

INSTITUTO FEDERAL DO NORTE DE MINAS GERAIS
CAMPUS MONTES CLAROS
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

**AVALIAÇÃO DA VIABILIDADE DE UM SISTEMA
DE IDENTIFICAÇÃO DE CHAMADAS FALSAS
EM CHAMADAS DE EMERGÊNCIA**

ROBERT CRISTIANO ALMEIDA VIANA
ORIENTADORA: LUCIANA BALIEIRO COSME

Montes Claros

Julho de 2019

ROBERT CRISTIANO ALMEIDA VIANA

**AVALIAÇÃO DA VIABILIDADE DE UM SISTEMA
DE IDENTIFICAÇÃO DE CHAMADAS FALSAS
EM CHAMADAS DE EMERGÊNCIA**

Projeto de Monografia apresentado ao Curso de Graduação em Ciência da Computação do Instituto Federal do Norte de Minas Gerais – Campus Montes Claros, como requisito parcial para a obtenção do grau de Bacharel em Ciência da Computação.

ORIENTADORA: LUCIANA BALIEIRO COSME

Montes Claros

Julho de 2019

“If it wasn’t hard, everyone would do it. It’s the hard that makes it great”
(Tom Hanks)

Resumo

Palavras-chave: Inteligência Artificial, Classificação de texto, Redes Neurais, Processamento de Linguagem Natural.

Abstract

Keywords: Artificial intelligence, emergency calls, text classification, neural networks, natural language processing.

Sumário

Resumo	vii
Abstract	ix
Lista de Figuras	xiii
Lista de Tabelas	xv
1 Introdução	1
1.1 Motivação	3
1.2 Objetivos	4
1.3 Estrutura do trabalho	4
2 Conceitos Básicos	7
2.1 Classificação de dados	7
2.1.1 Redes neurais artificiais (RNA)	8
2.1.2 K-vizinhos mais próximos	13
2.2 Processamento de linguagem natural	14
2.2.1 Classificação de texto	15
3 Trabalhos Relacionados	17
3.1 Classificadores de texto	17
3.2 Chamadas emergenciais falsas: Problemas e Soluções	19
3.3 Privacidade dos dados e ética	21
4 Metodologia	23
4.1 Ferramentas de apoio	24
4.2 Base de dados	25
4.3 Cronograma de atividades	26
Referências Bibliográficas	27

Lista de Figuras

2.1	Processo de classificação de flores do gênero Iris	8
2.2	Diagrama de um neurônio artificial	9
2.3	Diagramas que representam redes <i>Feed-Forward Network</i> e <i>Recurrent Neural Networks</i>	10
2.4	Algoritmo <i>Backpropagation</i> para treinamento de redes <i>Feed-Forward Network</i>	12
2.5	Ilustração do algoritmo K-vizinhos mais próximos aplicado em um plano cartesiano	13
A.1	Diagrama do Grafo G	32

Lista de Tabelas

3.1	Lista de códigos traduzidos do MPDS.	21
4.1	Tabela de atributos que descrevem uma ocorrência da base de dados <i>Call</i> <i>Data</i>	25
4.2	Cronograma de atividades planejadas para a segunda etapa deste projeto. .	26
A.1	Função incidência ψ de G	31

Capítulo 1

Introdução

Com o objetivo de ampliar, regulamentar e aumentar a eficiência dos serviços de urgência no Brasil, o Ministério da Saúde publicou duas portarias em 2003, a GM/MS nº 1863 e a GM/MS nº 1864 que instituíram, respectivamente, a Política Nacional de Atenção às Urgências e o componente pré-hospitalar móvel, por meio da implantação do Serviço de Atendimento Móvel de Urgência (SAMU), disponível através do número telefônico 192, como também os serviços de urgência associados, em todo o território nacional (Brasil, 2003a,b). A implantação desses projetos resultou em impactos positivos para toda a população (Vieira & Mussi, 2008; Machado et al., 2011; Minayo & Deslandes, 2008).

Contudo, também surgiram novos problemas e desafios. Um dos principais problemas encontrados são as chamadas falsas ou trotes, no qual o indivíduo entra em contato com a unidade de atendimento e relata uma situação inexistente, que se não identificada pelo atendente, pode resultar em deslocamentos desnecessários de unidades e de recursos ao local do chamado. Outra situação comum são as chamadas com categorização não emergencial, em que a população procura por orientações, tais como números de outros serviços, locais e endereços.

O grande problema dessas ligações é que elas ocupam a linha emergencial e consomem recursos, gerando transtornos e custos para os serviços de emergência. Este é um fator tão preocupante que, pode ocorrer a possibilidade da escassez total desses recursos em chamadas não emergenciais e caso haja outra pessoa que realmente necessite de atendimento, conseqüentemente não poderá receber os auxílios necessários a tempo e ir a óbito, em um grave acidente de trânsito, por exemplo.

O grande problema dessas ligações é que elas ocupam a linha emergencial, consomem recursos, geram transtornos e custos para os serviços de emergência. Este é um fator preocupante que, pode acarretar a escassez desses recursos em chamadas não emergenciais. Caso haja outra pessoa que realmente necessite de atendimento, ela po-

derá não receber os auxílios necessários a tempo e ir a óbito, em um grave acidente de trânsito, por exemplo.

Além desses riscos, uma mobilização indevida de recursos, tais como pessoas, viaturas, ambulâncias, carros de combate a incêndio e, principalmente, de tempo, geram também prejuízos financeiros ao Estado. E esses recursos que poderiam ser investidos em outros setores, como educação, infraestrutura, segurança ou aperfeiçoamento dos próprios serviços de emergência, por exemplo, acabam tendo que ser remanejados. Gastos estes que poderiam ser parcialmente evitados, se os atendentes puderem identificar melhor as chamadas falsas.

O ato do trote aos serviços de emergência é um crime previsto no Código Penal brasileiro. Segundo o Art. nº 340 do Código Penal (Brasil, 1940), "Provocar a ação de autoridade, comunicando-lhe a ocorrência de crime ou de contravenção que sabe não se ter verificado", quando identificado o autor, o mesmo pode ser detido por um período de um a seis meses ou multado. Com uma análise cautelosa ao artigo, percebe-se que o mesmo não abrange a comunicação falsa de situações de emergência que motivem o acionamento do Serviço de Atendimento Móvel de Urgência (SAMU) ou corpo de bombeiros, e sim somente nos casos em que há relato de infrações penais. Como relatado por Peixoto et al. (2015), existem outros artigos e projetos de leis que poderiam ajudar a punir tal ato, mas devido a sua natureza, talvez não sejam a melhor opção para o problema.

Embora que, grande parte dos autores de chamados falsos são crianças e adolescentes, o problema se encontra em chamadas falsas iniciadas por adultos, pois o conteúdo dessas mensagens tendem a serem mais próximas de uma ocorrência real. Dificultando assim, a identificação de ser uma chamada falsa pelo atendente.

Após consultas na literatura disponível, não foram encontrados muitos resultados que visam identificar tais chamadas emergenciais falsas. O que leva a crer que é uma área pouco visada e que existam outros fatores que influenciem o baixo investimento em pesquisas e soluções. Um desses fatores é que o conteúdo dessas ligações são sensíveis a privacidade e sujeitos a análise por um conselho de ética (Francisconi & Goldim, 1998). Sendo essa, uma barreira e um processo burocrático, porém necessário para o desencadeamento de pesquisas. Além disso, os gestores (responsáveis pelos dados) podem se sentir menos dispostos a disponibilizar dados anônimos para pesquisas devido a falta de divulgação de resultados positivos.

Quando se trabalha com metodologias para a indústria, como por exemplo, na identificação e classificação de falhas em equipamentos, há um grande número de dados abertos de funcionamento real (PHMsociety, 2009; NASA, 2007; Dua & Graff, 2017), que favorece a proposição de soluções para as mais variadas demandas da área.

Entretanto, existem poucas bases de dados públicas disponíveis para realizações de experimentos e prova de metodologias, que seriam capazes de solucionar o problema de identificação de chamadas falsas. E ainda dentre tais bases, a maioria está disponível no idioma inglês, e os dados que são disponibilizados são rasos, carecendo de detalhes e proporcionando pouco espaço para uma exploração aprofundada.

Portanto, a presente proposta não busca realizar apenas um experimento de classificação. O trabalho visa realizar um estudo de avaliação aprofundado e crítico sobre quais procedimentos técnicos devem ser realizados, qual o menor conjunto de informações a serem coletadas das chamadas, e quais questões de privacidade devem ser levantadas para que seja viável a construção de sistemas de suporte a decisão do gestor e do atendente nos serviços emergenciais.

1.1 Motivação

Segundo um estudo apontado pela Consultoria Legislativa do Senado, estima-se que o custo gerado por trotes aos serviços de emergência no Brasil, tais como o Serviço de Atendimento Móvel de Urgência (SAMU), Corpo de Bombeiros (CB) e Polícia Militar (PM), chega a R\$ 1 bilhão por ano. Esse levantamento foi apurado pela PM do estado do Amapá, no qual eles avaliaram que a cada deslocamento para um atendimento emergencial incompleto, gera um custo aproximado de R\$ 500,00 (G1, 2014; Peixoto et al., 2015).

Devido à proporção do problema, os atendentes do Centro Integrado de Operações de Segurança Pública (CIOSP) do governo de Mato Grosso, responsáveis estes pelo atendimento e despacho das chamadas emergenciais, receberam uma capacitação para identificar suspeitas de trote. Mesmo que não seja possível bloquear todas ligações indevidas, a triagem ajuda a amenizar os prejuízos (Araújo, 2016).

Ainda assim, nas pesquisas realizadas preliminarmente, não foram localizadas, no meio acadêmico, propostas de sistemas computacionais, que auxiliassem a resolução do problema. Diante desse cenário, que possui poucas pesquisas, visibilidade, resultados e também o não uso de técnicas computacionais para auxiliar os serviços de emergência, surge então, a possibilidade do levantamento de requisitos que seriam necessários para a construção de sistemas computacionais que auxiliem na identificação correta de chamadas falsas nos serviços de emergência. Busca-se ainda, motivar as autoridades envolvidas no processo, para que, invistam no desenvolvimento dessas ferramentas.

1.2 Objetivos

Deseja-se mostrar a viabilidade de produzir sistemas capazes de identificação de chamadas falsas, como também produzir documentos e argumentos para que os gestores desses serviços possam avaliar a possibilidade de liberação do acesso aos dados reais e completos das chamadas, sem prejuízo ou violação de privacidade e com o retorno de uma ferramenta de apoio a decisão aos operadores. Com a conclusão da avaliação, também será possível elencar quais dados podem ser suficientes ou necessários para a identificação.

Como objetivos específicos alcançam-se:

- Pesquisar o uso de técnicas computacionais que possam auxiliar os serviços de atendimento de emergência;
- Apontar como métodos de classificação de dados, junto ao processamento de linguagem natural possa viabilizar a construção de sistemas que auxiliem a identificação de chamadas falsas;
- Motivar, com resultados positivos, as autoridades responsáveis, para aumentar o investimento no desenvolvimento de ferramentas;

1.3 Estrutura do trabalho

O restante deste trabalho está organizado na seguinte estrutura:

- No capítulo 2 são apresentados conceitos básicos relacionados a classificação de dados, processamento de linguagem natural e classificação de texto. Além disso, dois algoritmos de classificação de texto são descritos, as redes neurais artificiais e o k-vizinhos mais próximos.
- No capítulo 3 é apresentada uma revisão na literatura, abordando os problemas gerados por chamadas falsas, como também tentativas e soluções propostas para reduzir o impacto consequente. Em seguida apresenta-se trabalhos da literatura sobre a classificação de texto, com o objetivo de compreender suas vantagens, capacidades e eficiência, quando aplicados em contextos diferentes ou semelhantes ao deste trabalho. Por fim, um tópico abordando sobre privacidade e ética que estão sujeitos os dados dessas chamadas emergenciais, bem como, as possíveis vantagens que seriam obtidas caso esses dados fossem de mais fácil acesso.

-
- No capítulo 4 é apresentado a metodologia proposta para alcançar os objetivos especificados, como também uma descrição da base de dado que será utilizada e o cronograma de atividades esperadas para a próxima etapa.

Capítulo 2

Conceitos Básicos

Nesta seção, serão explorados alguns fundamentos básicos para compreender, passo a passo, como será conduzida a metodologia proposta. Primeiramente, introduz-se o problema da classificação de dados. Logo após esse conceito, descreve-se como exemplos de técnicas de categorização de dados, as Redes Neurais Artificiais (RNA) e o K-vizinhos mais próximos. Por fim, é apresentada uma introdução ao Processamento de Linguagem Natural (PLN), dando ênfase a um sub-conceito muito valioso para os objetivos deste trabalho, que é a classificação de texto.

2.1 Classificação de dados

Classificação de dados é um problema que abrange inúmeras aplicações em diversos tipos de cenários no nosso dia a dia, tais como diagnóstico de doenças, identificação de objetos em fotos e vídeos, categorização de seres vivos e espécies, dentre outros. Esse problema é um dos tópicos mais ativos na área de aprendizado de máquina. Classificar dados consiste em determinar um rótulo ou classe para um objeto, baseado em um conjunto de características extraídas do mesmo (Duda et al., 1973; Bishop, 2006).

Em geral, cada dado é classificado como pertencente a uma única classe ou categoria. Essa forma de classificação é denominada classificação de rótulo único. Por outro lado, se houver mais uma forma de rotular a mesma entrada, então dá-se o nome de classificação de multi-rótulo.

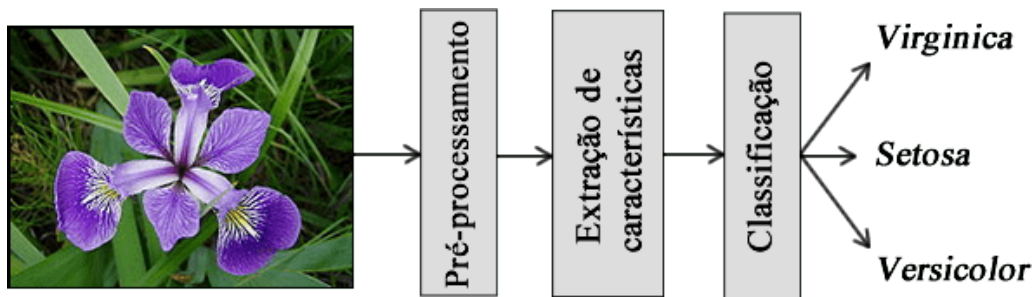
Formalmente, o processo de classificação consiste em: um conjunto de i entradas $X = \{X_1, \dots, X_i\}$, um conjunto de n classes $C = \{c_1, \dots, c_n\}$, tal que $n \geq 2$, e um conjunto de treinamento $Y = \{(X_1, \{c_1, \dots, c_j\}), \dots, (X_i, \{c_n, \dots, c_k\})\}$, no qual cada entrada X_i é categorizada por uma ou mais classes c_i . O objetivo geral de um classificador é aprender, através de seu conjunto de treinamento Y , uma possível correlação entre os atributos das entradas com suas classes, de tal forma que para uma entrada

$X' = \{X'_1, \dots, X'_i\}$ que não possua rótulo c qualquer, seja possível classificar-lá.

Para ilustrar o processo de classificação de dados, considere o problema da flor de Iris. Nesse problema, existe um conjunto de flores do gênero Iris que podem ser rotuladas de uma das três maneiras: do tipo setosa, virgínica ou versicolor. Partindo desse ponto, o objetivo é determinar a qual grupo uma determinada flor pertence baseado nas medidas de sépalas e pétalas da mesma. A Figura 2.1 ilustra o processo de classificação. Inicialmente as informações específicas sobre as sépalas e pétalas devem ser extraídas em um pré-processamento. Em seguida tais medidas são processadas e suas características extraídas. Por fim, é realizada a classificação das flores. Neste exemplo os valores de X serão as medidas de comprimento, largura das sépalas e pétalas e C assumirá os rótulos setosa, virgínica e versicolor.

Em geral, existem diversos algoritmos para classificação de dados, em que cada um possui sua especificidade, vantagens e desvantagens. Neste trabalho, aborda-se o uso de duas técnicas clássicas para classificação de dados: Redes Neurais Artificiais (RNA) e K-vizinhos mais próximos (KNN).

Figura 2.1. Processo de classificação de flores do gênero Iris



Fonte: Pacheco (2016)

2.1.1 Redes neurais artificiais (RNA)

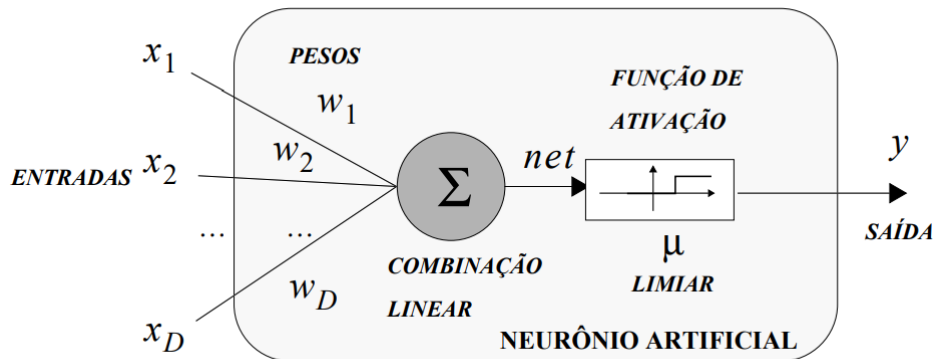
O ser humano possui capacidades cognitivas extraordinárias e, desde o surgimento da computação, desejou-se projetar máquinas capazes de realizar tarefas inteligentes que, até então, somente eram executadas por humanos. Os primeiros trabalhos desenvolvidos nessa área foram: um neurônio apresentado por McCulloch & Pitts (1943), usado posteriormente como base para a concepção do *Perceptron* por Rosenblatt (1958) e um neurônio chamado *Adaline* por Widrow & Hoff (1960). Tais trabalhos deram origem ao conceito da RNA que, em outras palavras, é uma tentativa de copiar a estrutura e o funcionamento do cérebro, composto este por bilhões de neurônios, para uma estrutura artificial, transformando assim as redes neurais biológicas em redes neurais artificiais.

(Rauber, 2005). Uma RNA é normalmente implementada através de um programa de computador (*software*) ou através de componentes eletrônicos (*hardware*).

Para compreender o conceito por trás de uma rede neural, é preciso introduzir um modelo simplificado de um neurônio e suas capacidades de processamento associadas. Cada neurônio é considerado como uma unidade básica de processamento que, quando estimulada por sinais de entrada, emite sinais de saída como uma reação. Tais sinais, emitidos por um neurônio, são repassados para outros neurônios através de uma conexão sináptica. Tal processo pode ser repetido por várias camadas de neurônios até chegar ao nosso cérebro, que então processa essa informação e produz novas reações (Baeza-Yates et al., 1999). A principal função de uma rede neural é armazenar conhecimento experimental e torná-lo disponível, o que em prática significa que este conhecimento é adquirido e armazenado em pesos sinápticos durante o processo.

Antes de definir e explora-se mais sobre as redes neurais, uma breve introdução a grafos é sugerida ao leitor (Apêndice A).

Figura 2.2. Diagrama de um neurônio artificial



Fonte: Rauber (2005)

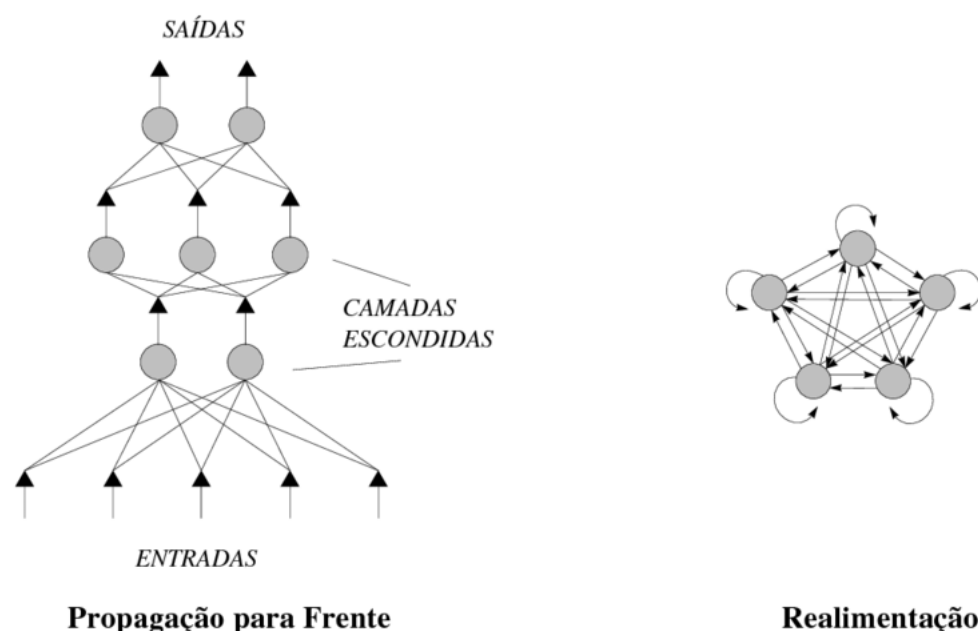
Uma rede neural pode ser representada matematicamente através de uma estrutura de grafo, em que os vértices fazem o papel dos neurônios e as arestas representam as conexões sinápticas entre os neurônios. Se adicionado pesos a tais arestas, é possível mensurar a força de tal conexão sináptica. Seja x_i entradas fornecidas por outros neurônios para um neurônio artificial. O processamento desse neurônio consiste em uma combinação linear das D entradas, tais que $\sum_{i=1}^D w_i x_i$, onde x_i é uma aresta com peso w_i . A computação desse valor, resulta em net (como ilustrado na Figura 2.2). Se o valor de net ultrapassar um limiar μ pré-definido, uma função de ativação será executada. Neste exemplo, a função de ativação escolhida foi a Heaviside ou degrau unitário, como é comumente chamada na matemática. Por ser uma função binária, dispara $y = 1$ ou $y = 0$ na saída, de acordo com μ . Além dessa função binária, existem

outras alternativas, nos quais devem ser avaliadas de acordo com suas características, tais como o uso da função linear, função tangente hiperbólica, função arco tangente, função sigmóide, dentre outras.

Geralmente o uso de somente um único neurônio não é suficiente para a efetuação de tarefas de classificação mais complexas, necessitando assim, do uso conjunto de outros neurônios, nos quais operem em paralelo a este, aumentando assim a capacidade de processamento da rede neural. A partir disso, surge o conceito de organização dos neurônios em camadas (Duda et al., 1973; Bishop, 2006; Martin & Jurafsky, 2018). Camadas estas que, quando não ligadas diretamente às entradas e nem às saídas da rede, chamam-se camadas escondidas ou ocultas (do inglês: *Hidden layers*).

Uma categorização fundamental da topologia dos neurônios pode ser feita em relação ao método escolhido para a propagação da informação recebida, ou em outras palavras, quem receberá a informação processada pelo último neurônio. Pode-se distinguir então, de duas formas, entre redes de propagação para frente (do inglês: *Feed-Forward Network* - FFN) e redes realimentadas (do inglês: *Recurrent Neural Networks* - RNN), ver Figura 2.3.

Figura 2.3. Diagramas que representam redes *Feed-Forward Network* e *Recurrent Neural Networks*



Fonte: Rauber (2005)

Uma *Feed-Forward Network* é uma rede multicamadas unidirecional, no qual não

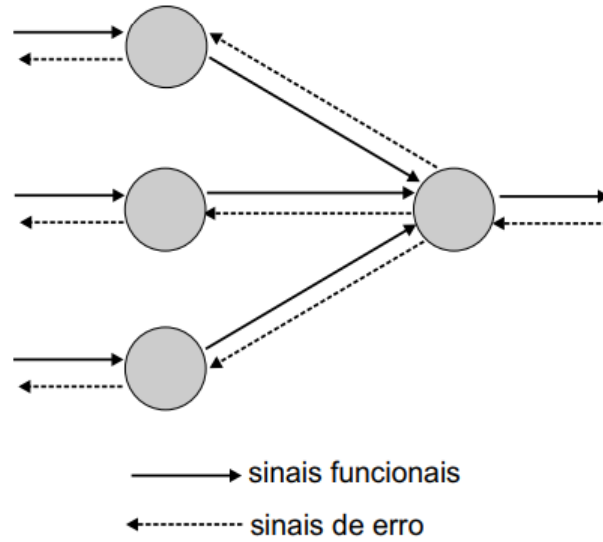
existe um ciclo entre as camadas de neurônios, ou seja, após o processamento de uma camada, as informações são sempre repassadas adiantes, para as sucessoras camadas, até a camada de saída, nunca sendo possível o retorno para camadas já executadas. Redes de propagação para frente, também chamadas de *Multilayer Perceptron - MLP*, é a topologia de neurônios mais utilizada e estudada na área de aprendizado de máquina. As redes FFN são comumente aplicadas para o reconhecimento de padrões e classificação de dados.

Uma *Recurrent Neural Networks* é uma rede que possui arestas entre os seus neurônios sem restrições, em que o comportamento dinâmico desempenha um papel fundamental nesse modelo. Em alguns casos, os valores de ativação da rede passam por um processo de relaxação, por múltiplos e até repetidos neurônios, até chegarem a um estado estável. Difere, principalmente, das redes FNN por possuir uma conexão com suas decisões passadas, em que cada saída pode ser tratada com uma nova entrada, armazenando conhecimento. É uma topologia poderosa e ao mesmo tempo complexa, tradicionalmente difícil de ser treinada. As redes RNN vêm sendo aplicadas com grande sucesso para o processamento de linguagem natural, com ênfase em textos e falas.

Uma propriedade relevante das redes neurais é sua habilidade de aprender a partir do ambiente a qual foi inserida, também chamado de ambiente de aprendizado, em que sua capacidade de aprender é sucessivamente melhorada através do processo de adaptação dos parâmetros livres (pesos sinápticos e limiares) de sua rede. Este aprendizado pode ser adquirido de várias formas (Bishop, 2006; Duda et al., 1973; Rauber, 2005). Duas formas de aprendizado comuns são: aprendizagem supervisionada, em que o conhecimento é transmitido por meio de exemplos de entrada e saída; e aprendizagem não-supervisionada, no qual a rede só dispõe dos valores de entrada e deve descobrir as correlações entre os exemplos de treino.

Um método amplamente utilizado para o treinamento e aquisição de aprendizagem para redes FNN é o algoritmo de retro-propagação do erro (do inglês: *Backpropagation*) proposto por Werbos (1974), baseado na regra de aprendizagem por correção de erro. Neste algoritmo (ilustrado na Figura 2.4), a aprendizagem se consiste em dois passos através das diferentes camadas da rede: um passo para frente (propagação), no qual os pesos sinápticos da rede estão fixos, repassando os sinais funcionais normalmente desde a entrada até a saída; e um passo para trás (retro-propagação), em que cada saída gerada por um neurônio é subtraída da resposta real da rede, produzindo assim um sinal de erro, propagado em direção inversa as arestas da rede. O objetivo desse algoritmo é ajustar os pesos sinápticos da rede de forma que, a resposta real obtida, se aproxime ao máximo da desejada, ou seja, minimizar o erro gerado pela processamento da rede.

Figura 2.4. Algoritmo *Backpropagation* para treinamento de redes *Feed-Forward Network*



Já para treinar uma rede RNN, o algoritmo de treinamento é baseado no *Backpropagation*, chamado de *Backpropagation Through Time - BPTT*. Que assim como o seu primitivo, calcula o erro gerado por cada neurônio, adaptando os pesos sinápticos. Como diferencial, a variável tempo é adicionada a cada passo. O que em prática significa, que para cada vez que é executado a retro-propagação do erro, uma camada cópia é gerada e deverá ser levada em consideração no próximo passo. Este algoritmo tende a ser computacionalmente custoso para cada novo passo dado, devido ao fato da ordem do erro aumentar a cada execução. Podendo reduzir os valores dos pesos a zero, como também crescerem exponencialmente, resultando em uma aprendizagem lerda e sujeito a ruído. Uma solução alternativa para este problema é o algoritmo aproximado *Truncated BPTT*, no qual um limite de passos a serem considerados é definido, reduzindo assim o custo total de computação para sequências longas.

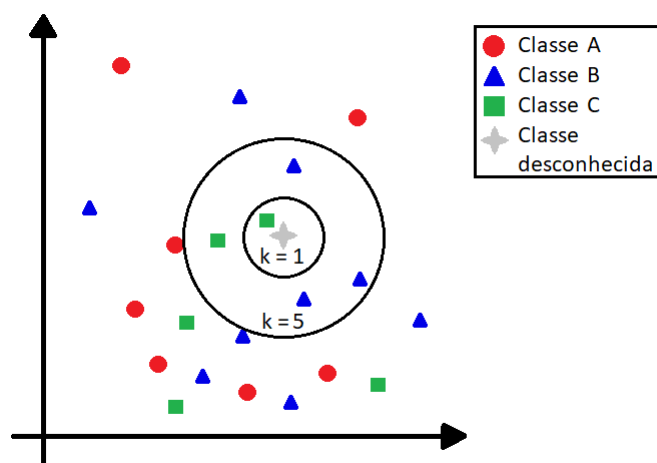
Por ser uma ferramenta poderosa, flexível e possuir uma grande capacidade de processamento, vem apresentando resultados excepcionais nas mais diversas aplicações da literatura (Gupta, 2018; Martin & Jurafsky, 2018; Bishop, 2006; Duda et al., 1973; Rauber, 2005), justificando assim a sua escolha como classificador de dados para este trabalho.

Por outro lado, devido a sua complexidade e alto custo computacional, pode ser um obstáculo a ser enfrentado. Supondo que existam n exemplos de teste, com m características, k *Hidden layers*, no qual cada uma contém h neurônios e existam p neurônios de saída. O tempo total de execução do algoritmo de *Backpropagation* para uma MLP seria $\mathcal{O}(n * m * h^k * p * i)$, no qual i é o número de iterações executadas.

2.1.2 K-vizinhos mais próximos

O algoritmo K-vizinhos mais próximos (do inglês: *K-nearest neighbours* - KNN) tem como objetivo determinar o rótulo de classificação de uma amostra, baseando-se em outras amostras vizinhas, advindas de um conjunto de treinamento. O classificador KNN, um dos mais simples e, ao mesmo tempo, um dos mais eficazes, dentre os algoritmos de classificação, é baseado em instâncias. Esse algoritmo encontra os k objetos mais similares ao termo de consulta e realiza uma votação de acordo com as classes às quais pertencem esses k objetos, assinalando por fim, uma classe ao objeto de teste. A literatura apresenta diversas formas para expressar essa distância/similaridade dentre os objetos de análise (Fukunaga & Narendra, 1975; Duda et al., 1973). Por exemplo, se os dados trabalhados estão em formato de texto, é comum utilizar a similaridade por cossenos. Por outro lado, se os dados possuírem formato numérico, possivelmente a distância euclidiana será mais eficaz.

Figura 2.5. Ilustração do algoritmo K-vizinhos mais próximos aplicado em um plano cartesiano



Fonte: Elaborado pelo autor

Na Figura 2.5, é ilustrado o processo de classificação com o algoritmo KNN. Neste exemplo, têm-se três classes, anteriormente conhecidas, sendo elas: classe A (círculo vermelho), classe B (triângulo azul) e classe C (quadrado verde). O objetivo é identificar, por similaridade, a qual classe pertence a amostra (estrela cinza), olhando para os seus k vizinhos mais próximos. Para $k = 1$, esse algoritmo classificaria a amostra como pertencente a classe C. Por outro lado, se o valor escolhido para k é 5, por votação majoritária, a amostra seria classificada como pertencente a classe B.

Um grande fator que pode definir a eficácia do algoritmo KNN no contexto em que é aplicado, é o valor de escolha para o k . Por ser um valor variável (não constante),

deve ser determinado de forma empírica, variando de acordo com a base de dados. Caso o valor escolhido para k seja muito baixo, a classificação ficará sujeita a *outliers*, ou seja, dados atípicos ou ruídos, que não condizem com o contexto da classificação. Já por outro lado, se k assumir um valor alto, a vizinhança poderá incluir elementos pertencentes a outras classes, não necessariamente relevantes ao objeto de análise, interferindo assim, no resultado final da rotulação (Fukunaga & Narendra, 1975). Por ser um fator relevante, deverá ser avaliado cuidadosamente.

2.2 Processamento de linguagem natural

O processamento de linguagem natural (PLN) têm como objetivo tratar os mais diversos aspectos presentes dentro da comunicação humana, tais como sons, palavras, sentenças e discursos, levando em consideração os seus formatos, referências, estruturas, significados, contextos e aplicações. Embora exista outros animais que possuem um vocabulário com centenas de sinais, tais como os elefantes e os golfinhos, somente os seres humanos possuem a capacidade de se comunicar, de forma confiável, em um número ilimitado de mensagens qualitativamente diferentes, sobre um tema qualquer (Russell & Norvig, 1994; Gonzalez & Lima, 2003).

Hoje em dia, com o constante crescimento da rede mundial de computadores, possibilitou o acesso a enumeras páginas de informações na *Web*, no qual quase todas elas estão em um formato de linguagem natural. Entretanto, disponibilidade não significa fácil acesso à informação. Para uma máquina adquirir tal conhecimento, ela precisa ser treinada, de forma exaustiva, para compreender as complexas, e muitas vezes ambíguas, linguagens em que os seres humanos se comunicam.

Segundo Russell & Norvig (1994), as linguagens naturais, tais como o português e o espanhol, não podem ser caracterizadas como um conjunto de sentenças definitivas, pois de acordo com o contexto em que for definida uma sentença de alguma dessas linguagens, ela pode possuir inúmeras interpretações diferentes. Portanto, convém definir um modelo de linguagem natural como uma distribuição de probabilidade sobre sentenças. Existe um famoso ditado popular brasileiro que diz "para um bom entendedor, meia palavra basta", o que pode ser comumente aplicado para nós humanos, que possuímos uma espécie de dispositivo de especialização para aquisição de linguagens (Chomsky, 2014). Já que meia palavra basta, pode-se concluir que uma sentença de uma linguagem natural não é sempre aleatória, e que sim possui algum grau de previsibilidade e correlação entre a escolha das palavras. Portanto, nos leva a acreditar que palavras similares estejam presentes no mesmo contexto.

O PLN consiste no emprego de um conjunto de técnicas computacionais para apren-

der, entender e reproduzir uma linguagem natural. No processo de tradução do significado, tratamento de ambiguidade e entre outros desafios, o PLN pode utilizar de conhecimentos linguísticos e métodos estatísticos para resolvê-los. Por exemplo, considere uma análise sobre dois textos semelhantes A e B, no qual desconfia-se que exista possibilidade de plágio. Com o uso de um pré-processamento, seria possível filtrar os textos para remover *Stopwords*, que são palavras funcionais, tais como artigos, preposições e conetivos, que quando analisadas individualmente não possuem grande relevância para o contexto. Após o pré-processamento, é possível aplicar um método da distância mínima de edição, que como o próprio nome diz, significa quantas operações de inserções, remoções ou substituições de caracteres são necessárias para que o texto A torne-se o texto B, ou vice-versa.

Uma das tarefas possíveis no PLN é a classificação de texto. Para compreender melhor o processo de aplicação de conhecimentos linguísticos para essa tarefa, é apresentado a seguir uma seção sobre classificação de texto.

2.2.1 Classificação de texto

Segundo Aggarwal (2014) um dos principais desafios encontrados durante o processo de classificação de texto é sobre o tamanho dos dados tratados, que podem variar de algumas poucas dezenas para milhões de palavras. Esses dados se encontram, quase sempre, de maneira esparsa, ou seja, possuindo baixa frequência de uso. Por outro lado, têm-se muitas vezes uma alta frequência de dados não úteis para tratamento, como as *Stopwords*.

É comum também, dependendo do contexto ou de como obteve-se o texto, que exista palavras com o mesmo significado, erros ortográficos ou até mesmo erros de codificação. Logo, muitas vezes é necessário uma etapa de normalização do texto. Normalizar um texto significa, segundo (Martin & Jurafsky, 2018), convertê-lo de forma conveniente à um formato padrão, que facilite as manipulações sobre os dados. Exemplos de sub-etapas da normalização são: *tokenization* e *lemmatization*.

Na *tokenization*, deseja-se separar as palavras contidas em um texto, de maneira que elas fiquem isoladas. Por exemplo, na frase "fui ao Rio de Janeiro", resultaria em 5 *tokens* distintos: "fui", "ao", "Rio", "de", "Janeiro". Contudo, uma separação em *tokens* por somente o uso do espaçamento em branco, nem sempre representa o contexto da frase, pois como no exemplo "fui ao Rio de Janeiro", Rio de Janeiro não deve ser tratado como três *tokens* distintos, e sim como um único *token*, que no caso representa a cidade ou estado brasileiro. Em algumas linguagens naturais, nem todas as palavras possuem espaço entre si, como é o caso do mandarim da China, por exemplo, dificultando ainda mais o processo de *tokenization*.

A etapa de *lemmatization*, é o processo, efetivamente, de deflexionar uma palavra para determinar o seu lema. Por exemplo, as palavras gato, gata, gatos, gatas são todas formas do mesmo lema: gato. Igualmente, as palavras estudou, estudava, estudaria, estudará, são do mesmo lema: estudar. A lematização é útil quando deseja-se ver os usos de palavras em contextos sem importância das flexões, como também para a criação e uso de índices ou na investigação linguística. Por poder se tornar um processo custoso, dependendo da quantidade de dados, uma alternativa mais barata acaba sendo mais viável nesses cenários. Surgindo então o conceito de uma análise morfológica, chamada *stemming*.

No *stemming*, deseja-se reduzir as palavras flexionadas, ou às vezes derivadas, ao seu tronco, base ou raiz, cortando assim caracteres dos seus sufixos. Por exemplo, a stemização das palavras estudou, estudava, estudaria, estudará, será "estud", visto que é o prefixo comum e redutível por todos. Um dos algoritmos mais conhecidos para *stemming* é o Porter, proposto por (Porter, 1980). Se por um lado, facilita a identificação de variantes para um mesmo lema e possui o potencial de lidar com palavras desconhecidas, transformando-as em uma semelhante conhecida, pelo outro lado, a utilização deste método pode gerar erros de interpretação, tanto quando ele corta demais as palavras, como também quando ele não corta o suficiente, permanecendo a ambiguidade.

Além de uma etapa de normalização dos dados, uma etapa para extração e seleção de características é essencial, e pode gerar benefícios como: redução da dimensão do problema, que por sua vez aumenta a velocidade de execução do algoritmo; redução na quantidade total de características; aumento na precisão de predição e acerto; e facilitar a visualização dos dados.

Uma vez escolhido um conjunto de características, é possível aplicar algumas técnicas de classificação de texto, inclusive duas das que já foram abordadas aqui, as redes neurais artificiais e o k-vizinhos mais próximos, nos quais apresentam uma precisão superior a 98% quando aplicados ao problema de identificação de *e-mail spam* (Russell & Norvig, 1994).

A classificação ou categorização de texto é portanto a tarefa de, dado algum tipo de texto, decidir a qual conjunto predefinido de classes o mesmo pertence. Por exemplo, decidir a qual linguagem pertence uma sentença ou identificar se um texto de uma chamada de emergência é falsa ou não, são exemplos de classificação de texto.

Capítulo 3

Trabalhos Relacionados

Neste capítulo, serão apresentados os trabalhos e as pesquisas existentes na área que visam solucionar parcialmente ou completamente o problema de identificação de chamadas emergenciais falsas. Um trabalho soluciona parcialmente o problema, se o trabalho dispor alguma metodologia útil para resolver o problema geral. Como por exemplo técnicas de classificação de dados eficientes para texto, ou algum processamento de linguagem natural que facilite a seleção das características relevantes.

Primeiramente, apresenta-se uma revisão na literatura sobre trabalhos que lidam com o problema da classificação de texto, bem como as metodologias propostas e os resultados obtidos. O objetivo desta seção é validar e assim, justificar a escolha dos classificadores que serão utilizados na metodologia deste trabalho.

Em seguida, é discutido o problema gerado pelas chamadas falsas em serviços de emergência em diversos países, como também as consequências deste. Uma análise crítica é feita sobre as soluções propostas e os resultados, até então, obtidos.

Por fim, é apresentado um tópico que aponta os desafios encontrados para trabalhar e desenvolver pesquisas na área, na questão da privacidade e ética sobre os dados dessas chamadas. Bem como, as possíveis vantagens que seriam obtidas caso os dados fossem de mais fácil acesso.

3.1 Classificadores de texto

Algoritmos de classificação de texto vêm sendo aplicados com sucesso em vários diferentes problemas e em diversas áreas de atuação e interesse, tais como rotulação de conteúdos, produtos, multimídia; Gerenciamento de Relacionamento com o Cliente (do inglês: *Customer Relationship Management* - CRM) para empresas; monitoramento de conteúdo e tendências em redes sociais, dentre outros (Uysal & Gunal, 2012; Gupta, 2018). Alguns exemplos desses algoritmos são: Árvore de decisão, Máquina de Vetores

de Suporte (do inglês: *Support Vector Machine* - SVM), RNA e KNN.

Em uma consulta preliminar na literatura disponível, não foram identificados trabalhos e pesquisas que utilizem de algoritmos de classificação de texto para identificação de chamados falsos. Portanto, uma revisão dos principais algoritmos de classificação de texto aplicados a outros contextos é feita a seguir.

Uysal & Gunal (2012) propuseram uma nova metodologia para extração de características em bases de dados textuais. Com o objetivo de avaliar a eficácia do novo método, os autores executaram uma sequência de testes em quatro diferentes bases de dados: *Reuters-21578*, uma base de dados sobre a revista online *Reuters* de 1987; *20 newsgroups*, uma base sobre 18.000 publicações em fóruns de discussão; *Short Message Service* (SMS), uma base de dados sobre o conteúdo de mensagens de texto; e *Enro1*, uma base de dados sobre conteúdo *spam* em *e-mail*. Tais bases de dados possuíam diferentes características, com diferentes medidas de avaliação e junto ao uso de algoritmos de categorização de texto, como Árvore de decisão, SVM e RNA, foram submetidos os testes. Os resultados desse novo método foram promissores, apresentando alto índice de acurácia e o menor tempo de execução dentre os outros métodos de extração de características comparados. Dentre os resultados obtidos, os métodos RNA e SVM se destacaram diante à Árvore de decisão em quase todos os testes, com taxas de precisão superiores a 90%. Mais precisamente, o método RNA apresentou melhores resultados quando foi aplicado junto as bases de dados de *e-mail spam* e mensagens de textos. Ressaltando assim, o potencial e a eficácia do uso da RNA para interpretação de linguagem natural.

O uso do SVM para classificação de texto foi proposto por Joachims (1998). Em sua publicação, Joachims identificou que os textos em geral possuem: um alto volume de dados, com várias características distintas; um baixo índice de características irrelevantes ou inúteis para serem eliminadas; e são linearmente separáveis. Analisando essas características ele deduziu que, qualquer algoritmo de classificação que trabalhe bem nessas circunstâncias, também conseguirá classificar eficientemente um texto. Como forma de validar sua hipótese, Joachims experimentou a eficácia do algoritmo SVM para classificação de texto utilizando as bases de dados *Reuters-21578* e *Ohsumed corpus*, quando comparado ao lado de métodos, até então, consolidados na literatura, como Naive Bayes, algoritmo de Rocchio, KNN e C4.5. Como resultados do experimento, obteve-se sucesso, visto que em média, o SVM acertou corretamente o *target* em 86% dos testes, superando todos os outros. Mais precisamente: Bayes (72%), C4.5 (79.4%), Rocchio (79,9%) e KNN (82.3%). Por outro lado, o custo de treinamento do SVM manteve-se comparado ao C4.5 e mostrou ser um algoritmo mais custoso que os demais métodos citados. Levando em consideração que as bases de dados escolhidas

podem não ter favorecido tanto o KNN, ele apresentou-se como um classificador de texto sólido.

3.2 Chamadas emergenciais falsas: Problemas e Soluções

Chamadas emergenciais falsas são consideradas um fardo para os serviços emergenciais. Tais chamadas são responsáveis por perda de tempo, esforço, energia, recursos, enfim um ônus para o Estado.

Segundo Waseem et al. (2010), uma análise mostrou que 97% dos dados das chamadas emergenciais recebidas entre outubro de 2004 e Maio de 2010, em Punjab, Paquistão, foram identificadas como falsas, mais precisamente, 91,5% destas foram identificadas como trote (brincadeiras ou insultos ao atendente); 7,27% por busca de informações não relacionadas ao serviço; 1,1% de chamadas por engano (discagem de número errado); e 0,13% de chamadas onde o solicitante forjou a necessidade de um atendimento emergencial. Portanto, apenas 3% eram chamadas que realmente resultaram prestação de serviço. Diante disso, ações foram efetivadas pelo governo local: a implantação de um sistema de monitoramento de chamadas e um sistema de bloqueio de chamadas, no qual após três ou quatro chamadas falsas ou trotes originadas do mesmo número, o bloqueio da linha telefônica seria realizado, impossibilitando assim, a efetuação de chamadas temporariamente. Entretanto, observa-se que as medidas propostas não obtiveram grande êxito, uma vez que não houve redução significativa da incidência dessas chamadas indevidas.

Rashford & Isoardi (2010) destacam a importância da otimização e do uso apropriado do sistema de chamadas emergenciais, combatendo os falsos chamados. Os autores realçam o desgaste financeiro que tal prática resulta na sociedade, retirando recursos que poderiam ser aproveitados em pacientes que realmente necessitam desse atendimento e transporte. Como medida preventiva, campanhas públicas de conscientização foram elaboradas, mas novamente não obtiveram os resultados esperados. Resultando assim, a introdução de penas mais severas, na legislação vigente da Austrália, para quem realizasse falsos chamados, podendo ser multado em até \$ 10.000 ou 1 ano de prisão. Ainda sobre a descrição dos serviços de ambulância no referido país, é relatado o uso do Sistema de Prioridades para Despacho Médico (do inglês: *Medical Priority Dispatch System* - MPDS). Neste sistema, o atendente é orientado a efetuar uma série de perguntas ao solicitante, com o objetivo de obter uma melhor descrição da emergência. Após essa triagem primária, a situação é então manualmente classificada em um código alfanumérico previamente definido, composto por: um número, entre 1 a

33 (Tabela 3.1), que descreve o principal sintoma identificado; uma letra, *Alpha*, *Beta*, *Charlie*, *Echo* ou *Omega*, que classifica a prioridade da emergência; e por fim, um outro número, que procura especificar ainda mais o problema. Este método, bem como outros métodos similares, foram elaborados com o objetivo de alcançar um padrão comum, reduzindo o critério subjetivo e pessoal dos atendentes sobre a situação do paciente. Há dados que denotam a eficiência destes métodos, sendo possível obter mais informações sobre o estado do paciente, aumentando assim a chance de sobrevivência, prestando um atendimento mais específico e eficiente (Gray & Walker, 2008).

Os estudiosos relatam ainda que as chamadas menos prioritárias devem ainda passar por uma segunda triagem, sendo esta, por médicos especialistas, com o objetivo de um diagnóstico mais preciso, sendo possível mudar o prognóstico anterior, aumentando o nível de prioridade ou até mesmo não efetivando o despacho da ambulância ao local, desencadeando medidas alternativas para o atendimento (Marks et al., 2002; Gray & Walker, 2008). Embora demonstrado alguma eficiência e otimização no processo, tais métodos ainda são efetuados manualmente pelos operadores. Não sendo aproveitado assim, do uso de ferramentas computacionais que automatizem e facilitem este processo.

Devido ao alto índice de trotes recebidos por serviços de telefonia mundial, tal como o serviço de Operador de Chamadas Internacionais (do inglês: International Operator Direct Calling - IODC), que têm como função conectar um turista em um viagem internacional, através de uma chamada, com um atendente de seu país natal. Kuroiwa et al. (2004) propuseram um sistema de identificação automática de trotes e chamadas falsas, baseado na tecnologia de reconhecimento de fala. Esse sistema foi desenvolvido para operar com base em ligações para o Japão, em que é verificado se a pessoa que deseja utilizar do serviço, compreende bem o idioma japonês. O teste é validado solicitando o usuário a repetir uma palavra em japonês, esperando um determinado tempo pela pronúncia. Caso o usuário pronuncie corretamente a palavra, pressupõem-se que o mesmo conhece o dialeto da região o qual está telefonando, portanto, as chances de ser uma ligação real são altas, sendo assim, o usuário é rapidamente redirecionado para o atendente. Caso contrário, uma segunda e última tentativa de pronúncia é solicitada ao usuário. Se o mesmo errar ou não pronunciar a palavra, um aviso é dado e a ligação é encerrada. Como resultados da metodologia proposta, os autores informaram que depois de analisar cerca de 100.000 chamadas, um total de 9489 foram identificadas como chamadas reais, apresentando uma taxa de acerto de 97% e uma taxa de rejeição de chamadas falsas de 93%. O volume de trotes eram tão alto que houve dias na semana em que a quantidade ultrapassava 6.000 de ligações rejeitadas. Contudo, apesar dos ótimos resultados apresentados pelos autores, esta aplicação é restrita ao contexto de chamadas internacionais e, ainda mais precisamente, específica e possível

Tabela 3.1. Lista de códigos traduzidos do MPDS.

Códigos MPDS			
1	Dor abdominal	18	Dor de cabeça
2	Reação alérgica	19	Problema no coração
3	Mordida por animal	20	Exposição a calor ou frio
4	Assalto	21	Hemorragia
5	Dor nas costas	22	Acidente industrial
6	Dificuldade para respirar	23	Overdose
7	Queimadura	24	Gravidez
8	Exposição a elemento químico perigoso	25	Problema psiquiátrico
9	Ataque cardíaco	26	Doença
10	Dor no peito	27	Esfaqueamento ou tiro
11	Asfixia	28	Acidente vascular
12	Convulsão	29	Acidente de trânsito
13	Diabete	30	Traumatismo
14	Afogamento	31	Indivíduo inconsciente
15	Eletrocussão	32	Sintoma não identificado
16	Problema ocular	33	Cuidado paliativo
17	Queda		

de ser implantada somente em alguns países, tal como o Japão, em que foi conduzido o estudo. Neste sentido, carece ainda o uso de uma ferramenta automatizada para ser aplicada no contexto do Brasil, por exemplo.

3.3 Privacidade dos dados e ética

Das bases de dados sobre chamados emergenciais disponíveis, os conteúdos apresentam-se limitados, ou seja, com poucas informações, o que pode dificultar as pesquisas sobre o tema. Tal limitação denota especialmente no aspecto à privacidade dos dados, uma vez que, por serem sigilosos, não podem ser compartilhados de forma a expor os indivíduos envolvidos.

A garantia da preservação do segredo das informações, além de uma obrigação legal contida na maioria dos Códigos de Ética profissional e também no Código Penal

Brasileiro (Brasil, 1940), é um dever *prima facie*¹ de todos e para com todos. Além do conteúdo sigiloso presente, faz-se necessário a efetivação de uma etapa referente a Ética:

Nas atividades de pesquisa, muitas vezes são utilizados dados constantes em prontuários e bases de dados. Essa utilização deve ser resguardada e permitida apenas para projetos previamente aprovados por um Comitê de Ética em Pesquisa, desde que plenamente descaracterizada a identificação do paciente, inclusive quanto as suas iniciais e registro hospitalar. Mesmo nas publicações científicas não deve ser possível identificar os pacientes através de fotografias ou outras imagens. Em caso de necessidade imperiosa, isto será permitido apenas com o consentimento, por escrito, dos mesmos o que possui amparo na própria Constituição Federal, em seu Art. 5º, item X. (Francisconi & Goldim, 1998).

Entretanto, como apontado por Francisconi & Goldim (1998), existem situações que claramente constituem exceções à preservação de segredos, devido ao risco de vida associado ou ao benefício social que pode ser obtido. Se tratando de uma área com poucas pesquisas realizadas, existem poucos resultados que comprovem a possível eficácia de um sistema que solucione a identificação correta de chamadas falsas em serviços de emergência. Consequentemente, os gestores que administram tais dados se sentem menos propensos a contribuir com pesquisas, por desconhecer o potencial que tais ferramentas poderiam oferecer ao processo.

Em suma, é fundamental compreender, respeitar e tratar com o devido cuidado, todas as informações dos usuários desses serviços emergenciais, como também, a necessidade de desenvolvimento de estratégias para tratar-las de forma eticamente adequada.

¹que se pode constatar de imediato, sem ser necessário examinar melhor; claro, evidente, óbvio.

Capítulo 4

Metodologia

O primeiro passo para a condução deste trabalho foi a verificação da existência de sistemas para identificação de chamadas falsas dentre a literatura acadêmica. Com esse objetivo, diversas consultas em repositórios de artigos e publicações foram feitas, tais como *Google Scholar*, Portal de periódicos da CAPES, *ACM Digital Library* e *dblp computer science bibliography*. Utilizando como índices, palavras chaves como: *emergency calls*, *prank calls*, *hoax calls*, *emergency service*, *911 calls*, *911 emergency calls*, *automatic prank call rejection*. Contudo, os resultados não foram satisfatórios. Ao menos inicialmente, acredita-se que não exista algum sistema de detecção de chamados falsos disponível. Logo, como sua viabilidade ainda é uma incógnita, este trabalho tem como objetivo avaliar a viabilidade de sistemas que identifiquem corretamente quando uma chamada emergencial é falsa ou não.

Para alcançar esse objetivo, é essencialmente necessário dispor de algumas ferramentas, sendo elas:



1. Um banco de dados, que disponha de informações que descrevam as chamadas. Sendo este comum a uma mesma região, em um mesmo idioma, para evitar análises imprecisas.
2. Identificar, selecionar e extrair características, para facilitar a classificação. Como também a criação do *target* que deseja-se prever.
3. Treinar os algoritmos de classificação de texto, como RNA e KNN, definindo seus parâmetros empiricamente.
4. Executar os algoritmos categorizadores e avaliar a predição fornecida por cada classificador.

Ao fim, com os resultados obtidos, será possível mensurar preliminarmente se é viável tal sistema. Apresentando as estatísticas obtidas, as características identificadas e a

performance individual de cada classificador utilizado. É importante ressaltar também, que caso o estudo não conclua positivamente, não significa que não exista viabilidade de tal sistema e sim que não exista nas circunstâncias em que foram testadas, com as especificidades dos dados, o idioma, características e métodos.

4.1 Ferramentas de apoio

Existem diversas ferramentas de apoio disponíveis que serão utilizadas de maneira intermediária para análise da viabilidade desse sistema.

A principal delas é a linguagem de programação Python Python Core Team (2019), que por ser uma linguagem estável, flexível e possuir um grande leque de ferramentas gratuitas disponíveis, é a linguagem de programação mais utilizada para projetos que envolvam aprendizado de máquina e inteligência artificial. Dentre tais ferramentas, algumas serão aplicadas neste projeto, como *scikit-learn* Pedregosa et al. (2011), *pandas* Wes McKinney (2010), *NumPy* Van Der Walt et al. (2011), *Natural Language Toolkit* (NLTK) Loper & Bird (2002) e *matplotlib* Hunter (2007).

A biblioteca *scikit-learn*, dispõem de diversas funcionalidades que irão facilitar a implementação dos algoritmos de classificação e execução dos testes. Por exemplo, ele fornece a implementação da RN supervisionada com topologia FNN, o que em prática significa que somente será necessário selecionar as características, definir o *target* e treinar a rede.

A biblioteca *pandas* fornece ferramentas de análise de dados, estruturas de dados de alta performance e também leitura de arquivos, inclusive no formato *Comma Separated Values* - *CSV* em que se encontra a base de dados planejada.

NumPy é um pacote para a linguagem Python que suporta *arrays* e matrizes multidimensionais, possuindo uma larga coleção de funções matemáticas para trabalhar com estas estruturas. Facilitando assim, as manipulações algébricas e cálculos científicos que venha a ser necessários.

A biblioteca NLTK é um recurso que dispõem diversos métodos para facilitar a manipulação da linguagem natural, como *tokenization*, *stemming*, *tagging*, *parsing*, *semantic reasoning* e *wrappers*. Vindo a ser muito útil durante a etapa de normalização dos dados.

Por fim, a biblioteca *matplotlib* uma biblioteca de plotagem e montagem de gráficos para o Python e sua extensão de matemática numérica *NumPy*. Ele fornece uma API orientada a objetos para incorporar gráficos em aplicativos usando *kits* de ferramentas *Graphical User Interface* - *GUI* para uso geral.

4.2 Base de dados

Devido a dificuldade de encontrar uma base específica para o problema, uma base que será adaptada é *Call Data*, do centro de comunicações do departamento de polícia de Seattle (*Seattle Police Department Communications Center* - SPD), dos Estados Unidos da América. Esta base de dados representa os relatórios das chamadas emergenciais originados pela comunidade local, na qual é armazenado cerca de 95% de todos os chamados recebidos pelo departamento. No total, estão disponíveis 11 colunas de informações que descrevem a ocorrência (Tabela 4.1). Existindo mais de 3,9 milhões ocorrências únicas registradas desde 6 de fevereiro de 2009. Novos dados são inseridos diariamente pela equipe administradora, o que é um ponto positivo, pois é possível estimar o número de chamadas falsas recebidas em relação ao número total de chamados nos últimos anos, como também validar os objetivos ressaltados neste trabalho no cenário atual.

Tabela 4.1. Tabela de atributos que descrevem uma ocorrência da base de dados *Call Data*.

Atributo	Descrição	Formato
ID	Identificador único	Texto
Registro	Como foi resolvido o chamado	Texto
Chamada	Origem da chamada (telefone, 911, alarme)	Texto
Prioridade	Prioridade assimilada ao chamado	Texto
Classificação inicial	Como foi classificado inicialmente a ocorrência	Texto
Classificação final	Como foi classificada pós o acompanhamento	Texto
Data e hora	Data e hora em que foi recebida o chamado	Data e hora
Tempo de chegada	Tempo de chegada do primeiro policial ao local do registro	Texto
Ponto cardeal	Ponto cardeal em relação ao mapa geográfico da cidade	Texto
Setor	Setor do ponto cardeal da ocorrência	Texto
Quadra	Quadra do setor da ocorrência	Texto

Por se tratar de uma base de dados pública e disponível na internet, as informações disponíveis são restritas em relação aos dados originais que são coletados diariamente pelo departamento, pois devido a sua natureza, são informações sigilosas e privadas da comunidade local. Dados como: nome, idade, sexo ou logradouro da ocorrência, foram removidos pelos administradores com o objetivo de preservar a privacidade dos envolvidos.

Outra característica dessa base de dados é que ela está disponível no idioma inglês e com informações e relatórios dos chamados já bem resumidos, o que não possibilita

tanta exploração e uso de técnicas de processamento de linguagem natural. Contudo, como toda base de dados, um pré-processamento é necessário, com objetivo de eliminar ou corrigir erros de ortografia, ambiguidades e informações irrelevantes.

4.3 Cronograma de atividades

O cronograma de atividades planejadas para execução deste projeto foi definido para os próximos cinco meses, começando a partir de agosto até dezembro de 2019. As atividades foram definidas e separadas por etapas (Tabela 4.2), sendo elas:

1. Implementação dos algoritmos, tratamento dos dados e testes.
2. Validação de resultados e discussão.
3. Escrita, conclusão e apresentação dos resultados obtidos.



Tabela 4.2. Cronograma de atividades planejadas para a segunda etapa deste projeto.

Mês/Etapa	Agosto	Setembro	Outubro	Novembro	Dezembro
1					
2					
3					

Referências Bibliográficas

- Aggarwal, C. C. (2014). *Data classification: algorithms and applications*. CRC press.
- Araújo, L. (2016). Trotes prejudicam serviços de emergência e dão prejuízo ao Estado. Disponível em <http://www.mt.gov.br/-/4135140-trotes-prejudicam-servicos-de-emergencia-e-dao-prejuizo-ao-estado>. Acesso em 23 de Junho de 2019.
- Baeza-Yates, R.; Ribeiro-Neto, B. et al. (1999). *Modern information retrieval*, volume 463. ACM press New York.
- Bishop, C. M. (2006). *Pattern recognition and machine learning*. springer.
- Bondy, J. A. & Murty, U. S. R. (1976). *Graph Theory with Applications*. Elsevier, New York.
- Brasil (1940). Decreto-Lei nº 2.848, de 7 de dezembro de 1940. Código Penal. Diário Oficial da União, Rio de Janeiro, 31 dez. 1940.
- Brasil (2003a). Portaria GM/MS n. 1863, de 29 de setembro de 2003. institui a política nacional de atenção às urgências, a ser implantada em todas as unidades federadas, respeitadas as competências das três esferas de gestão. Diário Oficial da União, Brasília, 6 out. 2003.
- Brasil (2003b). Portaria GM/MS n. 1864, de 20 de setembro de 2003. institui o componente pré-hospitalar móvel da política nacional de atenção às urgências, por intermédio da implantação do serviço de atendimento móvel de urgência em municípios e regiões de todo o território brasileiro: SAMU-192. Diário Oficial da União, Brasília, 6 out. 2003.
- Chomsky, N. (2014). *Aspects of the Theory of Syntax*, volume 11. MIT press.
- Dua, D. & Graff, C. (2017). UCI machine learning repository. Disponível em <https://archive.ics.uci.edu/ml/datasets.php?format=&task=cla&>

- [att=&area=comp&numAtt=&numIns=&type=&sort=nameUp&view=table](#). Acesso em 07 de Julho de 2019.
- Duda, R. O.; Hart, P. E. et al. (1973). *Pattern classification and scene analysis*, volume 3. Wiley New York.
- Francisconi, C. F. & Goldim, J. R. (1998). Aspectos bioéticos da confidencialidade e privacidade. *Costa SIF, Oselka G, Garrafa V, coordenadores. Iniciação à bioética. Brasília: CFM*, pp. 269–84.
- Fukunaga, K. & Narendra, P. M. (1975). A branch and bound algorithm for computing k-nearest neighbors. *IEEE Transactions on Computers*, C-24(7):750–753.
- G1 (2014). No amapá, trotes ao 190 geram um custo R\$ 55 milhões, diz polícia militar. Disponível em <http://g1.globo.com/ap/amapa/noticia/2014/12/no-amapa-trotes-ao-190-geram-um-custo-r-55-milhoes-diz-policia-militar.html>. Acesso em 23 de Junho de 2019.
- Gonzalez, M. & Lima, V. L. (2003). Recuperação de informação e processamento da linguagem natural. In *XXIII Congresso da Sociedade Brasileira de Computação*, volume 3, pp. 347–395.
- Graph Online (2015). An online project aimed at creation and easy visualization of graph and shortest path searching. Disponível em <https://graphonline.ru/en/?graph=Cayleygraph>. Acesso em 13 de Maio de 2019.
- Gray, J. & Walker, A. (2008). Ampds categories: are they an appropriate method to select cases for extended role ambulance practitioners? *Emergency Medicine Journal*, 25(9):601–603.
- Gupta (2018). Text Classification: Applications and Use Cases. Disponível em <https://towardsdatascience.com/text-classification-applications-and-use-cases-beab4bfe2e62>. Acesso em 02 de Julho de 2019.
- Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95.
- Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. In *European conference on machine learning*, pp. 137–142. Springer.

- Kuroiwa, S.; Naito, M.; Nakamura, M.; Sakayori, S. & Mukasa, T. (2004). Automatic prank call rejection system for home country direct based on speech recognition technology. *Electronics and Communications in Japan (Part II: Electronics)*, 87(4):44–52.
- Loper, E. & Bird, S. (2002). NLTK: the natural language toolkit. *CoRR*, cs.CL/0205028.
- Machado, C. V.; Salvador, F. G. A. F. & O'Dwyer, G. (2011). Serviço de Atendimento Móvel de Urgência no Brasil: análise da política nacional. *Revista de Saúde Pública*, 45:519 – 528.
- Marks, P.; Daniel, T.; Afolabi, O.; Spiers, G. & Nguyen-Van-Tam, J. (2002). Emergency (999) calls to the ambulance service that do not result in the patient being transported to hospital: an epidemiological study. *Emergency Medicine Journal*, 19(5):449–452.
- Martin, J. H. & Jurafsky, D. (2018). *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition*. Pearson/Prentice Hall Upper Saddle River.
- McCulloch, W. S. & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133.
- Minayo, M. C. d. S. & Deslandes, S. F. (2008). Análise da implantação do sistema de atendimento pré-hospitalar móvel em cinco capitais brasileiras. *Cadernos de Saúde Pública*, 24:1877 – 1886.
- NASA (2007). Nasa ames prognostics data repository. Disponível em <https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/>. Acesso em 07 de Julho de 2019.
- Pacheco, A. G. C. (2016). Agregação de classificadores neurais via integral de choquet com respeito a uma medida fuzzy. Master's thesis, Programa de Pós-Graduação em Informática.
- Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M. & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.
- Peixoto, M.; Sousa, R. A. F. & Odon, T. I. (2015). Combate ao trote telefônico: uma questão de emergência.

- PHMsociety (2009). 2009 phm challenge competition data set. Disponível em <https://www.phmsociety.org/references/datasets>. Acesso em 07 de Julho de 2019.
- Porter, M. F. (1980). An algorithm for suffix stripping. *Program*, 14(3):130–137.
- Python Core Team (2019). *Python: A dynamic, open source programming language*. Python Software Foundation.
- Rashford, S. & Isoardi, K. (2010). Optimizing the appropriate use of the emergency call system, and dealing with hoax callers. *Emergency Medicine Australasia*, 22(5):366–367.
- Rauber, T. W. (2005). Redes neurais artificiais. *Universidade Federal do Espírito Santo*.
- Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386.
- Russell, S. J. & Norvig, P. (1994). *Inteligência artificial*. Pearson Educación.
- Uysal, A. K. & Gunal, S. (2012). A novel probabilistic feature selection method for text classification. *Knowledge-Based Systems*, 36:226–235.
- Van Der Walt, S.; Colbert, S. C. & Varoquaux, G. (2011). The numpy array: a structure for efficient numerical computation. *Computing in Science & Engineering*, 13(2):22.
- Vieira, C. M. S. & Mussi, F. C. (2008). A implantação do Projeto de atendimento Móvel de Urgência em Salvador/BA: panorama e desafios. *Revista da Escola de Enfermagem da USP*, 42:793 – 797.
- Waseem, H.; Durrani, M. & Naseer, R. (2010). Prank calls: a major burden for an emergency medical service. *Emergency Medicine Australasia*, 22(5):480–480.
- Werbos, P. (1974). Beyond regression:"new tools for prediction and analysis in the behavioral sciences. *Ph. D. dissertation, Harvard University*.
- Wes McKinney (2010). *Pandas: Data Structures for Statistical Computing in Python*.
- Widrow, B. & Hoff, M. E. (1960). Adaptive switching circuits. Technical report, Stanford Univ Ca Stanford Electronics Labs.

Apêndice A

Grafos

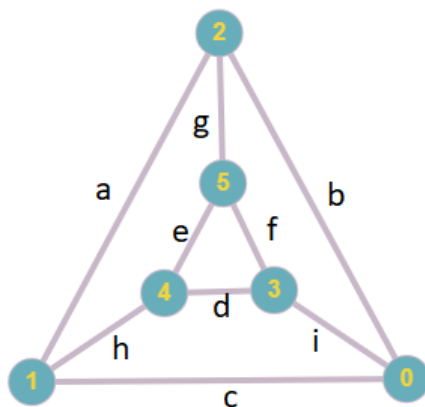
Muitas situações no mundo real podem ser descritas com o uso de um diagrama, composto por um conjunto de pontos e arestas, onde as arestas unem pares desses pontos. Por exemplo, os pontos podem representar cidades em um mapa, e as arestas representariam as estradas que ligam duas cidades. O conceito de grafo parte de uma abstração matemática para caracterizar situações com essas características (Bondy & Murty, 1976).

Matematicamente um grafo G é uma tripla (V, E, ψ) , consistido por um conjunto não vazio de vértices V , um conjunto de arestas E e uma função de incidência ψ que caracteriza quais vértices possuem uma relação (através de uma aresta) com outros vértices. Por exemplo, seja $G = (V, E, \psi)$ um grafo (Figura A.1), tal que $V = \{0, 1, 2, 3, 4, 5\}$, $E = \{a, b, c, d, e, f, g, i\}$ e ψ a função incidência representada na Tabela A.1.

Tabela A.1. Função incidência ψ de G

$\psi_a = 1, 2$
$\psi_b = 2, 0$
$\psi_c = 1, 0$
$\psi_d = 4, 3$
$\psi_e = 4, 5$
$\psi_f = 5, 3$
$\psi_g = 2, 5$
$\psi_h = 1, 4$
$\psi_i = 3, 0$

Segundo Bondy & Murty (1976), os grafos possuem esse nome porque eles possuem uma representação gráfica, e são essas representações que facilitam o entendimento de suas propriedades. De fato, o grafo é uma ferramenta poderosa para simplificação e entendimento de problemas, possibilitando uma modelagem computacional e assim

Figura A.1. Diagrama do Grafo G 

Fonte: Graph Online (2015).

contribuindo com a resolução dos mesmos. Além disso, muitas outras técnicas se beneficiam quando aplicadas utilizando o conceito de grafos, tal como as Redes Neurais Artificiais.