaufmann, 2005.

[23] FUGAL, D. L. *Conceptual wavelets in digital signal processing: an in-depth, practical approach for the non-mathematician*. [S.l.]: Space & Signals Technical Pub., 2009.

[24] WALKER, J. S. *A primer on wavelets and their scientific applications*. [S.l.]: CRC press, 2008.

[25] DOUGHERTY, G. *Pattern recognition and classification: an introduction*. [S.l.]: Springer Science & Business Media, 2012.

[26] LIU, J.; SUN, J.; WANG, S. Pattern recognition: An overview. *IJCSNS International Journal of Computer Science and Network Security*, v. 6, n. 6, p. 57–61, 2006.

[27] AGGARWAL, C. C. *Data classification: algorithms and applications*. [S.l.]: CRC Press, 2014.

[28] LIVINGSTON, F. Implementation of breiman's random forest machine learning algorithm. *ECE591Q Machine Learning Journal Paper*, 2005.

[29] LIAW, A.; WIENER, M. Classification and regression by randomforest. *R news*, v. 2, n. 3, p. 18–22, 2002.

[30] LEE, K.; CAVERLEE, J.; WEBB, S. Uncovering social spammers: social honeypots+ machine learning. In: ACM. *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*. [S.l.], 2010. p. 435–442.

[31] JIANG, M. et al. Detecting suspicious following behavior in multimillion-node social networks. In: INTERNATIONAL WORLD WIDE WEB CONFERENCES STEERING COMMITTEE. *Proceedings of the companion publication of the 23rd international conference on World wide web companion*. [S.l.], 2014. p. 305–306.

[32] SONG, J.; LEE, S.; KIM, J. Spam filtering in twitter using sender-receiver relationship. In: SPRINGER. *Recent Advances in Intrusion Detection*. [S.l.], 2011. p. 301–317.

[33] LUMEZANU, C.; FEAMSTER, N. Observing common spam in tweets and email. In: CITESEER. *Proc. IMC*. [S.l.], 2012.

[34] MARTINEZ-ROMO, J.; ARAUJO, L. Detecting malicious tweets in trending topics using a statistical analysis of language. *Expert Systems with Applications*, Elsevier, v. 40, n. 8, p. 2992–3000, 2013.

[35] BENEVENUTO, F. et al. Detecting spammers on twitter. In: *Collaboration, electronic messaging, anti-abuse and spam conference (CEAS)*. [S.l.: s.n.], 2010. v. 6, p. 12.

[36] SANTOS, I. et al. Twitter content-based spam filtering. In: SPRINGER. *International Joint Conference SOCO'13-CISIS'13-ICEUTE'13*. [S.l.], 2014. p. 449–458.

[37] CAO, Q. et al. Aiding the detection of fake accounts in large scale social online services. In: USENIX ASSOCIATION. *Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation*. [S.l.], 2012. p. 15–15.

[38] STEIN, T.; CHEN, E.; MANGLA, K. Facebook immune system. In: ACM. *Proceedings of the 4th Workshop on Social Network Systems*. [S.l.], 2011. p. 8.

[39] CRESCI, S. et al. A fake follower story: improving fake accounts detection on twitter. *IIT-CNR, Tech. Rep. TR-03*, 2014.

[40] MESSIAS, J. et al. You followed my bot! transforming robots into influential users in twitter. *First Monday*, v. 18, n. 7, 2013.

[41] EDWARDS, C. et al. Is that a bot running the social media feed? testing the differences in perceptions of communication quality for a human agent and a bot agent on twitter. *Computers in Human Behavior*, Elsevier, v. 33, p. 372–376, 2014.

[42] MENG, T. et al. Wavelet analysis in current cancer genome research: A survey. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, IEEE Computer Society Press, v. 10, n. 6, p. 1442–14359, 2013.

[43] LIN, T.-C.; LIN, C.-M. Wavelet-based copyright-protection scheme for digital images based on local features. *Information Sciences*, Elsevier, v. 179, n. 19, p. 3349–3358, 2009.

[44] TEDMORI, S.; AL-NAJDAWI, N. Image cryptographic algorithm based on the haar wavelet transform. *Information Sciences*, Elsevier, v. 269, p. 21–34, 2014.

[45] WALKER, J. S. *A primer on wavelets and their scientific applications*. [S.l.]: CRC press, 1999.

[46] HAN, X.; CHANG, X. An intelligent noise reduction method for chaotic signals based on genetic algorithms and lifting wavelet transforms. *Information Sciences*, Elsevier, v. 218, p. 103–118, 2013.

[47] YANNAN, W.; SHUDONG, Z.; HUI, L. Study of image compression based on wavelet transform. In: *Intelligent Systems Design and Engineering Applications, 2013 Fourth International Conference on*. [S.l.: s.n.], 2013. p. 575–578.

[48] PARK, L. A.; PALANISWAMI, M.; RAMAMOHANARAO, K. A novel web text mining method using the discrete cosine transform. In: *Principles of Data Mining and Knowledge Discovery*. [S.l.]: Springer, 2002. p. 385–397.

[49] PARK, L. A.; RAMAMOHANARAO, K.; PALANISWAMI, M. Fourier domain scoring: A novel document ranking method. *Knowledge and Data Engineering, IEEE Transactions on*, IEEE, v. 16, n. 5, p. 529–539, 2004.

[50] PURWITASARI, D.; OKAZAKI, Y.; WATANABE, K. A study on web resources_ navigation for e-learning: Usage of fourier domain scoring on web pages ranking method. In: IEEE. *Innovative Computing, Information and Control, 2007. ICICIC'07. Second International Conference on*. [S.l.], 2007. p. 458–458.

[51] PURWITASARI, D. A study on ranking method in retrieving web pages based on content and link analysis: Combination of fourier domain scoring and pagerank scoring. *Jurnal Ilmiah Teknologi Informasi*, v. 7, n. 1, p. 9–18, 2008.

[52] THAICHAROEN, S.; ALTMAN, T.; CIOS, K. J. Structure-based document model with discrete wavelet transforms and its application to document classification. In: AUSTRALIAN COMPUTER SOCIETY, INC. *Proceedings of the 7th Australasian Data Mining Conference-Volume 87*. [S.l.], 2008. p. 209–217.

[53] XEXÉO, G. et al. Using wavelets to classify documents. In: IEEE. *Web Intelligence and Intelligent Agent Technology, 2008. WI-IAT'08. IEEE/WIC/ACM International Conference on*. [S.l.], 2008. v. 1, p. 272–278.

[54] WENG, J.; LEE, B.-S. Event detection in twitter. *ICWSM*, v. 11, p. 401–408, 2011.

[55] POTHA, N.; STAMATATOS, E. A profile-based method for authorship verification. In: *Artificial Intelligence: Methods and Applications*. [S.l.]: Springer, 2014. p. 313–326.

[56] PARK, L. A.; PALANISWAMI, M.; RAMAMOHANARAO, K. A new implementation technique for fast spectral based document retrieval systems. In: IEEE. *Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on*. [S.l.], 2002. p. 346–353.

[57] ARRU, G. et al. Signal-based user recommendation on twitter. In: INTERNATIONAL WORLD WIDE WEB CONFERENCES STEERING COMMITTEE. *Proceedings of the 22nd international conference on World Wide Web companion*. [S.l.], 2013. p. 941–944.

[58] IGAWA, R. et al. Adaptive distribution of vocabulary frequencies: A novel estimation suitable for social media corpus. In: *Intelligent Systems (BRACIS), 2014 Brazilian Conference on*. [S.l.: s.n.], 2014. p. 282–287.

[59] HALL, M. A. *Correlation-based feature selection for machine learning*. Tese (Doutorado) — The University of Waikato, 1999.

[60] SINGH, K. et al. Big data analytics framework for peer-to-peer botnet detection using random forests. *Information Sciences*, Elsevier, v. 278, p. 488–497, 2014.

[61] RÍO, S. del et al. On the use of mapreduce for imbalanced big data using random forest. *Information Sciences*, Elsevier, 2014.

[62] MIRJALILI, S.; MIRJALILI, S. M.; LEWIS, A. Let a biogeography-based optimizer train your multi-layer perceptron. *Information Sciences*, Elsevier, v. 269, p. 188–209, 2014.

[63] CALCAGNO, G. et al. A multilayer perceptron neural network-based approach for the identification of responsiveness to interferon therapy in multiple sclerosis patients. *Information Sciences*, Elsevier, v. 180, n. 21, p. 4153–4163, 2010.

[64] BLISS, C. A. et al. Twitter reciprocal reply networks exhibit assortativity with respect to happiness. *Journal of Computational Science*, Elsevier, v. 3, n. 5, p. 388–397, 2012.

[65] WALD, R.; KHOSHGOFTAAR, T.; NAPOLITANO, A. Filter-and wrapper-based feature selection for predicting user interaction with twitter bots. In: IEEE. *Information Reuse and Integration (IRI), 2013 IEEE 14th International Conference on*. [S.l.], 2013. p. 416–423.

[66] WALD, R. et al. Which users reply to and interact with twitter social bots? In: IEEE. *Tools with Artificial Intelligence (ICTAI), 2013 IEEE 25th International Conference on*. [S.l.], 2013. p. 135–144.

[67] HWANG, T.; PEARCE, I.; NANIS, M. Socialbots: Voices from the fronts. *interactions*, ACM, v. 19, n. 2, p. 38–45, 2012.

[68] CINGIZ, M. Ö.; DIRI, B.; BIRICIK, G. Am i typing fresh tweets: Detecting up-to-dateness and worth of categorical information in microblogs. *Expert Systems with Applications*, Elsevier, 2015.

[69] JUNIOR, S. B. et al. Improved dynamic time warping based on the discrete wavelet transform. In: IEEE. *Multimedia Workshops, 2007. ISMW'07. Ninth IEEE International Symposium on*. [S.l.], 2007. p. 256–263.

[70] BARBON, S. et al. Wavelet-based dynamic time warping. *Journal of Computational and Applied Mathematics*, Elsevier, v. 227, n. 2, p. 271–287, 2009.

[71] ALBORZI, S. Z. et al. Cudagrn: Parallel speedup of inferring large gene regulatory networks from expression data using random forest. In: *Pattern Recognition in Bioinformatics*. [S.l.]: Springer, 2014. p. 85–97.

[72] REED, R. D.; MARKS, R. J. *Neural smithing: supervised learning in feedforward artificial neural networks*. [S.l.]: Mit Press, 1998.

TRABALHOS PUBLICADOS PELO AUTOR

Trabalhos publicados pelo autor durante o programa.

1. Igawa, R.A.; Sakaji Kido, G.; Seixas, J.L.; Barbon, S., **Adaptive Distribution of Vocabulary Frequencies: A Novel Estimation Suitable for Social Media Corpus**, Intelligent Systems (BRACIS), 2014 Brazilian Conference on, 2014, IEEE

2. Igawa, R.A.; Almeida A.M.G.; Zarpelão B.B.; Barbon, S., **Recognition of Compromised Accounts on Twitter**, Brazilian Symposium on Information Systems (SBSI), 2015, ACM, (Best Paper Candidate - Qualis CC 2012, B4)

3. Igawa, R.A.; Almenida A.M.G.; Zarpelão B.B.; Barbon, S., **Recognition on Online Social Network by user's writing style**, Revista Brasileira de Sistemas de Informação (iSys), 2015, SBC (Qualis CC 2014, B5)

4. Igawa, R.A.; Sakaji Kido G.; Barbon, S.; Guido, R.C., Proença, M.L., **Account Classification in Online Social Networks with LBCA and Wavelets**, Information Sciences, 2016, Elsevier, (Qualis CC 2014, A1)

5. Almeida, A.M.G.; Barbon, S.; Igawa, R.A.; Paraiso, E.C.; Moriguchi, S.N., **Opinion Mining: A Comparison of Hybrid Approaches**, International Conferences on Advances in Multimedia (MMedia), 2016, IEEE (Qualis CC 2012, B4)

6. Campos, G.F.C.; Igawa, R.A.; Seixas, J.L.; Almeida, A.M.G.; Guido, R.C.; Barbon, S., **Supervised Approach for Indication of Contrast Enhancement in Application of Image Segmentation**, International Conferences on Advances in Multimedia (MMedia), 2016, IEEE (Qualis CC 2012, B4)