



**UNIVERSIDADE ESTADUAL DA PARAÍBA
CAMPUS I – CAMPINA GRANDE
CENTRO DE CIÊNCIA E TECNOLOGIA
CURSO DE CIÊNCIA DA COMPUTAÇÃO**

ANDRÉ MARQUES GUEDES

**UTILIZAÇÃO DE TÉCNICAS DE APRENDIZAGEM DE MÁQUINA APLICADAS
AO GERENCIAMENTO DE CONSUMO DE ENERGIA EM DISPOSITIVOS
MÓVEIS**

**CAMPINA GRANDE
2017**

ANDRÉ MARQUES GUEDES

**UTILIZAÇÃO DE TÉCNICAS DE APRENDIZAGEM DE MÁQUINA APLICADAS
AO GERENCIAMENTO DE CONSUMO DE ENERGIA EM DISPOSITIVOS
MÓVEIS**

Trabalho de Conclusão de Curso da Universidade Estadual da Paraíba, como requisito parcial à obtenção do título de Graduado em Ciência da Computação.

Área de concentração: Inteligência Artificial.

Orientador: Prof. Dr. Frederico Moreira Bublitz.

Coorientador: Prof. Me. Israel Aires Costa Leal.

**CAMPINA GRANDE
2017**

É expressamente proibido a comercialização deste documento, tanto na forma impressa como eletrônica. Sua reprodução total ou parcial é permitida exclusivamente para fins acadêmicos e científicos, desde que na reprodução figure a identificação do autor, título, instituição e ano do trabalho.

G924u Guedes, André Marques.

Utilização de técnicas de aprendizagem de máquina aplicadas ao gerenciamento de consumo de energia em dispositivos móveis [manuscrito] : / André Marques Guedes. - 2017.

61 p. : il. colorido.

Digitado.

Trabalho de Conclusão de Curso (Graduação em Computação) - Universidade Estadual da Paraíba, Centro de Ciências e Tecnologia, 2017.

Orientação : Prof. Dr. Frederico Moreira Bublitz, Coordenação do Curso de Computação - CCT.

*Coorientação: Prof. Me. Israel Aires Costa Leal, Coordenação do Curso de Computação - CCT.**

1. Aprendizagem de máquina. 2. Gerenciamento de baterias. 3. Dispositivos móveis.

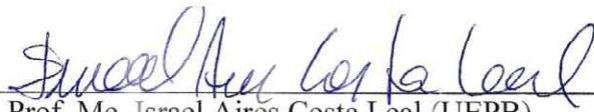
21. ed. CDD 600

ANDRÉ MARQUES GUEDES

**UTILIZAÇÃO DE TÉCNICAS DE APRENDIZAGEM DE
MÁQUINA APLICADAS AO GERENCIAMENTO DE
CONSUMO DE ENERGIA EM DISPOSITIVOS MÓVEIS**

Trabalho de Conclusão de Curso de Graduação
em Ciência da Computação da Universidade
Estadual da Paraíba, como requisito à obtenção
do título de Bacharel em Ciência da
Computação.

Aprovada em 14 de Dezembro de 2017.



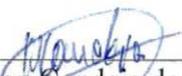
Prof. Me. Israel Aires Costa Leal (UEPB)

Orientador(a)



Prof. Me. Edson Holanda Cavalcante Júnior (UEPB)

Examinador(a)



Prof. Dr. Wellington Candeia de Araújo (UEPB)

Examinador(a)

AGRADECIMENTOS

À paciência de todos os que contribuíram com a minha evolução acadêmica, meus longos momentos de trabalho, dedicação e meus intermináveis aborrecimentos.

Aos meus professores, Prof. Israel e Prof. Frederico, pelo incentivo e presteza no auxílio as atividades desenvolvidas, contribuindo de forma decisiva para o término deste trabalho monográfico, utilizando de toda sua inteligência e capacidade para me passar as informações necessárias de forma clara e objetiva.

A toda minha família, em especial a meu pai e a minha mãe, Antônio e Valdirene, todos fonte imensurável de amor e humildade, responsáveis por minha educação e conseqüentemente caráter.

A minha namorada, Monalisa, que participou deste momento de minha vida, passando-me tranquilidade e fazendo os mais sinceros elogios motivadores.

A todos os meus professores, que ao longo desses quatro anos, ajudaram-me a produzir conhecimento e aprendizagem.

Não poderia deixar de agradecer pelos estágios, primeiramente, no NUTES e depois na empresa AGR Sistemas, a qual vem sendo grande influenciadora para minha formação profissional.

Aos meus amigos e companheiros de turma em geral, dentre outros de igual estima.

A todos que participaram, de forma direta ou indireta, na formação deste trabalho.
Muito obrigado.

“Não deixe o barulho da opinião dos outros abafar sua voz interior. E mais importante, tenha a coragem de seguir seu coração e sua intuição. Eles de alguma forma já sabem o que você realmente quer se tornar. Tudo o mais é secundário”

Steve Jobs

RESUMO

Na medida em que os *smartphones* têm sido incluídos de forma fundamental na vida das pessoas, o alto consumo das baterias dos dispositivos móveis tem gerado expectativas em torno dos fabricantes, pois, estes devem estabelecer um mercado que forneça, cada vez mais, produtos de qualidade, inclusive no que diz respeito às suas baterias e a otimização destas. Neste contexto, também se imagina o impacto que o constante consumo dessas baterias pode vir causar no meio ambiente; sendo assim, buscou-se mostrar métodos, por meio do estudo bibliográfico, capazes de alcançar um melhor gerenciamento da energia fornecida por estas baterias aos dispositivos móveis, utilizando-se da inteligência artificial, mais especificamente o estudo da Aprendizagem de máquina e suas perspectivas. Assim como, em consequência, realizou-se um arranjo experimental por meio dos dados coletados na pesquisa e no próprio experimento realizado, no intuito de solucionar a problemática citada.

Dessa forma, como objetivo geral da pesquisa tem-se: apresentar métodos baseados na aprendizagem de máquina (AM), com o intuito de construir uma revisão bibliográfica e um arranjo experimental voltados a aplicação dos conhecimentos obtidos no gerenciamento do consumo de energia de dispositivos móveis. Assim como os objetivos específicos: mostrar o funcionamento da abordagem de AM e apresentar as suas principais técnicas.

A pesquisa tem cunho qualitativo e quantitativo que, visa a utilização de métodos dedutivos e dialéticos na busca por uma melhor explanação do conjunto de teses formuladas pela literatura especializada em tecnologia e comunicação, relacionada a dispositivos móveis. Para isso, foram utilizados material bibliográfico e periódicos voltados a área, tendo sido os dados confrontados por meio de pesquisas.

Palavras-Chave: Aprendizagem de máquina. Gerenciamento de baterias. Dispositivos móveis.

ABSTRACT

As smartphones have been fundamentally included in people's lives, the high consumption of mobile devices' batteries has created a backdrop of huge expectations for manufacturers as they must establish a market that, quality products, including their batteries and their optimization. In this context, we also imagine the impact that the constant consumption of these batteries can cause in the environment; thus, we tried to show methods, through the bibliographic study, able to reach a better management of the energy provided by these batteries to the Mobile devices, using artificial intelligence, more specifically the study of machine learning and its perspectives. As well as, consequently, na experimental arrangement was made through the data collected in the research and in the experimente itself, in order to solve the aforementioned problem.

Thus, the general objective of the research is to presente methods based on machine learning (AM), with the purpose of constructing a literature review and na experimental arrangement aimed at applying the knowledge obtained in the management of the energy consumption of mobile devices. As well as the specific objectives: show the functioning of the AM approach and presente its main techniques.

The research is qulitative and quantitative, which aims to use deductive and dialectical methods in the search for a better explanation of the set of theses formulated by literature specialized in technology and communication, related to mobile devices. For this, we used bibliographical material and periodicals focused on the area, and the data were confronted through research.

Keywords: Machine learning. Battery management. Mobile devices.

LISTA DE ILUSTRAÇÕES

Figura 1.1 – Evolução de acessos da telefonia móvel	13
Figura 3.1 – Indução de classificador em aprendizado supervisionado	27
Figura 3.2 – Conjunto de treinamento binário e três diferentes hipóteses	27
Figura 3.3 – Árvore de decisão para jogar tênis	32
Figura 3.4 – Relação entre nós e folhas na árvore de decisão	33
Figura 3.5 – Classificação de um exemplo	33
Figura 3.6 – Conjunto e disjunção de atributos na árvore de decisão	34
Figura 3.7 – Representação da árvore de decisão para “Aspecto” sendo “Sol” ou “Chuva”	34
Figura 3.8 – Variação da entropia de acordo com a classificação.....	36
Figura 3.9 – Nó da árvore de decisão referente ao atributo umidade	37
Figura 3.10 – Nó da árvore de decisão referente ao atributo aspecto.....	38
Figura 3.11 – Nó da árvore de decisão referente ao atributo aspecto, detalhado	39
Figura 3.12 – Nó da árvore de decisão referente ao atributo aspecto, com resultado positivo ou negativo para jogo.....	40
Figura 4.1 – Arquivo de alimentação do algoritmo ID3.....	43
Figura 4.2 – Parte do código responsável pelo cálculo da Entropia.....	44
Figura 4.3 – Parte do código responsável pelo cálculo do Ganho.....	44
Figura 4.4 – Parte do código responsável pela chamada da função de construção da árvore.	45
Figura 4.5 – Parte do código responsável pela chamada da função de execução do ID3.....	45
Figura 4.6 – Parte do código responsável pela construção da árvore.....	46
Figura 5.1 – Parte do código responsável pela construção do XML	49
Figura 5.2 – Parte do código responsável pela construção da árvore graficamente	50
Figura 5.3 – Parte do código responsável pela medição da precisão da árvore.....	51
Figura 5.4 – Parte do código responsável pela coleta das métricas.....	52
Figura 6.1 – Variação do tempo de acordo com o método utilizado	54
Figura 6.2 – Valores de mediana da precisão de acordo com o método utilizado	55
Figura 6.3 – Convergência de ganho de informação de acordo com o método utilizado.....	55
Figura 6.4 – Árvore gerada com maior ganho de informação no atributo “HoraAtual”	56
Figura 6.5 – Árvore gerada com maior ganho de informação no atributo “PotenciaRestante”	56

LISTA DE TABELAS

Tabela 1 – Características do Sistema 1G	16
Tabela 2 – Características da configuração de Segunda Geração	16
Tabela 3 – Relação de aplicativos que sobrecarregam o consume de energia das baterias em dispositivos móveis	20
Tabela 4 – Exemplo de um conjunto para classificação	30
Tabela 5 – Exemplo de treino	32
Tabela 6 – Exemplo de treino	36

LISTA DE ABREVIATURAS E SIGLAS

1G	Primeira Geração de tecnologia móvel
2G	Segunda Geração de tecnologia móvel
3G	Terceira Geração de tecnologia móvel
4G	Quarta Geração de tecnologia móvel
5G	Quinta Geração de tecnologia móvel
ANATEL	Administração Nacional de Telecomunicações
AM	Aprendizagem de Máquina
ARFF	Formato de Arquivo de Atributo-Relação (<i>Attribute-Relation File Format</i>)
CDMA	Acesso Múltiplo por Divisão de Código (<i>Code Division Multiple Access</i>)
COMSAT	Corporação Comunicação por Satélite (<i>Communications Satellite Corporation</i>)
D-AMPS	Serviço de telefonia móvel-digital avançado (<i>Digital-Advanced Mobile Phone Service</i>)
DynaTAC	Primeiro celular móvel da Motorola
GSM	Sistema Móvel Global
GPS	Sistema de Posicionamento Global
GB	Gigabyte
Ghz	Gigahertz
IA	Inteligência Artificial
IBGE	Instituto Brasileiro de Geografia e Estatística
mAh	Miliampéres-hora
Mbits/s	Megabits por segundo
MHz	Megahertz
NUTES	Núcleo de Tecnologias Estratégicas em Saúde
NFC	Near Field Communication
RAM	Memória de Acesso Aleatório (<i>Random Access Memory</i>)
TDMA	Acesso Múltiplo por Divisão de Tempo (<i>Time Division Multiple Access</i>)
Wi-Fi	Fidelidade Sem Fio (<i>Wireless Fidelity</i>)

SUMÁRIO

1 INTRODUÇÃO.....	12
1.1 MOTIVAÇÃO E CARACTERIZAÇÃO DO PROBLEMA	12
1.2 OBJETIVO GERAL.....	14
1.3 OBJETIVOS ESPECÍFICOS	14
2 FUNDAMENTAÇÃO TEÓRICA	15
2.1 ABORDAGEM ACERCA DE DISPOSITIVOS MÓVEIS.....	15
2.2 DISPOSITIVOS MÓVEIS E VIDA ÚTIL DE SUAS BATERIAS	19
2.2.1 APLICATIVOS RESPONSÁVEIS POR MAIOR CONSUMO DE ENERGIA NOS DISPOSITIVOS MÓVEIS	20
2.2.2 APLICATIVOS MAIS UTILIZADOS PARA A OTIMIZAÇÃO DE BATERIAS ..	21
2.2.3 VIDA ÚTIL DAS BATERIAS: ALGUNS MÉTODOS DE GERENCIAMENTO DE ENERGIA.....	22
3 APRENDIZAGEM DE MÁQUINA	25
3.1 TIPOS DE APRENDIZAGEM DE MÁQUINA.....	25
3.2 TÉCNICAS DE APRENDIZAGEM DE MÁQUINA	28
3.2.1 CONSIDERAÇÕES AO SE ESCOLHER UM ALGORITMO.....	28
3.2.2 ÁRVORE DE DECISÃO	31
3.2.2.1 ALGORITMO	34
3.2.2.2 WEKA	40
4 METODOLOGIA.....	42
5 ARRANJO EXPERIMENTAL	48
6 RESULTADOS	54
7 CONSIDERAÇÕES FINAIS	58
7.1 TRABALHOS FUTUROS	59
REFERÊNCIAS	60

1 INTRODUÇÃO

O aumento do consumo das baterias dos dispositivos móveis vem influenciando diretamente a fabricação destes, pois a cada dia, as necessidades dos consumidores desses produtos crescem e, com isso, torna-se um mercado cada vez mais exigente. Na medida que os *smartphones* têm sido incluídos de forma fundamental na vida das pessoas, suas aplicabilidades aumentam e, conseqüentemente, o gasto de energia para mantê-los em funcionamento também aumenta.

Dentre as aplicabilidades dos *smartphones*, pode-se destacar a capacidade de armazenar grande número de informações. Tem-se observado que as tecnologias de informação são cada vez mais utilizadas, no intuito de tornar o acervo de informações crescente e o conhecimento mais acessível.

De acordo com essa constante evolução tecnológica, vários desenvolvedores de aplicativos voltados ao gerenciamento de energia têm criado inúmeros aplicativos destinados a otimizar a vida útil das baterias. Os métodos utilizados são os mais diversos, dentre eles podemos destacar o monitoramento da bateria e do tráfego de dados, os sistemas de bloqueio e desinstalação de aplicativos e o processo de finalização de tarefas, eliminando os dados desnecessários (DAQUINO, 2014).

As técnicas de AM têm sido amplamente utilizadas no processo de desenvolvimento de métodos computacionais para a construção de sistemas capazes de obter conhecimento de forma automática, por meio da utilização de algoritmos escolhidos de forma específica. Esse conhecimento pode ser utilizado nas mais diversas áreas do conhecimento, desde a telefonia e dispositivos móveis até na bioinformática (PISTORI E SOUZA, 2010).

Em torno do estudo sobre a aprendizagem de máquina encontram-se subtipos a serem melhores exemplificados ao longo deste trabalho de pesquisa, no entanto, é relevante destacar que a ênfase a ser dada está voltada ao AM supervisionado, pois este tipo apresenta melhores condições para a obtenção dos resultados esperados na pesquisa.

1.1 MOTIVAÇÃO E CARACTERIZAÇÃO DO PROBLEMA

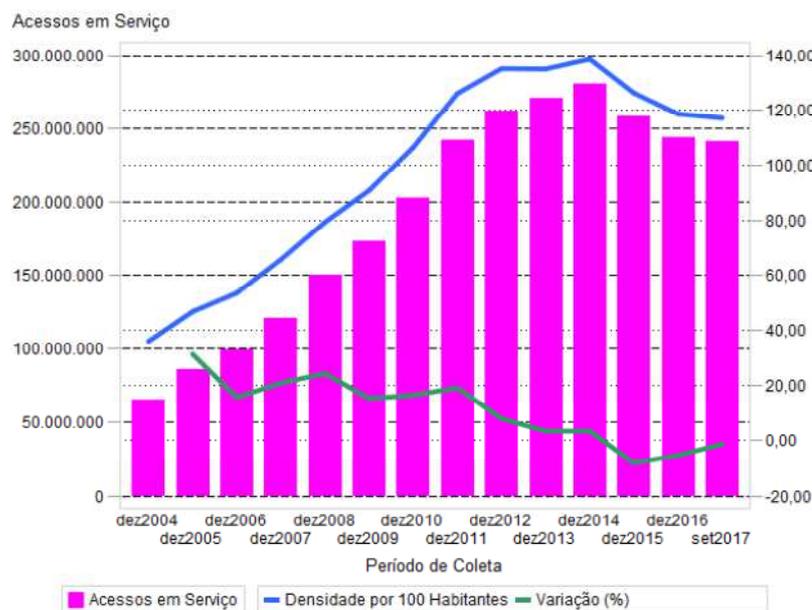
A presente pesquisa tem especial relevância para a área das aplicações móveis, pois a partir do momento que são descobertos novos métodos de otimização do consumo das baterias, será reduzido o descarte dessas, conseqüentemente, haverá um menor impacto para o

meio ambiente. Além disso, em termos econômicos, as empresas que realizam o comércio dos dispositivos móveis trarão um melhor serviço para a população, por meio de dispositivos com maior autonomia.

Essa crescente demanda por dispositivos móveis é detectada por meio de pesquisas estatísticas da ANATEL. Uma dessas pesquisas realizada em 2017, mostra que a telefonia móvel registrou 241.062.955 linhas em operação em setembro de 2017 (ANATEL, 2017).

A Figura 1.1 mostra a evolução dos dispositivos móveis celulares em operação no Brasil. Pode-se observar, o crescimento vertiginoso até 2014 e uma redução do número de usuários entre dezembro de 2004 a setembro de 2017, possivelmente devido a mudança de hábitos do usuário, que passou a usar aplicativos para comunicação de dados e voz como *WhatsApp* reduzindo o número de operadoras por aparelho móvel. Entretanto, esse número é maior que o número de habitantes do país.

Figura 1.1: Evolução de acessos da telefonia móvel.



Fonte: ANATEL, 2017.

Em relação a comparação entre o uso de *tablets*, *desktops* e *notebooks*, “O Brasil vendeu mais *tablets* do que notebooks e computadores de mesa pela primeira vez no ano de 2013 [...]” (GOMES, 2014). Sendo assim, é notória a necessidade da descoberta de métodos que possibilitem uma melhor utilização dos dispositivos móveis, dentro das possibilidades fornecidas pelas baterias.

Um sistema de AM poderá vir a se destacar como uma solução para o problema de otimização de baterias, a partir do momento que, por ser um programa que toma decisões baseadas em experiências acumuladas, induzirá soluções mais efetivas e práticas (BARANAUSKAS, S/D).

1.2 OBJETIVO GERAL

O objetivo geral da pesquisa é apresentar métodos baseados na aprendizagem de máquina, com o intuito de construir uma revisão bibliográfica e arranjo experimental voltados a aplicação dos conhecimentos obtidos no gerenciamento do consumo de energia dos dispositivos móveis.

1.3 OBJETIVOS ESPECÍFICOS

Como objetivo específicos são tratados os seguintes tópicos:

- Mostrar o funcionamento da abordagem de AM;
- Apresentar as principais técnicas de AM;
- Elaborar um arranjo experimental utilizando uma das técnicas de AM, conforme os métodos estudados no presente trabalho.

No Capítulo 2 dessa monografia, busca-se fazer uma abordagem da evolução dos dispositivos móveis e de suas baterias em um contexto geral, de forma que, a partir dessa abrangência, serão demonstradas as formas de utilização dessas baterias nos aparelhos, como as aplicabilidades desses aparelhos influenciam no consumo de energia e as principais formas de otimização.

No Capítulo 3, é abordada a aprendizagem de máquina como uma inovação em inteligência artificial e como o uso dos algoritmos em árvore de decisão pode vir a solucionar os problemas de gerenciamento de energia.

É apresentado no Capítulo 4 a metodologia utilizada na presente pesquisa, baseada na revisão da literatura, no intuito de mostrar os resultados obtidos.

Os resultados e a conclusão da pesquisa são mostrados nos capítulos 5 e 6, respectivamente.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 ABORDAGEM ACERCA DE DISPOSITIVOS MÓVEIS

Desde que o homem passou a utilizar os dispositivos móveis como forma de facilitar as suas atividades diárias e tê-los como uma ferramenta fundamental em sua vida, esses têm se modernizado cada dia mais no intuito de promover um maior atendimento das necessidades buscadas pelo consumidor.

Dessa forma, faz-se necessário mostrar a evolução histórica desses dispositivos móveis, como forma de atentar para as mudanças que estes trouxeram para o nosso meio. Observe-se uma sucinta linha do tempo que constam os principais marcos do avanço tecnológico dos celulares, que são os precursores dessa tecnologia (JORDÃO, 2009):

- 1947 – Começaram as primeiras ideias acerca da telefonia móvel;
- 1973 – Efetuada a primeira chamada de um telefone móvel para um telefone fixo, a empresa que demonstrou o funcionamento do aparelho foi a Motorola, o nome do aparelho era DynaTAC;
- 1983 – Foi liberado o primeiro aparelho para o comércio, DynaTAC 8000x Motorola, pesando cerca 1kg com 30 cm de altura;
- Década de 90 – Os fabricantes estavam prontos para lançar novos aparelhos, seriam a segunda geração de aparelhos com novos padrões de comunicação, essa geração trouxe a tecnologia do SMS (mensagem de texto);
- 1993 – Foi enviada a primeira mensagem de texto;
- 1994 – Surgimento do primeiro Smartphone com características do primeiro conceito de combinação de telefonia com computação;
- Década de 2000 – Surgiram os primeiros celulares com várias cores em suas aplicações, substituindo os aparelhos monocromáticos; Possibilidade de visualização de imagens coloridas, recursos das mensagens multimídias, inclusão das câmeras digitais e finalmente a disponibilidade da internet de modo primitivo (JORDÃO, 2009).

No que diz respeito ao processo evolutivo da rede de telefonia móvel, a partir da propagação de ondas é que foi possível transmitir mensagens a longa distância, sendo assim, o telefone fixo analógico que faz parte da primeira geração, deu seu lugar as novas técnicas denominadas de Segunda Geração (2G), Geração Dois e Meio (2,5G), Terceira Geração (3G) e Quarta Geração (4G) (MENDES, 2014).

A primeira geração analógica surgiu na década de 80, essa geração teria sido criada em 1983 na Austrália, e em seguida, no norte e sul da China. É de se registrar que as principais características dessa geração eram: capacidade limitada e comunicação fraca (MENDES, 2014). A seguir, a Tabela 1, que mostra as principais características:

Tabela 1: Características do Sistema 1G.

Estação Base Txbanda	869 – 894 MHz
Banda MUTx	824 – 549MHz
Canal Bandwidth	30 Khz
Número dos canais de voz	790
Número de canais de controle	42
Potência máxima M U	3 W
Raio de tamanho celular	2 – 20 km
Canais de voz de modulalão	FM
Canais de controle de modulação	FSK

Fonte: MENDES apud DEBNATH, 2014.

De conseguinte, foi criada a 2G de telefonia móvel, mais conhecida como “sinal digital”, ela veio para melhorar o sinal e aumentar a quantidade de usuários com uma melhor qualidade de comunicação (MENDES, 2014).

As principais características do sistema 2G são: é uma tecnologia que comprime e codifica a voz, transformando-a em sinal digital, permite a comunicação simultânea para que duas chamadas sejam feitas ao mesmo tempo. Segue a Tabela 2, a qual mostra tais características:

Tabela 2: Características da configuração da Segunda Geração.

Taxa de transporte de bits	270,8 kbps
Taxa de codificação de voz	13 kbps
Canal de largura de banda	200 KHZ
Usuários por canal	8
Potência máxima móvel	20 W

Fonte: MENDES apud DEBNATH, 2014.

A geração 2,5G é baseada nos sistemas GSM e CDMA os quais permitem o fornecimento de dados por pacote. No entanto, esse serviço de fornecimento de dados evoluiu ao ponto de gerar novos sistemas como *General Packet Radio Service* (GPRS) que trouxe um alívio no processamento dos dispositivos, assim como o *Enhanced Data rates for GSM Evolution* (EDGE), responsável pela triplicação da capacidade de tráfego do GPRS (LEAL, 2015).

A medida que o mercado exigiu cada vez mais da rede de telefonia móvel, houve um avanço na tecnologia digital que permitiu não só a chamada de voz, mas também a comunicação de dados através de imagens, áudio e vídeo. Esse é o marco da chegada da 3G, sistema este que foi implantado pela primeira vez em 2001 no Japão, no mesmo ano na Coreia do Sul, em 2003 nos Estados Unidos e lançado na Índia em 2008. Em 2010, quase todos os países já possuíam a cobertura (MENDES, 2014).

Dentre as principais características da 3G podem ser citadas: multimídia avançada, usabilidade em todos os dispositivos, banda larga e alta velocidade comparada a 2G.

Por fim, ainda se tratando do avanço da rede de telefonia móvel é notório destacar a importância da chegada da 4G, a qual traz uma estrutura composta por um conjunto de redes que usa o IP, fazendo com que vários usuários acessem a rede de internet, obtendo vários tipos de serviços como: dados, fotos e vídeos em qualquer lugar, e ainda fazer uma ligação com alta qualidade. Tal processo é feito por um transmissor (Multiplexação por Divisão de Frequência Ortogonal) que aceita os dados a partir de uma rede IP, convertendo e codificando antes de fazer a modulação.

Abaixo serão elencadas algumas inovações trazidas pelo sistema 4G:

01. Evolução do Projeto de Parceria da Terceira Geração (3GPP) para um aproveitamento mais eficiente das bandas acima de 5 MHz.
02. Uso de técnicas de Múltiplas Entradas e Múltiplas Saídas (MIMO), para tecnologia baseada em Acesso Múltiplo por Divisão de Código (CDMA).
03. Uso da técnica de Multiplexação Por Divisão de Frequência Ortogonal (OFDMA) no downlink e Divisão de Frequência de Acesso Múltiplo por Portadora Única (SC-FDMA) no uplink, para reduzir a Taxa Média de Pico de Pacote (PARP) por ser ineficientes pela perda de alto ganho, com isso reduz a complexidade do dispositivo móvel.
04. Latência abaixo de 5ms com 5 MHz ou largura de banda maior. Com essa alocação de banda abaixo de 5 MHz, a latência abaixo de 10ms pode ser viabilizada.
05. Suporta somente Comutação por Pacote (PS). Ou seja o serviço de voz para usuário é fornecido através de VoIP ou utilizando tecnologias legadas. A largura de banda escalável é até 20 MHz, com bandas menores cobrindo 1,25 MHz, 2,5 MHz, 5 MHz, 10 MHz e 15 MHz. Considerando também a banda 1,6 MHz para casos específicos.
06. A taxa máxima de dados no downlink de até 100 Mbits e uplink de até 50 Mbits/s com largura de banda de 20 MHz. (MENDES, 2014).

Ainda que a tecnologia 3G esteja sendo aperfeiçoada em alguns países e muitos não tenham alcançado o melhor padrão de qualidade, atualmente, fala-se em tecnologia de Quinta Geração (5G), a qual trabalha com interconexão ilimitada da rede sem fio através da convergência e cooperação entre as tecnologias proporcionando uma variedade de serviços com uma alta taxa de transmissão (MENDES, 2014).

A tecnologia 5G foi iniciada como uma mistura de níveis de redes de diferentes tamanhos, com a capacidade de transmitir uma conexão inteligente e heterogênea a qual poderia ser acessada por um grande número de dispositivos sem fio. O avanço desse ideal juntamente com o crescimento da tecnologia de comunicação física avançada e multiplexação espacial de alta ordem de saída e entrada múltipla (MIMO) vieram a ser considerados proporcionadores de uma maior capacidade de acesso simultâneo (MENDES, 2014).

Acrescenta-se que a tecnologia 5G também veio como forma de ampliar a capacidade de acessar todos os aplicativos a partir de qualquer plataforma, em qualquer lugar e em qualquer horário, por intermédio de um ambiente integrado. São atividades como aviso inteligente por mensagem para o celular, a forma de abrir um carro, ou uma conta única para todos os serviços de comunicação independente da aplicação ou rede operadora (MENDES, 2014).

Em continuidade, também é relevante mostrar a evolução histórica das baterias de modo geral, estas que funcionam, atualmente, como fonte de alimentação para o funcionamento dos dispositivos móveis. Segue-se a explanação feita por Tasso Tirapani, (PINTO, apud VEJA, 2012):

1800 – criação da primeira bateria por Alessandro Volta (nome da bateria: Pilha Voltaica);

1836 – criação da primeira composta por uma folha de zinco imerso em uma vasilha de barro poroso com uma solução de sulfato de zinco, por John Frederic Daniell (nome da bateria: Pilha de Daniell);

1844 – Willian Robert Grove criou a Pilha de Grove, composta por uma placa de zinco, imersa em ácido sulfúrico e uma placa de platina imersa em ácido nítrico, separados por uma bateria porosa (possuía quase o dobro da tensão da Pilha de Daniell);

1859 – Gaston Planté criou a Pilha de Chumbo, constituída por uma placa de óxido de chumbo e outra de chumbo metálico, imersa em ácido sulfúrico (a reação química dentro da bateria podia ser revertida);

1860 – Callaud realizou uma melhoria na pilha de Daniell, denominando-a Célula de Gravidade, a qual posteriormente foi utilizada pela rede de telégrafos;

1866 – Carl Gassner criou a Pilha de Zinco-Carbono, misturando o cloreto de amônia com gesso para criar uma pasta junto com o cloreto de zinco (a pilha poderia ser utilizada em qualquer posição);

1899 – Waldemar Jungner criou a Bateria de Níquel-Cádmio a qual consiste em duas placas, uma positiva e uma negativa, que se encontram em um mesmo recipiente, a placa positiva e coberta de hidróxido de níquel e a negativa com cádmio metálico. As placas são envoltas por uma solução de hidróxido de potássio;

1970 – Manley Stanley criou a Bateria de Lítion-Íon que é utilizada atualmente em notebooks e na maioria dos smartphones, é composta por duas placas, uma positiva e uma negativa, a positiva consiste de óxidos de cobalto e lítio sobre uma lâmina de alumínio e a negativa por carbeto de lítio sobre uma lâmina de cobre, separados por uma lâmina de material plástico poroso encharcado com uma solução orgânica de sais de lítio, enrolados em uma forma cilíndrica e prismática;

1977 – a COMSAT criou a Pilha de Hidrogênio a qual é utilizada pelo telescópio Hubble e possui uma placa de níquel conectada a célula de hidrogênio pressurizada, separados por uma membrana de zircônio;

1979 – Michel Armand criou a Bateria de Lítio-Polímero que possui o separador plástico e o eletrólito substituídos por uma folha de polímero condutor e de íons de lítio;

1989 – Daimler AG criou a Bateria de Níquel-Hidreto Metálico, similar à de níquel-cádmio, a não ser pela liga metálica que absorve hidrogênio em vez de cádmio;

Atualmente – as baterias mais utilizadas em dispositivos móveis são a Bateria de Lítio-Íon, Bateria de Níquel-Cádmio e Bateria de Níquel-Hidreto Metálico. (PINTO, apud VEJA, 2012)

A partir dessa contextualização, pode-se começar a compreender como a aplicabilidade dos dispositivos móveis tem aumentado e conseqüentemente a necessidade por baterias que suportem as condições a serem fornecidas por estas novas tecnologias.

2.2 DISPOSITIVOS MÓVEIS E VIDA ÚTIL DE SUAS BATERIAS

A grande maioria dos dispositivos móveis utiliza as baterias de Íon-Lítio. O componente dessas baterias, o Lítio puro é tão reativo que pode incendiar-se se exposto ao ar, nesse caso, a maioria das baterias usam uma opção chamada de óxido de lítio e cobalto. Se esta bateria for carregada, o óxido de lítio e cobalto capturam e armazenam os elétrons, os quais são liberados durante a utilização da bateria (BAGULEY, 2013).

A quantidade de energia a ser armazenada em um menor espaço, nesses dispositivos, é medida pela densidade de energia, em Watts-Hora. Segundo Baguley, uma bateria de Li-íon pode variar entre 150 a 250 Watts-Hora por Kg, enquanto a bateria de Níquel Metal Hidreto a densidade é de 100 Watts-Hora. Dessa forma, percebe-se que a bateria de Li-íon é menor e mais leve que outros tipos, conseqüentemente, mais apropriadas para os aparelhos, gerando uma melhor autonomia (BAGULEY, 2013).

Ainda em relação às baterias, podemos verificar a sua capacidade a partir da medição em Miliampères-Hora, indicando o quanto de energia é fornecida ao longo do tempo. O autor supracitado exemplificou essa medição da seguinte maneira: “Se uma bateria tem capacidade de 1000mAh, significa que pode fornecer uma corrente de 1000 miliampères por uma hora. Se seu aparelho precisa só de 500 miliampères, então a bateria deve durar duas horas” (BAGULEY, 2013).

Não obstante, para medir a autonomia de um aparelho esse método é muito simplório, pois a quantidade de energia a ser utilizada vai variar de acordo com o uso do dispositivo. Sabe-se que os dispositivos móveis utilizam uma grande variedade de aplicabilidades, dentre

elas, o Sistema de posicionamento global (*Global Positioning System* – GPS), *Wireless Fidelity* (WiFi), 3G, chamadas telefônicas, os aplicativos em geral e seu processamento.

Nesse sentido, NarseoVallina Rodriguez, et. al., realizaram um estudo experimental para identificar, em dados estatísticos, quais os principais fatores que afetam a vida útil de uma bateria para dispositivos móveis. O consumo de energia foi mensurado a partir dos valores coletados em avaliações de 18 dispositivos celulares que possuem Sistema operacional Android por um período de duas semanas. Dentre os resultados encontrados na pesquisa em questão, descobriu-se que a avaliação na demanda de energia pode ser dividida em dois grupos, o primeiro grupo foca no impacto da rede sem fio na vida da bateria e o segundo grupo nas técnicas das redes 3G (VALLINA-RODRIGUEZ, 2010).

2.2.1 APLICATIVOS RESPONSÁVEIS POR MAIOR CONSUMO DE ENERGIA NOS DISPOSITIVOS MÓVEIS

Existem aplicativos que são responsáveis por um alto grau de consumo de energia em dispositivos móveis. Segundo Stephan Serowy, alguns aplicativos são responsáveis por um maior consumo em detrimento de outros, ele afirma que estes aplicativos acabam por deixar os dispositivos mais lentos, provocando menor autonomia e conseqüentemente um menor potencial de consumo. A lista de aplicativos responsáveis pelo maior índice de consumo de baterias, de acordo com a pesquisa de Serowy, é mostrada na Tabela 3 (SEROWY, 2015).

Tabela 3: Relação de aplicativos que sobrecarregam o consumo de energia das baterias em dispositivos móveis (de forma automática e manual).

SOBRECARRREGAM O SISTEMA (AUTOMÁTICO)	SOBRECARRREGAM O SISTEMA (MANUAL)
Facebook	Spotify Music
BBM	AmazonKindle
8 Ball Pool	Line Messenger
Instagram	Samsung WatchON
Facebook Messenger	Snapchat
WeChat	Netflix

Fonte: SEROWY, 2015.

Segundo Daniel Veiga, o fato de permanecer conectado a esses aplicativos pode sim ser um grande influenciador para o consumo das baterias, já que o grande número de

notificações recebidas ao longo do dia e execução de atividades acabam consumindo bastante. Sendo que, ele afirma que outras aplicabilidades operacionais destes dispositivos também são “suspeitas” em se tratando do índice de consumo de energia. Como suspeito número um, ele indica o *display* do dispositivo, já que as telas têm grande responsabilidade no gasto de energia, por conta do brilho e de sua constante atividade; de forma a complementar a sua argumentação, Veiga traz como segundo “suspeito” a radiofrequência, exemplificando que o alto consumo estaria relacionado a atividade que o dispositivo exerce na busca do melhor sinal ou frequência (VEIGA, 2013). Veja o trecho da pesquisa em questão:

Telefonia, Rede de dados, WiFi, NFC, Bluetooth. Nossos *smartphones* são transceptores incríveis que, embora sejam dotados de muitas utilidades, nem sempre são necessários. Para controlá-los, a maioria dos fabricantes já inclui no aparelho *widgets* com botões para ligar e desligar estes recursos, geralmente na barra de notificações. Não é milagroso, mas pode ajudar a ganhar tempo extra de bateria (VEIGA, 2013).

O autor supracitado também traz o processamento de aplicativos como o terceiro “suspeito” pelo alto índice de consumo de energia nos dispositivos, afirmando que o processamento a ser utilizado mantém os aplicativos em funcionamento, pois segundo ele, o recurso denominado *wake/locks* é o responsável por manter esse processamento mesmo após a tela ser desligada. Este recurso impede que o processador entre em sono profundo (*deepsleep*), dessa forma, ainda segundo Veiga, mesmo que os *wake/locks* não possam demorar muito em funcionamento, apenas o tempo de realizar as tarefas executadas, alguns aplicativos não se comportam de forma esperada, executando suas funções de forma desenfreada, conseqüentemente, consumindo bastante energia (VEIGA, 2013).

2.2.2 APLICATIVOS MAIS UTILIZADOS PARA A OTIMIZAÇÃO DE BATERIAS

Como visto anteriormente, muitos são os aplicativos que geram maior consumo de energia, no entanto, também existem aplicativos criados no intuito de buscar uma melhor otimização das baterias e assim aumentar o seu tempo de vida útil. Segundo Fernando Daquino, alguns modelos avançados de *smartphones* possuem funções nativas especiais bastante eficientes no controle do consumo de energia, mesmo assim, os aplicativos otimizadores ainda exercem uma importante função neste contexto. Dentre os aplicativos citados pelo autor em questão podemos destacar: *one-TouchbatterySaver*; *ZDbox*; *BatteryDoctor*; *AndroidBoosterFree* (DAQUINO, 2014).

Todos os aplicativos demonstrados no parágrafo anterior estão baseados no método de monitoramento das baterias e de tráfego de dados, no sistema de bloqueio e desinstalação de aplicativos e no processo de finalização de tarefas e eliminação de dados desnecessários.

Por fim, o que pôde-se aferir é que as aplicabilidades em utilização vão gastar mais ou menos energia dependendo das variáveis que o dispositivo apresentar, por exemplo, o aparelho pode encontrar-se em *stand-by* ou desativado. Sendo assim, a aferição do consumo de energia não é tão simples, já que não podemos mensurar perfeitamente as maiores causas do consumo de energia; mesmo assim, sabe-se que o problema de otimização de energia existe e que precisa ser solucionado. Dessa forma, deduz-se que a aplicação da aprendizagem de máquina será um eficiente método para otimizar o gerenciamento das baterias em termos de recursos consumidos.

2.2.3 VIDA ÚTIL DAS BATERIAS: ALGUNS MÉTODOS DE GERENCIAMENTO DE ENERGIA

Em se tratando de um sistema gerenciador de energia, pode-se afirmar que ele é composto por uma série de componentes, dentre os quais destacam-se: um serviço provedor (SP), um solicitador (SR) e um gerenciador de energia (PM). Enquanto o Serviço provedor processa solicitações, o gerenciador de energia emite comandos para alterar o estado deste provedor (LUIZ, et. al, 2013).

Alega a pesquisa que na literatura em estudo foram vistos experimentos como do serviço provedor com dois estados diferentes, ligado e desligado. Em que foi possível concluir que no estado desligado foi tido um baixo consumo de energia, enquanto que no estado ligado, o serviço tem um consumo de energia médio. Quando o serviço está ativamente sendo solicitado, esse gasto de energia cresce, ficando mais alto (LUIZ, et. al, 2013).

Como já foi explanado anteriormente, os dispositivos eletrônicos atuais possuem alto consumo de bateria, consequência das várias interfaces de rede, como o *wifi*, *Ethernet* dados para redes móveis. Dessa forma, conclui-se que a partir do uso diário dessas interfaces de rede, mesmo quando ociosas, consomem bastante energia.

De acordo com a literatura em estudo, existem vários desafios ao estimar a ociosidade das interfaces de rede, porque possuem características, como o tráfego da rede, solicitações dos usuários tendo como exemplo a navegação no e-mail ou mesmo solicitações de sistemas operacionais e aplicativos. Segundo a literatura abordada, nos casos das solicitações dos usuários, como no exemplo da navegação de e-mail, o usuário está consciente que o gasto de

energia será maior, no entanto, no que diz respeito solicitações de sistemas operacionais e de aplicativos, existem técnicas a serem abordadas para estimar se a interface de rede está ociosa ou não (LUIZ, et. al, 2013).

Neste aspecto, a pesquisa em estudo considerou o método de *the timeout policy* (a política de tempo limite), a qual considera que se a duração de um dado intervalo inativo for menor que o tempo limite, então o servidor permanecerá no estado ligado. Enquanto que, se a duração do intervalo ocioso for maior ou igual ao tempo limite, o gerenciador emitirá um comando para desligar (LUIZ, et. al, 2013).

Em outra pesquisa, verificou-se que uma máquina virtual pode ser utilizada como técnica para reduzir o consumo de energia alternando entre os estados ativo e de baixa potência, em vez de manter no estado ativo. Na literatura foi possível evidenciar formas de saber como uma máquina virtual pode fazer uma melhor transição para os estados de baixa potência. No entanto, também foi diagnosticado na pesquisa que essa alternância pode causar um atraso no serviço, por exemplo, pessoas que utilizam serviços na nuvem e ao mesmo tempo recebem uma chamada telefônica. O gerenciador detecta que a Máquina virtual está desligada e pode ser comutada para um estado de menor potência, causando um atraso no serviço (NGUYEN, et. al., 2015).

Na pesquisa supramencionada, foram deixadas duas importantes contribuições para construção de uma melhor forma de otimização das baterias dos dispositivos. Neste aspecto, estas foram as contribuições avaliadas: 1. Política de energia instantânea, na qual o algoritmo foi adaptado para lidar com a desconexão não estacionária; 2. Política de energia baseada em previsão, a qual se baseia em um processo de renovação para encontrar dois pontos de transição entre os estados ativo e inativo (NGUYEN, et. al., 2015).

Ademais, outra pesquisa traz o tema “otimização de políticas dinâmicas para o gerenciamento de energia” como forma de mostrar Esquemas dinâmicos de gerenciamento de energia (também chamados de políticas), os quais podem ser usados para controlar os níveis de consumo de energia dos sistemas eletrônicos. Isto seria possível a partir do estabelecimento dos componentes desses sistemas em estados diferentes, cada um deles caracterizado por um nível de desempenho e um consumo de energia (PALEOLOGO; S/D).

É de se registrar que nesta última pesquisa em foco foi possível avaliar a validade da utilização dessas “políticas ótimas”, ou seja, dessas políticas de otimização, por meio da análise de um disco rígido de um computador. Segundo os próprios pesquisadores, a unidade de disco rígido pode ser operada em cinco estados de energia diferentes, e, em quatro desses

cinco estados, o dispositivo não pode executar leitura ou gravação de dados, portanto, são estados inativos) (PALEOLOGO; S/D).

Ainda de acordo com a pesquisa supramencionada, para que o dispositivo passe de um estado inativo para ativo também haverá gasto de energia. Esse estado inativo seria semelhante ao estado ocioso, pois, neste último caso, a transição do ocioso para o ativo é extremamente rápida. Enquanto no inativo o dispositivo está incapaz de realizar as tarefas necessárias, no ocioso, o disco estaria “espionando”, só esperando o momento de executar tarefas (PALEOLOGO; S/D).

Nesse sentido, os autores da pesquisa em questão utilizaram o método matemático da cadeia de Markov para delimitar a probabilidade de qualquer comportamento futuro do processo não ser alterado pelo conhecimento obtido do seu comportamento passado. Assim, delimitaram dois estados do dispositivo, ativo e inativo, o qual já foi suficiente para caracterizar o sistema. A partir disso foi possível criar os comandos necessários para que a máquina aprenda, dos quais destacam-se três comandos: GOACTIVE (fique ativo); GO-IDLE (fique ocioso); GO-SLEEP (Vá dormir) (PALEOLOGO; S/D).

Portanto, no presente trabalho buscou-se comparar os métodos avaliados nas pesquisas citadas, levando-se em consideração que o gerenciamento de energia baseia-se em um método voltado a inteligência artificial, pois, o aplicativo que venha a ser construído com base na aprendizagem dos dados necessários, quanto a atividade ou inatividade do dispositivo, será capaz de armazenar as informações relevantes para a otimização de energia.

3 APRENDIZAGEM DE MÁQUINA

A AM pode ser considerada uma subárea da inteligência artificial (IA) capaz de desenvolver técnicas de aprendizagem com o intuito de adquirir conhecimento de forma automática. A ferramenta para alcançar a automatização do conhecimento são os algoritmos, os quais são responsáveis pela busca dos classificadores, essenciais à técnica de AM. Existem dois tipos de aprendizagem de máquina: supervisionado e não supervisionado, estes serão melhores descritos na próxima subseção.

3.1 TIPOS DE APRENDIZAGEM DE MÁQUINA

O tipo supervisionado é caracterizado por ter a figura de uma espécie de professor responsável pela apresentação de conjuntos de exemplos que servirão de molde, extrairá a representação do conhecimento a ser obtido a partir dos modelos dados. Ainda de acordo com os autores, o tipo não supervisionado difere-se do primeiro pois não há a presença de um “professor”, sendo assim, não existem modelos prontos (LORENA, 2007).

Tomando como base os conceitos supracitados, entende-se que o aprendizado supervisionado está baseado em uma espécie de treinamento para que sejam seguidas as tarefas a serem executadas. Sendo assim são feitas previsões de acordo com os modelos pré-determinados. Um exemplo prático desse tipo de previsão seria a possibilidade de acertar a partir dos preços atuais do mercado imobiliário, quanto custariam os imóveis em um determinado espaço de tempo (VALINNA-RODRIGUEZ, 2010).

Enquanto isso, o não supervisionado, por não possuir modelos prontos, tenta organizar os dados de maneira a estruturar a execução de tarefas sem um treinamento anterior. Dessa forma, esse tipo acaba por agir de maneira autônoma em relação a produção do conhecimento.

O tipo de aprendizagem supervisionado é costumeiramente mais popular e útil, de modo que existem vários tipos específicos de aprendizagem supervisionado, inclusive, grandes empresas como a Microsoft utilizam desse método para encontrar os melhores padrões possíveis para executar as atividades em suas aplicações (MICROSOFT AZURE, 2016).

A Microsoft Azure, por exemplo, utiliza três tipos específicos de aprendizagem supervisionado: classificação, regressão e detecção de anomalias. Em termos gerais, a classificação é utilizada quando os dados são usados para prever uma categoria, essa classificação pode ser binominal ou de duas classes (MICROSOFT AZURE, 2016).

Em se tratando da classificação binominal, ela é feita quando existem apenas duas opções da classificação dos dados, como seria o caso de um aplicativo de dispositivo móvel que trabalhasse com a obtenção de conhecimento acerca do gerenciamento de energia e que fizesse isto por meio da categorização apenas duas categorias de dados, utilização da rede 3G e utilização de aplicativos do celular no consumo das baterias. Enquanto isso, a classificação de multiclasse trabalha com mais opções de categorização de dados, como por exemplo, um dispositivo móvel em que estejam sendo avaliados a utilização da rede 3G, gastos de energia com a tela e processamento dos dados e aplicativos utilizados.

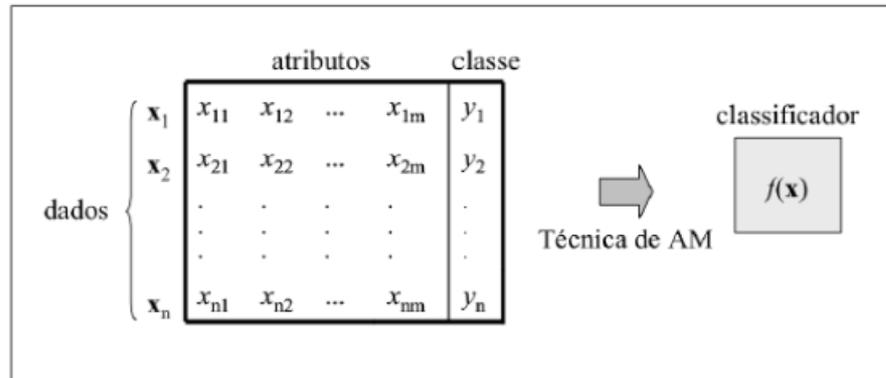
No que diz respeito ao aprendizado supervisionado por regressão, este é utilizado quando um valor é previsto, como por exemplo na avaliação de preços e cotações. Por último, o aprendizado por detecção de anomalias identifica os pontos de dados incomuns, sendo assim, é muito utilizado na detecção de fraudes e outras transações (MICROSOFT AZURE, 2016).

Em relação ao aprendizado não supervisionado é utilizado o tipo específico de clusterização, pelo qual inexistente classe ou categoria pré-definida e é dado um conjunto de observações com o intuito de estabelecer tais classes (OLIVEIRA, S/D). Após a definição das categorias se faz necessário avaliar o que cada agrupamento de dados formado, para entender o significado do conhecimento obtido e assim determinar o contexto do problema.

O processo de aprendizagem de máquina é baseado no método de indução que permite a formação de uma conclusão formada acerca de todos os membros de uma classe através da análise de poucos membros. A esse processo denomina-se processo de indução de um classificador, é feito a partir de uma amostra de dados e denominado treinamento. O classificador obtido também pode ser visto como uma função f , a qual recebe um dado x e fornece uma predição y (LORENA, 2007).

Ainda em relação aos conceitos relacionados à geração de um classificador, os autores supracitados demonstram a representação de uma forma simplificada da obtenção de classificador por meio da aprendizagem supervisionada. Como pode ser visto na Figura 3.1.

Figura 3.1 Indução de classificador em aprendizado supervisionado.



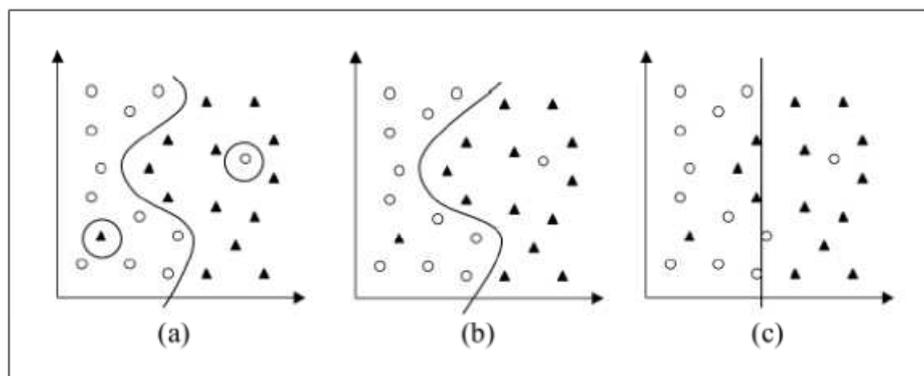
Fonte: LORENA, 2007.

Dessa forma, a partir de um estudo aprofundado acerca do aprendizado supervisionado, pode-se perceber que:

Seja f um classificador e F o conjunto de todos os classificadores que um determinado algoritmo de AM pode gerar. Esse algoritmo, durante o processo de aprendizado, utiliza um conjunto de treinamento T composto de n pares (x_i, y_i) para gerar um classificador particular $f \in F$.” (LORENA, 2007).

Em continuidade, seguindo o conceito de aprendizado supervisionado, sabe-se que este tipo possui a característica de uma espécie de “tutor” que gerará o treinamento explícito pelos autores em questão. O objetivo deste treinamento, ainda segundo os autores, (LORENA, 2007) é encontrar um classificador que separe os dados das classes. Sendo assim, foi utilizado por eles um conjunto de treinamento binário de três diferentes hipóteses, utilizando círculos e triângulos. Observam-se as fronteiras de decisão traçadas na pesquisa citada na Figura 3.2:

Figura 3. 2 Conjunto de treinamento binário e três diferentes hipóteses.



Fonte: LORENA, 2007.

Na Figura 3.2 é possível perceber o conjunto de treinamento binário de três diferentes formas, primeiro é importante explicar que esse treinamento binário é feito a partir da diferenciação de duas classes, as quais, no caso em concreto são mostradas pelos símbolos círculo e triângulo. Nas hipóteses (a) e (b), são exemplificadas as separações de forma não linear, enquanto na hipótese (c), o treinamento é feito de forma linear. Consequentemente são obtidos resultados diferenciados, enquanto o corte não linear consegue dividir melhor as classes, o linear apresenta dificuldade para essa separação (LORENA, 2007).

3.2 TÉCNICAS DE APRENDIZAGEM DE MÁQUINA

Neste ponto da pesquisa, serão apresentadas algumas considerações acerca de técnicas utilizadas para induzir a aprendizagem de máquina, como por exemplo, a escolha de um bom algoritmo, situação esta que acaba por influenciar na extração de um bom classificador e, conseqüentemente, possibilitando a solução da problemática inicial.

3.2.1 CONSIDERAÇÕES AO SE ESCOLHER UM ALGORITMO

Para a escolha de um bom indutor, ou algoritmo de aprendizado, o qual posteriormente será utilizado na extração de um bom classificador, é essencial que sejam listados os atributos necessários. O fato de se utilizar um indutor de conhecimento, é justamente resultado da aprendizagem de máquina usar da inferência como forma de, a partir dos conhecimentos obtidos, prever eventos futuros (OLIVEIRA, S/D).

De acordo com Oliveira, o método de classificação, utilizado pela Aprendizagem de máquina supervisionado tem como primeiro passo “encontrar um modelo para o atributo-meta”, ele considera ainda, que tais atributos assumem valores e são esses valores é que vão definir os treinamentos (OLIVEIRA, S/D).

Ainda segundo o autor supracitado, a classificação pode ser utilizada no objetivo de solucionar diversos obstáculos, dentre eles, “classificar tumores entre benignos e malignos ou classificar transações de cartões de crédito como legítimas ou fraudulentas”. Sendo que um bom classificador necessita apresentar algumas características para que melhor se adeque ao caso concreto, como por exemplo, precisão, velocidade, robustez, escalabilidade (eficiência no banco de dados), interoperabilidade e relevância na seleção de regras (OLIVEIRA, S/D).

Para que haja a classificação se faz necessária a escolha dos atributos necessários, os quais orientarão o algoritmo a partir da definição de seus valores. Nesse processo de escolha

de atributos, alguns métodos vêm sendo utilizados por pesquisadores, aqueles que se baseiam em paradigmas dos mais diversos, os quais de acordo com José Augusto Baranauskas, dividem-se em: paradigma simbólico, paradigma *instancebased* (memorização), paradigma conexionista, paradigma genético e paradigma estatístico (BARANAUSKAS, S/D).

Os paradigmas trabalham com metodologias diferenciadas, por exemplo, enquanto o paradigma *instancebased* trabalha com sistemas *lazy* (preguiçosos), utilizando algoritmos como os *NearestNeighbours* (vizinhos mais próximos) ou Raciocínio baseado em casos, o paradigma simbólico trabalha com algoritmos baseados em árvores de decisão (BARANAUSKAS, S/D).

Em relação ao atributo, esse pode gerar um conjunto de valores, que de acordo com José Augusto, é definido como o “domínio”, para melhor definir essa questão, o autor exemplifica tipos de atributo e domínio. Seguem os exemplos citados pelo autor:

1. Atributo SEXO, Conjunto de valores: $\text{dom}(\text{SEXO}) = \{\text{Feminino}, \text{Masculino}\}$;
2. Atributo COR, Conjunto de valores: $\text{dom}(\text{COR}) = \{\text{Verde}, \text{Vermelho}\}$;
3. Atributo TEMPERATURA, Conjunto de valores: $\text{dom} = \{\text{Baixa}, \text{média e alta}\}$ (BARANAUSKAS, S/D).

Segundo Baranauskas, o atributo que for melhor avaliado será considerado especial e esse será considerado uma CLASSE, responsável por descrever o fenômeno de interesse, ou seja, a meta que se deseja aprender para que no futuro sejam realizadas as previsões necessárias (o conjunto de exemplos). Na aprendizagem supervisionada, os conjuntos de exemplos são definidos de acordo com sua classe, dessa forma, o mesmo autor ainda trouxe um exemplo de um conjunto de exemplos a ser utilizado em uma classificação, como pode ser observado a seguir (BARANAUSKAS, S/D):

Exemplo de um Conjunto de Exemplos para Classificação:

- a) Dez exemplos $n = 10$
- b) Duas classes $k = 2$, (Sim;Não)
- c) Dois atributos $m = 2$:
 - a. Idade é ordenado
 - b. Tipo de Veículo é categórico (Esporte, Van, Caminhão)
- d) Rótulo da classe indica se a pessoa comprou o produto
- e) Atributo dependente (classe) é categórico

O que ficará evidenciado pela **Tabela 4**, (BARANAUSKAS, S/D):

Tabela 4: Exemplo de um conjunto para classificação.

IDADE	VEÍCULO	CLASSE
20	V	SIM
30	V	SIM
25	C	NÃO
30	E	SIM
40	E	SIM
20	C	NÃO
30	V	SIM
25	V	SIM
40	V	SIM
20	E	NÃO

Fonte: BARANAUSKAS, S/D.

Em resumo, existem algumas definições importantes a serem feitas em relação ao aprendizado de máquina, como por exemplo, diferenciar os termos atributo, indutor, exemplo e classe. Para o autor, o indutor é o algoritmo de indução que visa extrair um bom classificador; o exemplo seria um dado ou registro; o atributo descreve uma característica do exemplo e a classe é um atributo especial. Ele ainda trabalha com outros conceitos, como por exemplo, o das *Bias*, as quais são espécies de preferência de uma hipótese sobre a outra. (OLIVEIRA, S/D).

As equações relativas a medida de desempenho de um classificador, como o de erro e o de acurácia também são relevantes para o autor supracitado, na medida em que elas definem expressamente os valores a serem alcançados, veja as equações 3.1, 3.2 e 3.3 (OLIVEIRA, S/D):

$$\text{Considerando, } E = \begin{cases} 1 & \text{se a expressão for verdadeira} \\ 0 & \text{caso contrario,} \end{cases} \quad (3.1)$$

$$\text{err } h = \frac{1}{n} \sum_{i=1}^n ||y_i \neq h(x_i)|| \quad e \quad (3.2)$$

$$\text{acc } h = 1 - \text{err}(h), \quad (3.3)$$

Outros conceitos, como o de distribuição de classes e erro majoritário, afirmam que a distribuição de classes é uma forma de definir as classes majoritária e minoritária em um

conjunto de exemplos, veja-se de forma exemplificada, a demonstração da definição trazida pelo autor supramencionado (OLIVEIRA, S/D).

Distribuição de classes $distr(C_j)$: Para cada Classe C_j , sua distribuição $distr(C_j)$ é calculada como sendo o número de exemplos em T que possuem classes C_j dividido pelo número total de exemplos (n), ou seja, a proporção de exemplos em cada classe, conforme a Equação 3.4:

$$distr C_j = \frac{1}{n} \sum_{i=1}^n ||y_i = C_j||. \quad (3.4)$$

Exemplo: Conjunto com 100 exemplos: 60 Classe C_1 , 15 Classe C_2 , 25 Classe C_3 , de acordo com a Equação 3.5:

$$distr C_1, C_2, C_3 = 60\%, 15\%, 25\% . \quad (3.5)$$

Nesse exemplo, Classe Majoritária (ou Prevalente) é C_1 , enquanto a Classe Minoritária é C_2 .

O conceito de erro majoritário, o qual seria o limite máximo abaixo do qual o erro do classificador deve ficar também é mencionado na exemplificação de definições relacionadas à Aprendizagem de máquina. Observe-se:

Erro Majoritário $maj - err T$: Limite máximo abaixo do qual o erro de um classificador deve ficar, observe a equação 3.6:

$$maj - err T = 1 - \max_{i=1, \dots, k} distr C_i . \quad (3.6)$$

3.2.2 ÁRVORE DE DECISÃO

De acordo com Oliveira, as árvores de decisão são espécies de fluxograma com a estrutura de uma árvore, as quais são capazes de representar testes em relação a atributos. Para o autor em questão, cada ramo da árvore representa um resultado do teste. Ainda segundo Oliveira, a geração de uma árvore se baseia em duas fases: 1. Construção da árvore, momento em que seria feito o particionamento dos atributos, e 2. A fase da poda, responsável pela identificação e remoção dos ramos que refletem ruídos ou interferem nos bons resultados dos testes (OLIVEIRA, S/D).

Em continuidade, (OLIVEIRA, S/D) traz um exemplo de árvore de decisão em se tratando do EXEMPLO: Condições do tempo para jogar tênis. É possível observar na Tabela 5, o passo a passo da construção dessa árvore.

01. Para construir a Árvore de Decisão de Jogar Tênis são tidos em conta exemplos (dias) passados, conforme Tabela 5 (OLIVEIRA, S/D).

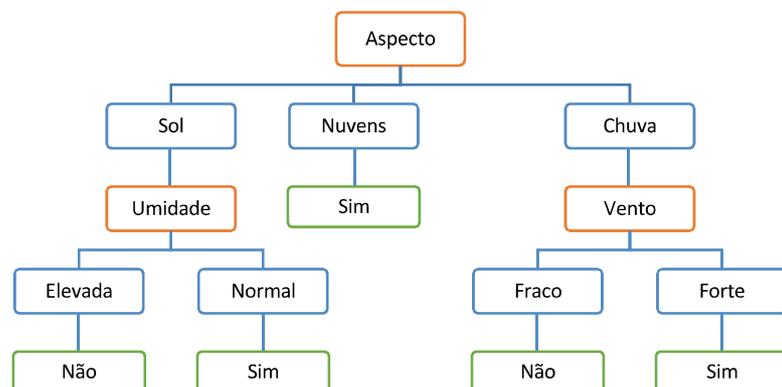
Tabela 5: Exemplo de Treino.

Dia	Aspecto	Temperatura	Umidade	Vento	Jogar Tênis
D1	Sol	Quente	Elevada	Fraco	Não
D2	Sol	Quente	Elevada	Forte	Não
D3	Nuvens	Quente	Elevada	Fraco	Sim
D4	Chuva	Ameno	Elevada	Fraco	Sim
D5	Chuva	Fresco	Normal	Fraco	Sim
D6	Chuva	Fresco	Normal	Forte	Não
D7	Nuvens	Fresco	Normal	Fraco	Sim
D8	Sol	Ameno	Elevada	Fraco	Não
D9	Sol	Fresco	Normal	Fraco	Sim
D10	Chuva	Ameno	Normal	Forte	Sim
D11	Sol	Ameno	Normal	Forte	Sim
D12	Nuvens	Ameno	Elevada	Forte	Sim
D13	Nuvens	Quente	Normal	Fraco	Sim
D14	Chuva	Ameno	Elevada	Forte	Não

Fonte: OLIVEIRA, S/D

02. Através destes exemplos é possível construir a árvore de decisão da Figura 3.3.

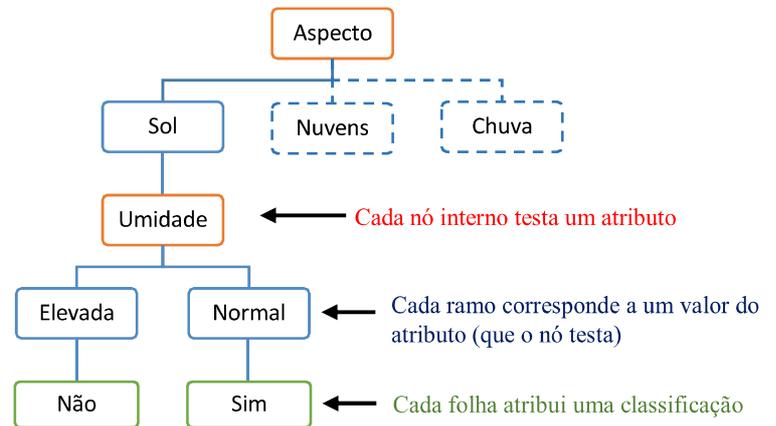
Figura 3.3: Árvore de Decisão para Jogar Tênis.



Fonte: OLIVEIRA, S/D.

03. A relação entre os elementos da árvore (nós e folhas) e os atributos, valores e classificações pode ser entendida na Figura 3.4.

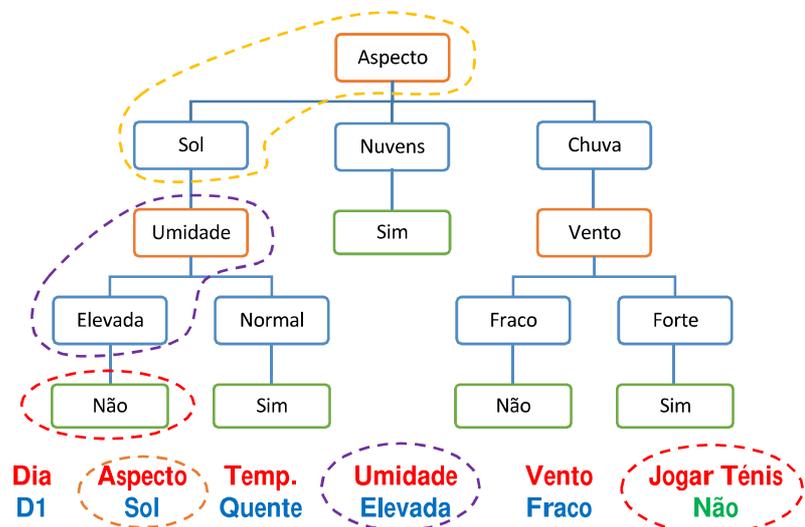
Figura 3.4: Relação entre nós e folhas na árvore de decisão.



Fonte: OLIVEIRA, S/D.

04. A classificação de um exemplo de acordo com a esta árvore é feita da conforme Figura 3.5.

Figura 3.5: Classificação de um exemplo.

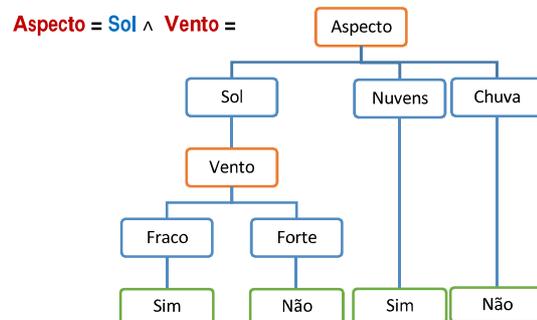


Fonte: OLIVEIRA, S/D.

05. O atributo “Aspecto” tem o valor “Sol” e a “Umidade” tem o valor “Elevada”. O exemplo é classificado como “Não”, ou seja quando esteve sol e umidade elevada não se jogou ténis. Os atributos “Temperatura” e “Vento” não são considerados, pois são desnecessário para classificar este exemplo;

06. Com Árvores de Decisão é possível representar a conjunção e disjunção de atributos. A árvore de decisão que representa a classificação para os dias em que o “Aspecto” é “Sol” e que o “Vento” está “Fraco” encontra-se na Figura 3.6.

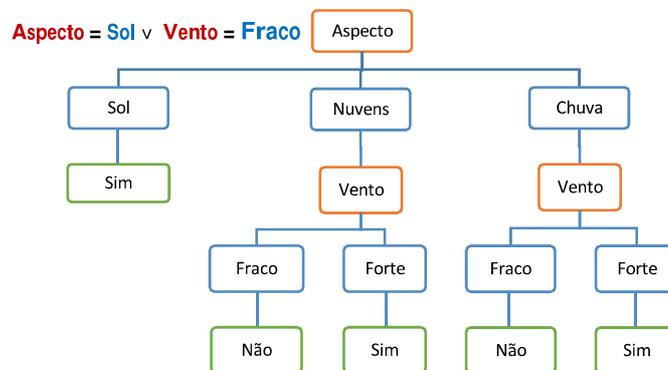
Figura 3.6: Conjunção e disjunção de atributos na árvore de decisão.



Fonte: OLIVEIRA, S/D.

07. A árvore de decisão que representa os dias em que o “Aspecto” é “Sol” ou o “Vento” está “Fraco” é dada por (OLIVEIRA, S/D) na Figura 3.7.

Figura 3.7: Representação da árvore de decisão para “Aspecto” sendo “Sol” ou “Chuva”.



Fonte: OLIVEIRA, S/D.

3.2.2.1 ALGORITMO

Segundo Simone C. Garcia e Luis O. Alvares, “uma árvore de decisão tem a função de particionar recursivamente um conjunto de treinamento, até que cada subconjunto obtido deste particionamento contenha casos de uma única classe”. Para esses autores, os algoritmos ID3 e C4.5 foram introduzidos no intuito de induzir os modelos de classificação, mais especificamente as árvores de decisão (GARCIA; ALVARES, S/D).

Outros algoritmos, como o exemplo do *Support Vector Machines* (SVM), o qual é uma abordagem geométrica utilizada para solucionar o problema de classificação, também pode ser utilizado no conjunto de treinamento no intuito de “dividir” os elementos positivos dos negativos no espaço em estudo (OGURI, 2006).

Dentre os algoritmos pesquisados, o que melhor se adaptou na situação problema foi o algoritmo ID3, o qual teria sido um dos primeiros em se tratando da metodologia de construção de árvore de decisão, ele tem a função de construir a árvore de decisão, a partir de um conjunto de exemplos, como já fora demonstrado anteriormente.

O ID3 tem a responsabilidade de separar um conjunto de treinamento em subconjuntos, dessa forma, serão obtidos exemplos de uma única classe. Essa divisão seria efetuada por meio da escolha do atributo, o qual é selecionado mediante uma probabilidade estatística, denominada ganho, responsável pela medida do valor do atributo (GARCIA; et. al., S/D).

Após a construção da árvore, seriam analisadas as generalizações dos dados que possam ter ocorrido e como essa generalização se comportaria em casos futuros, dessa forma, se estimam os erros e acertos ocorridos.

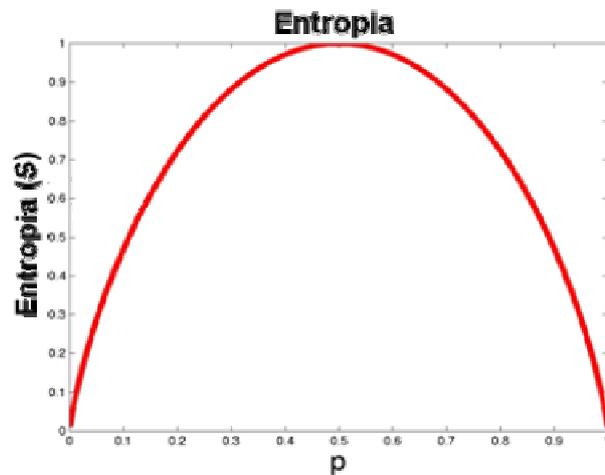
Para que seja feita a escolha do melhor atributo, seriam realizados cálculos estatísticos, entropia e ganho, os quais, por fim, definem o valor de cada atributo trazido no exemplo. Segue a sequência deste cálculo, ainda de acordo com o exemplo de (OLIVEIRA, S/D), em relação ao jogo de tênis:

01. A entropia de um conjunto pode ser definida como sendo o grau de pureza desse conjunto. Este conceito emprestado pela Teoria da Informação define a medida de "falta de informação", mais precisamente o número de *bits* necessários, para representar a informação em falta, usando codificação máxima. Dado um conjunto S , com instâncias pertencentes à classe i , com probabilidade p_i , pode-se descrever a entropia E conforme a seguinte equação 3.7:

$$E S = - \sum p_i \log_2 p_i \quad (3.7)$$

02. No exemplo apenas existem duas classes de classificação, ou seja, "Jogar Tênis" (positivo, +) ou "Não Jogar Tênis" (negativo, -). Assim sendo, o valor da entropia varia de acordo com o gráfico mostrado na Figura 3.8.

Figura 3.8: Variação da entropia de acordo com a classificação.



Fonte: OLIVEIRA, S/D.

03. Sendo S o conjunto de exemplo de treino; p_+ a porção de exemplos positivos; p_- a porção de exemplos negativos, a entropia E será dada pelo desdobramento da equação 3.8:

$$E(S) = -p_+ \log_2 p_+ - p_- \log_2 p_-; \quad (3.8)$$

04. O ganho (*gain*) define a redução na entropia E . $G(S,A)$ significa a redução esperada na entropia de S , ordenando pelo atributo A . O ganho é dado pela equação 3.9:

$$G(S,A) = E(S) - \sum_{v \in \text{values } A} \frac{S_v}{S} \cdot E(S_v) \quad (3.9)$$

05. Para o primeiro passo são analisados todos os atributos, começando pela “Umidade”, por exemplo, o demonstrado na Tabela 6 (OLIVEIRA, S/D).

Tabela 6: Exemplo de Treino. Fonte:(OLIVEIRA, S/D)

Dia	Aspecto	Temperatura	Umidade	Vento	Jogar Tênis
D1	Sol	Quente	Elevada	Fraco	Não
D2	Sol	Quente	Elevada	Forte	Não
D3	Nuvens	Quente	Elevada	Fraco	Sim
D4	Chuva	Ameno	Elevada	Fraco	Sim
D5	Chuva	Fresco	Normal	Fraco	Sim
D6	Chuva	Fresco	Normal	Forte	Não
D7	Nuvens	Fresco	Normal	Fraco	Sim
D8	Sol	Ameno	Elevada	Fraco	Não
D9	Sol	Fresco	Normal	Fraco	Sim
D10	Chuva	Ameno	Normal	Forte	Sim

D11	Sol	Ameno	Normal	Forte	Sim
Dia	Aspecto	Temperatura	Umidade	Vento	Jogar Tênis
D12	Nuvens	Ameno	Elevada	Forte	Sim
D13	Nuvens	Quente	Normal	Fraco	Sim
D14	Chuva	Ameno	Elevada	Forte	Não

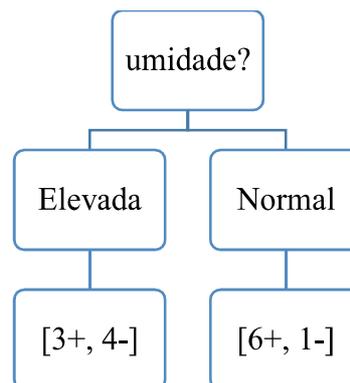
Conforme a Tabela 6 serão analisados todos os atributos, começando pela “umidade”:

Se “S” é o conjunto de exemplo de treino, temos 9 (nove) possibilidades positivas para a ocorrência do jogo de tênis e 5 (cinco) negativas, conforme as equações 3.10, 3.11, 3.12, 3.13 e 3.14 e Figura 3.9, então:

$$S = 9+, 5- \quad (3.10)$$

$$E = 0,940 = \frac{-9}{14} \log_2 \frac{9}{14} - \frac{5}{14} \log_2 \frac{5}{14} \quad (3.11)$$

Figura 3.9: Nó da árvore de decisão referente ao atributo umidade.



Fonte: OLIVEIRA, S/D.

$$G_{S,umidade} = 0,940 - \left(\frac{7}{14}\right) \times 0,985 \quad (3.12)$$

$$- \frac{7}{14} \times 0,592 = 0,151 \quad (3.13)$$

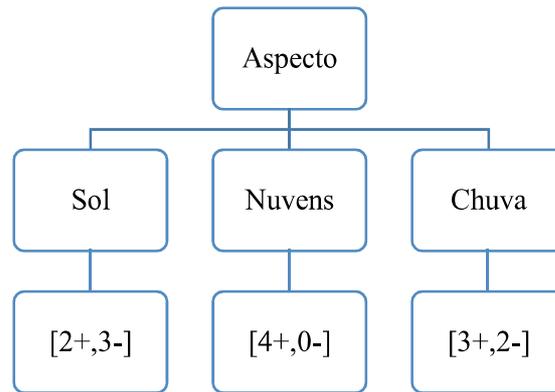
$$G_{S,umidade} = 0,151 \quad (3.14)$$

06. Ainda conforme a Tabela 3, calcula-se o ganho para todos os atributos, verificamos que o tem maior ganho é o **Aspecto**, conforme equações 3.15, 3.16, 3.17 e 3.18 e Figura 3.10.

$$S = 9+, 5- \quad (3.15)$$

$$E = 0,940 \quad (3.16)$$

Figura 3.10 :Nó da árvore de decisão referente ao atributo aspecto.



Fonte: OLIVEIRA, S/D.

$$E = 0,971$$

$$E = 0E = 0,971$$

$$\begin{aligned}
 G_{S,Aspecto} &= \\
 &= 0,940 - \frac{5}{14} \times 0,971 \\
 &\quad - \frac{4}{14} \times 0,0 \\
 &\quad - \frac{5}{14} \times 0,0971 \\
 &= \mathbf{0,247}
 \end{aligned} \tag{3.17}$$

$$\text{MAX} \left\{ \begin{array}{l} \phantom{\text{MAX}} \\ \phantom{\text{MAX}} \\ \phantom{\text{MAX}} \end{array} \right. \tag{3.18}$$

dessa forma, o Atributo com maior Ganho é “Aspecto”.

07. No próximo passo o atributo “Aspecto” já não é tido em conta, conforme a Figura 3.11 e equação 3.19:

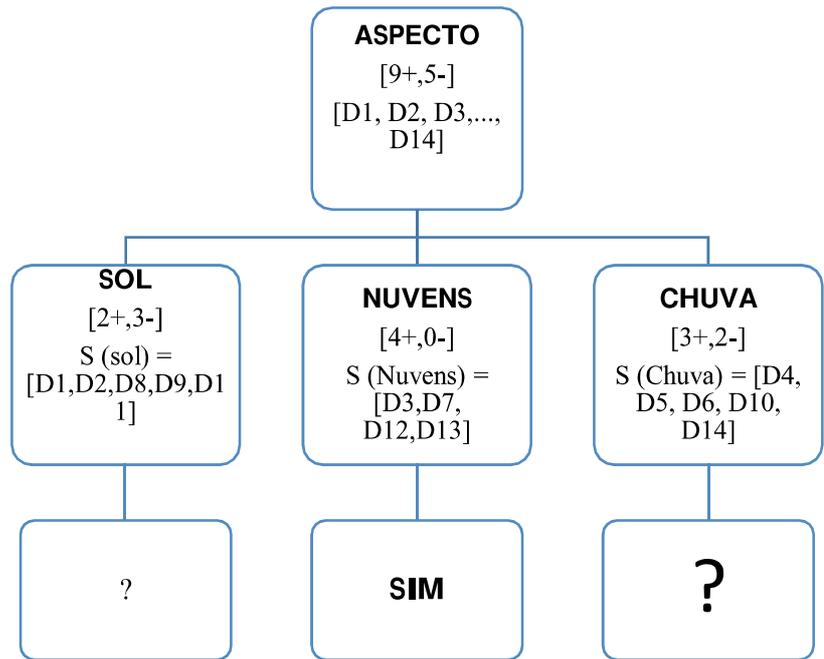
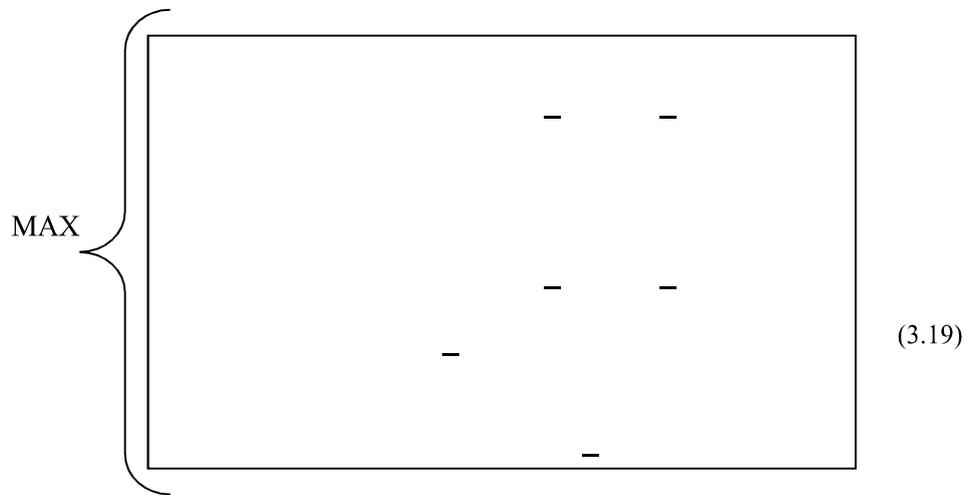


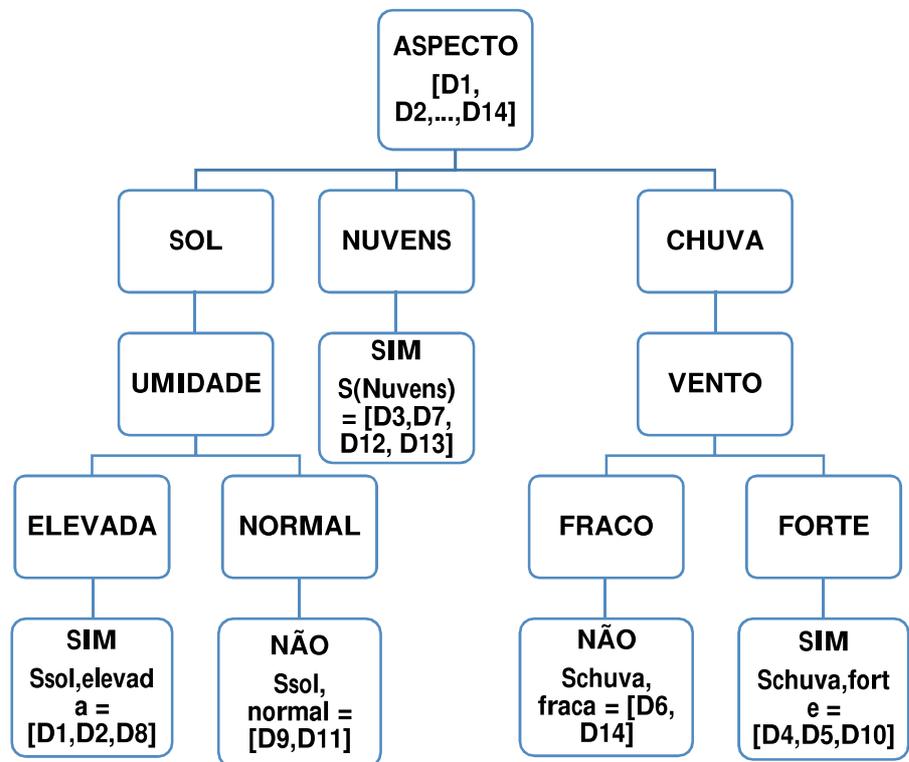
Figura 3.11: Nó da árvore de decisão referente ao atributo aspecto, detalhado (OLIVEIRA, S/D)



Dessa forma, o maior Ganho constatado está no conjunto $Ganho(Ssol, umidade)$

08. Quando a entropia da fórmula obtém-se a seguinte árvore de decisão, conforme a Figura 3.12 (OLIVEIRA, S/D):

Figura 3.12: Nó da árvore de decisão referente ao atributo aspecto, com resultado positivo ou negativo para jogo



Fonte: OLIVEIRA, S/D.

Por fim, é possível perceber, que a partir da construção da árvore de decisão por meio do algoritmo ID3, será gerado o conhecimento necessário para o programa a ser desenvolvido. Sendo assim, a problemática do gerenciamento de energia dos dispositivos móveis pode ser solucionada por meio de um algoritmo ID3 que permita a construção de uma árvore de decisão, gerando-se, conseqüentemente, o conhecimento necessário para a realização das tarefas de gerenciamento, por meio dos atributos definidores.

3.2.2.2 WEKA

O WEKA pode ser considerado como um pacote de *software* o qual começou a ser desenvolvido em 1993, utilizando a linguagem JAVA, na universidade de Waikato, Nova Zelândia. Atualmente esta ferramenta encontra-se licenciada pela *General Public License* (GNU), dessa forma, é possível estudar e manipular o possível código fonte (WEKA, S/D).

O principal objetivo dessa plataforma é agregar algoritmos de diferentes abordagens na subárea da aprendizagem de máquina, portanto, permitir que um computador possa

aprender, no sentido de obter novo conhecimento, os quais permitam a indução dedutiva da máquina.

Sendo assim, a partir do que foi explanado, nos próximos capítulos, serão destacados a metodologia de pesquisa, bem como as considerações finais acerca do objeto de estudo em análise.

4 METODOLOGIA

Trata-se de uma pesquisa qualitativa e quantitativa que visa a utilização de métodos dedutivos e dialéticos na busca por uma melhor explanação do conjunto de teses formuladas pela literatura especializada em tecnologia e comunicação, relacionada a dispositivos móveis. Foram utilizados material bibliográfico e periódicos voltados a área, tendo sido os dados colhidos na pesquisa confrontados com um arranjo experimental.

Este estudo está baseado na análise de experimentos realizados por pesquisadores e especialistas na área de inteligência artificial. Os experimentos obtidos figuram-se na utilização das técnicas de aprendizagem de máquina no intuito de gerar um conhecimento automático para dispositivos móveis em relação ao padrão de consumo de energia de suas baterias, para que dessa forma, seja possível otimizar o consumo.

A pesquisa foi iniciada pela avaliação de quais as principais variáveis que afetam a vida útil das baterias. A partir dos resultados, foram estabelecidas as hipóteses do alto consumo de energia. Sendo assim, foi possível determinar o conjunto de atributos a serem considerados para a construção da árvore de decisão, a qual é um método assimilado pela inteligência artificial, bem como, do algoritmo a ser utilizado para construção da árvore de decisão.

As bases de pesquisa do presente trabalho estão baseadas em dados de pesquisas de âmbito acadêmico, como trabalhos de conclusão de curso e teses de mestrado e doutorado; assim como foi avaliado o estudo específico de aulas baseadas em Inteligência artificial, é o caso do estudo do professor Stanley R. M. Oliveira, o qual é graduado em Ciência da Computação pela Universidade Federal de Campina Grande, PB (1990), mestre em Ciência da Computação pela Universidade Federal de Campina Grande, PB (1995) e PhD em Ciência da Computação pela Universidade de Alberta, Canadá (2005).

Portanto, buscou-se fazer uma revisão bibliográfica a partir do estudo das pesquisas realizadas pelo Professor supracitado, sendo possível verificar que o método “Árvore de decisão” é capaz de fornecer meios de classificação de dados, os quais foram listados em um conjunto de treinamento capaz de produzir conhecimento, esta foi a motivação para a escolha da árvore de decisão como método utilizado.

O exemplo trazido pelo professor em questão foi o jogo de tênis, as variáveis climáticas para quando ocorre ou não o jogo, isso é uma classificação binária. Ao transferir esse método para o estudo do gerenciamento das baterias de dispositivos móveis, é possível

alinhar uma série de características de consumo de bateria que podem ser utilizadas para gerar o conhecimento à máquina.

No contexto do experimento, é possível verificar características como: gasto de energia da tela do celular, gasto com aplicativos, gasto com o processamento de dados, com o gerenciamento das redes móveis, o que possibilita a máquina aprender os dados necessários para gerenciar esse consumo.

Nesse sentido, a partir das pesquisas em periódicos, artigos e literatura em geral, buscou-se tentar adequar o método da Inteligência Artificial em questão com um arranjo experimental realizado por meio da verificação da funcionalidade do aplicativo *Whatsapp* no sistema operacional *Android*, através da utilização do algoritmo ID3, o qual é capaz de identificar a melhor forma de mudar o estado do aplicativo discutido, estado este que pode ser *ativo*, *inativo* ou em *stand-by*, conforme as pesquisas realizadas. Tal implemento foi feito a partir da verificação de que, atualmente, os aplicativos utilizados nos dispositivos móveis que fazem uso do *Android* demoram cerca de três dias para entrar no modo inativo.

Conforme estabelecido no parágrafo anterior, o algoritmo utilizado no estudo foi o ID3, o qual é alimentado de acordo com os seguintes parâmetros: *@relation* – nome do relacionamento; *@attribute* – nome do atributo e o espaço amostral ao qual poderá receber, *@data* – dados que irão alimentar o ID3. Segue-se a Figura 4.1, a qual demonstra um exemplo de alimentação do ID3:

Figura 4.1: Arquivo de alimentação do algoritmo ID3.

```

1 @relation battery
2
3 @attribute PotenciaRestante { 9.88, 9.87, 9.86, 9.85, 9.84, 9.83, 9.82, 9.81, 9.80, 9.79, 9.78, 9.77, 9.76, 9.75
4 @attribute HoraAtual { 00:00, 00:01, 00:02, 00:03, 00:04, 00:05, 00:06, 00:07, 00:08, 00:09, 00:10, 00:11, 00:12
5 @attribute Consumo { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25
6 @attribute Estado { Ativo, Desativado, StandBy }
7
8 @data
9 9.88, 00:00, 0, Desativado
10 9.88, 01:00, 0, Desativado
11 9.88, 02:00, 0, Desativado
12 9.88, 03:00, 0, Desativado
13 9.88, 04:00, 0, Desativado
14 9.88, 05:00, 0, Desativado
15 9.88, 06:00, 0, Desativado
16 9.88, 06:50, 0, Ativo
17 9.78, 08:15, 0, Ativo
18 9.70, 09:15, 0, StandBy
19 9.62, 10:10, 1, Ativo
20 9.58, 11:15, 1, StandBy
21 9.50, 12:15, 1, StandBy
22 9.38, 13:20, 2, Ativo
23 9.28, 13:30, 2, Ativo
24 8.99, 14:04, 3, Ativo
25 8.89, 14:46, 4, Ativo
26 8.69, 16:14, 5, Ativo
27 8.59, 16:55, 5, Ativo
28 8.29, 18:26, 6, Ativo
29 8.20, 17:10, 6, Ativo
30 8.20, 17:30, 6, StandBy
31 8.20, 18:00, 6, StandBy
32 8.20, 18:30, 6, StandBy
33 8.20, 19:30, 6, StandBy

```

Diante do apresentado, com a verificação de quais aplicabilidades do celular gastam mais bateria, tanto de acordo com as pesquisas já citadas, quanto de acordo com o arranjo experimental, é possível estabelecer os atributos necessários para a construção de uma árvore de decisão e conseqüentemente, gerir de melhor forma a mudança de estado do aplicativo.

De conseguinte, como já foi demonstrado no Capítulo 3, para a construção da árvore de decisão são necessários os cálculos de Entropia e Ganho, os quais são responsáveis, respectivamente, por definir o grau de pureza do conjunto de treinamento e redução na entropia. Observe-se as Figura 4.2, 4.3 que esquematizam os cálculos supracitados:

Figura 4.2: Parte do código responsável pelo cálculo da Entropia.

```

260 /**
261  *
262  * @param ocorrencias
263  * @return
264  */
265 private double calcularEntropia (int[] ocorrencias) {
266     double totalOcorrencias = 0;
267     int count = 0;
268     for (int i = 0; i < ocorrencias.length; i++) {
269         if ((double)ocorrencias[i] == 0)
270             count++;
271         totalOcorrencias += (double) ocorrencias[i];
272     }
273     if (count == ocorrencias.length - 1)
274         return 0;
275     if (totalOcorrencias == 0)
276         return 0;
277     double entropia = 0;
278     for (int i = 0; i < ocorrencias.length; i++) {
279         if (ocorrencias[i] != 0)
280             entropia += (-1) * (ocorrencias[i]/totalOcorrencias) * (Math.Log10(ocorrencias[i]/totalOcorrencias) / Math.Log10(2));
281     }
282     return entropia;
283 }

```

Figura 4.3: Parte do código responsável pelo cálculo do Ganho.

```

230  /**
231  *
232  * @param atributo
233  * @param exemplos
234  * @param atributosAlvo
235  * @param atributos
236  * @return
237  */
238  private double calcularGanho (int atributo, Matrix exemplos, Matrix atributosAlvo, LinkedHashSet<Atributo> atributos) {
239      ArrayList<int[]> valorOcorrencias = gerenciadorTabela.getValorOcorrencias(atributo, exemplos, atributosAlvo);
240      int[] ocorrenciasAlvo = gerenciadorTabela.getOcorrenciasAtributoAlvo(atributosAlvo);
241      double totalOcorrencias = 0;
242      for (int i = 0; i < ocorrenciasAlvo.length; i++) {
243          totalOcorrencias += ocorrenciasAlvo[i];
244      }
245      double[] somaValores = new double[valorOcorrencias.size()];
246      for (int i = 0; i < valorOcorrencias.size(); i++) {
247          for (int j = 0; j < ocorrenciasAlvo.length; j++) {
248              somaValores[i] += valorOcorrencias.get(i)[j];
249          }
250      }
251      double value = 0;
252      for (int i = 0; i < valorOcorrencias.size(); i++) {
253          value += (-1) * (somaValores[i] / totalOcorrencias) * calcularEntropia(valorOcorrencias.get(i));
254      }
255      double ganho = calcularEntropia(ocorrenciasAlvo);
256      ganho += value;
257      return ganho;
258  }

```

De forma a ilustrar a construção da árvore, as Figura 4.4, 4.5 e 4.6 esquematizam as partes do código que têm essa função:

Figura 4.4: Parte do código responsável pela chamada da função de construção da árvore.

```

23  /**
24  * @param caracteristicas
25  * @param rotulos
26  */
27  public void treinar(Matrix caracteristicas, Matrix rotulos) throws Exception {
28
29      this.algoritmo = new ID3();
30      this.raiz = this.algoritmo.buildTree(caracteristicas, rotulos);
31      String nome_arquivo = "arvore_taxa_ganho";
32      this.raiz.convertXML(caracteristicas, rotulos, "saida/"+nome_arquivo+".xml");
33
34  }

```

Figura 4.5: Parte do código responsável pela chamada da função de execução do ID3.

```

61  /**
62  *
63  * @param exemplos
64  * @param atributosAlvo
65  * @return
66  */
67  public No buildTree (Matrix exemplos, Matrix atributosAlvo) {
68      LinkedHashSet<Atributo> atributos = criarTodosAtributos(exemplos);
69      gerenciadorTabela = new GerenciadorTabela(exemplos, atributosAlvo, atributos);
70      No raiz = runID3(exemplos, atributosAlvo, atributos);
71      return raiz;
72  }

```

Figura 4.6: Parte do código responsável pela construção da árvore.

```

106  /**
107   *
108   * @param exemplos
109   * @param atributosAlvo
110   * @param atributos
111   * @return
112   */
113  public No runID3 (Matrix exemplos, Matrix atributosAlvo, LinkedHashSet<Atributo> atributos) {
114      No root = new No(_nodeCounter);
115      _nodeCounter++;
116      if (allExemplosPositivo(atributosAlvo) || atributos.isEmpty()) {
117          double l = atributosAlvo.get(0, 0);
118          String strl = atributosAlvo.attrValue(0, (int) l);
119          Rotulo label = new Rotulo(strl, l);
120          root.setRotulo(label);
121      }
122      else {
123          Atributo A = findMelhorAtributo(exemplos, atributosAlvo, atributos);
124          this.melhorAtributo = A;
125          root.setAtributo(A);
126          for (double value : A.getValores()) {
127              Matrix[] exemplos_vi = gerenciadorTabela.getMatrizPodada(A.getPosicaoColunaID(), (int) value, exemplos, atributosAlvo);
128              if (exemplos_vi[0].rows() == 0) {
129                  No noFolha = new No(_nodeCounter);
130                  _nodeCounter++;
131                  double mcv = atributosAlvo.mostCommonValue(0);
132                  Rotulo label = new Rotulo(atributosAlvo.attrValue(0, (int) mcv), (int) mcv);
133                  noFolha.setRotulo(label);
134                  root.addRamo(value, noFolha);
135              }
136              else {
137                  atributos.remove(A);
138                  root.addRamo(value, runID3(exemplos_vi[0], exemplos_vi[1], atributos));
139                  atributos.add(A);
140              }
141          }
142      }
143      return root;
144  }
145  }

```

Sendo assim, no próximo capítulo, será esquematizado o exemplo de amostragem realizado com fins de demonstrar a implementação do ID3, por meio da coleta de dados efetuada em um ciclo completo de descarga do dispositivo móvel, de modo a identificar o atributo que possui o maior ganho de informação, por meio da utilização das seguintes técnicas: treinamento e cruzamento.

De forma a esclarecer o arranjo experimental em discussão, será feita uma comparação didática dos métodos de treinamento e cruzamento, com o intuito de mostrar qual deles melhor se enquadra na situação problema em pesquisa. Para isso, foram escolhidos os seguintes atributos: potência restante (miliampères), hora atual, consumo (miliampères), e, o estado em que o aplicativo se encontra, através dos quais foram aplicados nas técnicas já mencionadas.

Para a realização da quantificação dos dados coletados, foi utilizado o gráfico *Boxplot*, o qual é um diagrama de caixa que serve no intuito de representar a variação de dados observados em uma variável numérica; o gráfico de Barras, para dimensionar a precisão dos diferentes métodos do algoritmo ID3 e por último o gráfico de curva, para visualizar a convergência de ganho de informação dos diferentes métodos utilizados no algoritmo ID3.

De acordo com o que havia sido verificado ao longo da pesquisa, em relação as características dos aplicativos, percebeu-se que os atributos com maior representatividade ou mesmo relevância foram os seguintes: hora atual, potência restante, consumo e estado do aplicativo.

Como já foi abordado no Capítulo 2, foram listados alguns aplicativos responsáveis pela sobrecarga das baterias, o processamento a ser utilizado para manter esses aplicativos em estado ativo gera um enorme consumo de energia.

Neste aspecto, o *Whatsapp* é um aplicativo popular, presente em quase todos os aparelhos móveis, motivo pelo qual foi escolhido para ser analisado no conjunto de treinamento. Este utilizou dois métodos de avaliação: treinamento e o cruzamento.

5 ARRANJO EXPERIMENTAL

Os dados utilizados foram coletados por meio da avaliação do comportamento do *Whatsapp* no aparelho celular modelo J5, da marca Samsung, com Sistema Operacional Android 6.0, bateria com capacidade de 2.600 mAh de Lí-ion, processador Cortex A53 1.2 Ghz Quad Core, 1.5 GB RAM. A análise em questão foi feita em um período de um ciclo completo de descarga, o qual durou cerca de dois dias (48 horas).

O conjunto de treinamento obtido baseou-se na coleta feita a cada uma hora, baseando-se no modelo de tráfego telefônico de Erlang (ALENCAR, 2012), e ao final de cada utilização do aplicativo. Desse modo, foram obtidas 62 amostras pelas quais foram submetidas aos dois métodos do algoritmo ID3, treinamento e cruzamento. Essas amostras alimentaram um arquivo ARFF, como mostra a Figura 4.1, o qual foi submetido aos métodos listados anteriormente para seu treinamento. O algoritmo foi implementado na linguagem em JAVA.

Inicialmente, para exibição da árvore de decisão construída pelo algoritmo ID3 foi necessária a construção de um arquivo XML, o qual foi criado pelo método `convertXML`, ilustrado pela Figura 5.1. O método em questão percorre a árvore construída pelo ID3 recursivamente e gera o XML com suas conexões.

Figura 5.1: Parte do código responsável pela construção do arquivo XML.

```

144  /**
145  *
146  * @param exemplos
147  * @param atributosAlvo
148  * @param nomeArquivo
149  * @return
150  * @throws IOException
151  */
152  public boolean convertXML(Matrix exemplos, Matrix atributosAlvo, String nomeArquivo) throws IOException {
153      PrintWriter out = new PrintWriter(new File(nomeArquivo));
154      out.println("<?xml version=\"1.0\" encoding=\"UTF-8\"?>");
155      out.println("<graphml xmlns=\"http://graphml.graphdrawing.org/xmlns\">");
156      out.println("<graph edgedefault=\"undirected\">");
157      out.println("<key id=\"name\" for=\"node\" attr.name=\"name\" attr.type=\"string\"/>");
158      this.convertXML(out, exemplos, atributosAlvo);
159      out.println("</graph>");
160      out.println("</graphml>");
161      out.close();
162
163      return !out.checkError();
164  }
165  }
166
167  /**
168  *
169  * @param out
170  * @param exemplos
171  * @param atributosAlvo
172  */
173  private void convertXML(PrintWriter out, Matrix exemplos, Matrix atributosAlvo) {
174      String myLabel = "";
175      if (this.ramos.isEmpty()) {
176          myLabel = this.rotulo.getValueStr();
177      }
178      else {
179          myLabel = this.atributo.getNome();
180      }
181
182      out.println("<node id=\"R"+this.noID+"\"><data key=\"name\"> " + myLabel + "</data></node>");
183
184      if (!this.ramos.isEmpty()) {
185          for (double key : this.ramos.keySet()) {
186              No childNode = this.ramos.get(key);
187              String edgeLabel = "";
188              edgeLabel = exemplos.attrValue(this.atributo.getPosicaoColunaID(), (int)key);
189              int childNodeID = childNode.getNoID();
190              //out.print(" " + this.noID + " -> " + childNodeID);
191              //out.print(" [label=\" " + edgeLabel + "\"]; \n");
192              String id_rotulo_1 = "R"+this.noID;
193              String id_rotulo_2 = "R" + childNodeID;
194              String id_label = "L"+this.noID+""+childNodeID;
195              out.println("<node id=\""+ id_label + "\"><data key=\"name\"> " + edgeLabel + "</data></node>");
196              out.println("<edge source=\""+ id_rotulo_1 + "\" target=\""+ id_label + "\"></edge>");
197              out.println("<edge source=\""+ id_label + "\" target=\""+ id_rotulo_2 + "\"></edge>");
198              // System.out.println(edgeLabel);
199              childNode.convertXML(out, exemplos, atributosAlvo);
200          }
201      }
202  }

```

De forma a dar continuidade, após a construção do arquivo XML, a função demo é responsável pela montagem da árvore graficamente. Esta função necessita de um parâmetro o qual é o caminho do arquivo XML que, foi gerado pela função convertXML da Figura 5.1. Nesse sentido, a Figura 5.2 ilustra a função mencionada:

Figura 5.2: Parte do código responsável pela construção da árvore graficamente.

```

240  /**
241  *
242  * @param datafile
243  * @param label
244  * @return
245  */
246  public static JPanel demo(String datafile, final String label) {
247      Graph g = null;
248      try {
249          g = new GraphMLReader().readGraph(datafile);
250      } catch ( Exception e ) {
251          e.printStackTrace();
252          System.exit(1);
253      }
254      return demo(g, label);
255  }
256
257  /**
258  *
259  * @param g
260  * @param label
261  * @return
262  */
263  public static JPanel demo(Graph g, final String label) {
264      // Criar arvore radial
265      final GraficoArvoreDecisao gview = new GraficoArvoreDecisao(g, label);
266      Visualization vis = gview.getVisualization();
267
268      // Criar search no painel da arvore
269      SearchQueryBinding sq = new SearchQueryBinding(
270          (Table)vis.getGroup(treeNodes), label,
271          (SearchTupleSet)vis.getGroup(Visualization.SEARCH_ITEMS));
272      JSearchPanel search = sq.createSearchPanel();
273      search.setShowResultCount(true);
274      search.setBorder(BorderFactory.createEmptyBorder(5,5,4,0));
275      search.setFont(FontLib.getFont("Tahoma", Font.PLAIN, 11));
276      search.setLabelText("Pesquisar");
277      search.setShowResultCount(false);
278
279      final JFastLabel title = new JFastLabel("                ");
280      title.setPreferredSize(new Dimension(350, 20));
281      title.setVerticalAlignment(SwingConstants.BOTTOM);
282      title.setBorder(BorderFactory.createEmptyBorder(3,0,0,0));
283      title.setFont(FontLib.getFont("Tahoma", Font.PLAIN, 16));
284
285      gview.addControlListener(new ControlAdapter() {
286          public void itemEntered(VisualItem item, MouseEvent e) {
287              if ( item.canGetString(label) )
288                  title.setText(item.getString(label));
289          }
290          public void itemExited(VisualItem item, MouseEvent e) {
291              title.setText(null);
292          }
293      });
294
295      Box box = new Box(BoxLayout.X_AXIS);
296      box.add(Box.createHorizontalStrut(10));
297      box.add(title);
298      box.add(Box.createHorizontalGlue());
299      box.add(search);
300      box.add(Box.createHorizontalStrut(3));
301
302      JPanel panel = new JPanel(new BorderLayout());
303      panel.add(gview, BorderLayout.CENTER);
304      panel.add(box, BorderLayout.SOUTH);
305
306      Color BACKGROUND = Color.WHITE;
307      Color FOREGROUND = Color.DARK_GRAY;
308      UIlib.setColor(panel, BACKGROUND, FOREGROUND);
309
310      return panel;
311  }

```

Sendo assim, a partir da construção da árvore, foi aferida a precisão da mesma pela função de medirPrecisao, a qual é descrita na Figura 5.3:

Figura 5.3: Parte do código responsável pela medição da precisão da árvore.

```

40  /**
41  *
42  * @param caracteristicas
43  * @param rotulos
44  * @param confusao
45  * @return
46  * @throws Exception
47  */
48  public double medirPrecisao(Matrix caracteristicas, Matrix rotulos, Matrix confusao) throws Exception
49  {
50      if(caracteristicas.rows() != rotulos.rows())
51          throw(new Exception("Esperava que os recursos e os rótulos tivessem o mesmo número de linhas.));
52      if(rotulos.cols() != 1)
53          throw(new Exception("Desculpe, este método atualmente apenas suporta rótulos unidimensionais.));
54      if(caracteristicas.rows() == 0)
55          throw(new Exception("Esperava ao menos uma linha.));
56
57      int rotulo_valores = rotulos.valueCount(0);
58      if(rotulo_valores == 0) // Se o rótulo for contínuo...
59      {
60          // O rótulo é contínuo, portanto, mede o erro quadrado médio da raiz
61          double[] pred = new double[1];
62          double sse = 0.0;
63          for(int i = 0; i < caracteristicas.rows(); i++)
64          {
65              double[] feat = caracteristicas.row(i);
66              double[] targ = rotulos.row(i);
67              pred[0] = 0.0; // Certifique-se de que a previsão não é prejudicada por uma previsão anterior
68              prever(feat, pred);
69              double delta = targ[0] - pred[0];
70              sse += (delta * delta);
71          }
72          return Math.sqrt(sse / caracteristicas.rows());
73      }
74      else
75      {
76          // O rótulo é nominal, portanto, mida a precisão preditiva
77          if(confusao != null)
78          {
79              confusao.setSize(rotulo_valores, rotulo_valores);
80              for(int i = 0; i < rotulo_valores; i++)
81                  confusao.setAttrName(i, rotulos.attrValue(0, i));
82          }
83
84          int correctCount = 0;
85          double[] prediction = new double[1];
86          for(int i = 0; i < caracteristicas.rows(); i++)
87          {
88              double[] feat = caracteristicas.row(i);
89              int targ = (int)rotulos.get(i, 0);
90              if(targ >= rotulo_valores)
91                  throw new Exception("Rótulo fora do alcance.));
92              prever(feat, prediction);
93              int pred = (int)prediction[0];
94              //System.out.println("TARG: " + targ + " PRED: " + pred);
95              if(confusao != null)
96                  confusao.set(targ, pred, confusao.get(targ, pred) + 1);
97              if(pred == targ)
98                  correctCount++;
99          }
100          return (double) correctCount / caracteristicas.rows();
101      }
102  }

```

Para finalizar o procedimento experimental, foi realizado a coleta das métricas por intermédio da classe EscreverArquivo, cuja qual cria um arquivo TXT e o alimenta chamando

a função escrever passando como parâmetro a linha a qual deseja gravar no arquivo, conforme ilustra a Figura 5.4:

Figura 5.4: Parte do código responsável pela coleta das métricas.

```

5 /**
6 *
7 * @author Andre Marques
8 *
9 */
10 public class EscreverArquivo {
11
12     private static EscreverArquivo escreverArquivo;
13
14     /**
15     *
16     */
17     private EscreverArquivo() {
18     }
19
20     /**
21     *
22     * @return
23     */
24     public static synchronized EscreverArquivo getInstance() {
25         if (escreverArquivo == null)
26             escreverArquivo = new EscreverArquivo();
27
28         return escreverArquivo;
29     }
30
31     /**
32     *
33     * @param texto
34     */
35     public void escrever(String texto) {
36         try {
37             // O parametro é que indica se deve sobrescrever ou continua no
38             // arquivo.
39             FileWriter fw = new FileWriter("saida/estatistica_arquivo.txt", true);
40             BufferedWriter conexao = new BufferedWriter(fw);
41             conexao.write(texto);
42             conexao.newLine();
43             conexao.close();
44         } catch (Exception e) {
45             e.printStackTrace();
46         }
47     }
48 }

```

Por fim, a partir do treinamento, foi realizada uma comparação entre os dois métodos citados em termos de tempo de execução e precisão que o algoritmo obteve. Assim, foram realizadas 30 vezes de 1000 execuções de cada método, sendo que em particular no método de cruzamento foram feitos cruzamentos entre 2, 3, 4, 5 e 6 pares de atributos.

A configuração da máquina utilizada para realização dos testes no algoritmo ID3 foram as seguinte: Intel(R) Core(TM) i5-4200U CPU @1.60GHz, NVIDIA GeForce 820M, 8 GB RAM DDR3 1600Mhz, Sistema Operacional Microsoft Windows 10 Professional 64-bit (Build 15063) utilizando o Java Versão 8 Atualização 151 (Build 1.8.0_151-b12).

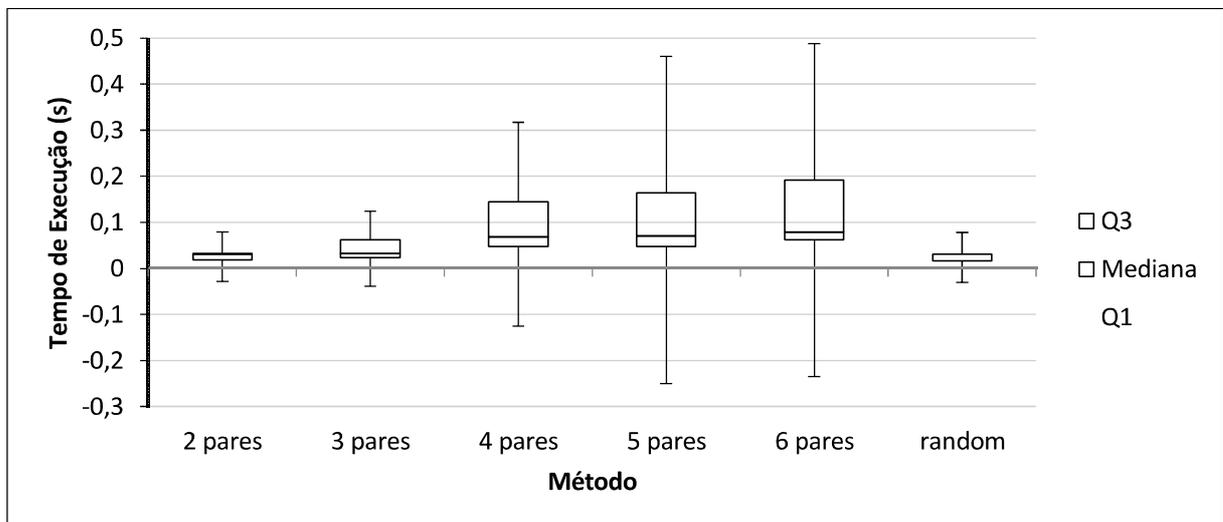
6 RESULTADOS

Os principais resultados alcançados na presente pesquisa estão descritos nas Figuras 6.1, 6.2, 6.3, 6.4, 6.5, cujas quais abordaremos a seguir:

Na Figura 6.1 é possível destacar a variância de tempo conforme o método de treinamento utilizado. De acordo com o verificado, o treinamento com cruzamento de 2 pares de atributos necessita de uma quantidade de tempo equivalente ao treinamento *random*, no entanto, sua variabilidade é mais eficaz ao produzir a árvore, pois, utiliza uma menor variabilidade temporal.

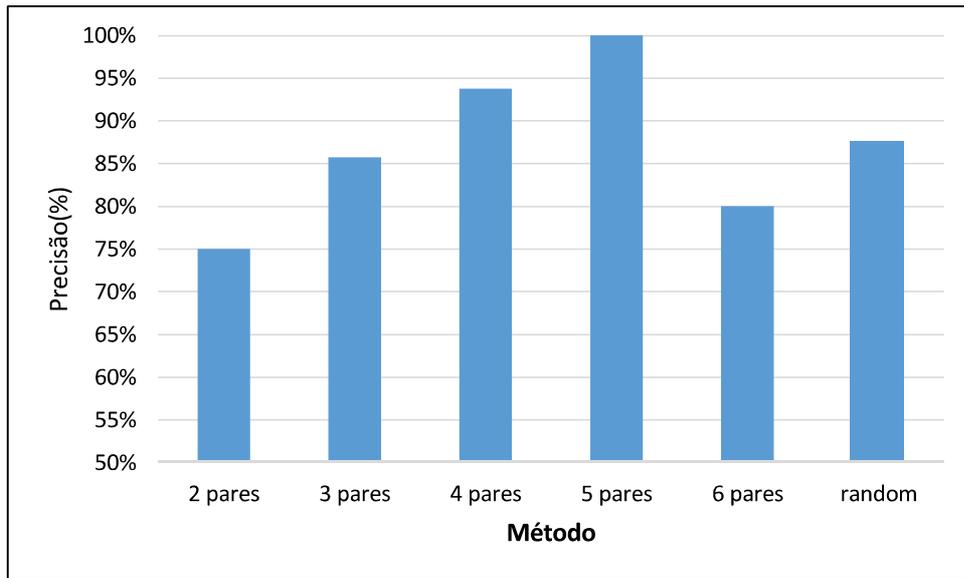
Em continuidade, verifica-se que o treinamento com 3 pares de atributos produz a árvore com maior tempo sendo que sua variabilidade de tempo se aproxima bastante de sua mediana. Por último, é possível perceber que os demais produzem uma variabilidade temporal muito grande, o que nos faz concluir que estes devem ser evitados por motivo do fator de tempo de treinamento. Observe-se a Figura 6.1:

Figura 6.1: Variação do tempo de acordo com o método utilizado.



De forma a dar prosseguimento a análise dos resultados, ilustra-se na Figura 6.2 que, o gráfico de barras utilizado mostra a mediana da precisão conforme os treinamentos realizados pelo algoritmo ID3. Neste aspecto, o gráfico em questão indica que o método 5 pares de atributos produziu a melhor precisão dentre os demais métodos utilizados. Segue-se a Figura 6.2:

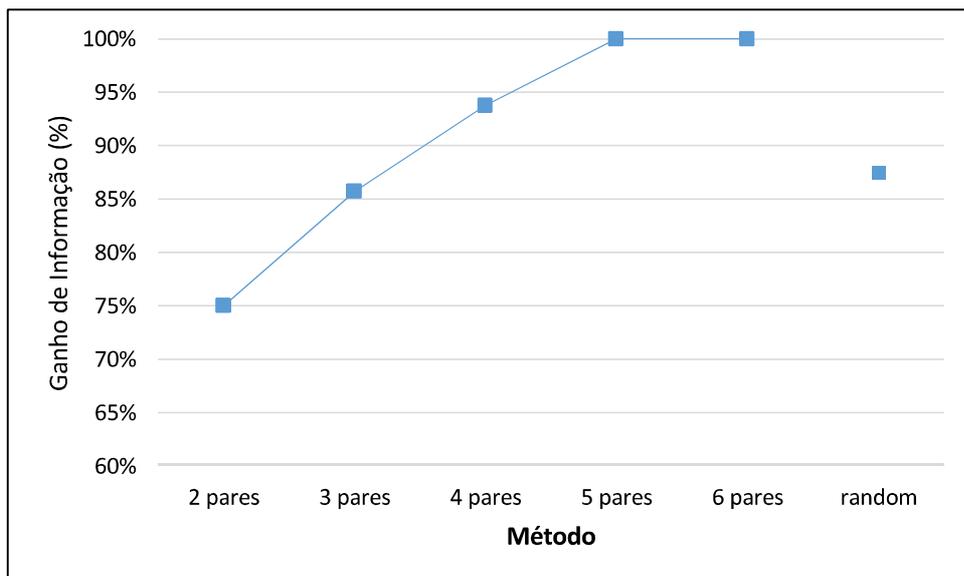
Figura 6.2: Valores de mediana da precisão de acordo com o método utilizado.



Ao dar continuidade, é possível perceber que o gráfico trazido pela Figura 6.3 mostra a convergência que o algoritmo ID3 utiliza para encontrar árvores com raízes que contenham maior ganho de informação nos nós das raízes das árvores, na medida que aumenta a quantidade de cruzamentos.

Nesse sentido, entre os métodos de 5 e 6 pares de atributos não houve ganho significativos de informação, assim como, no que diz respeito ao método *random*, este equiparou-se ao método de 3 pares. Analise-se a Figura 6.3:

Figura 6.3: Convergência de ganho de informação de acordo com o método utilizado.



De conseguinte, em atendimento aos objetivos específicos do presente, ressalta-se ainda o funcionamento da abordagem em AM, por meio das suas técnicas, ficando evidenciado na Figura 6.4, de forma gráfica, por intermédio do *framework Prefuse*, a árvore construída a partir do algoritmo ID3. Percebe-se que, a árvore em questão teve o atributo “HoraAtual” com maior ganho de informação. Logo após, a Figura 6.5 retrata o atributo “PotenciaRestante” como sendo o que possui maior ganho de informação. Seguem-se as Figuras 6.4 e 6.5:

Figura 6.4: Árvore gerada com maior ganho de informação no atributo “HoraAtual”.

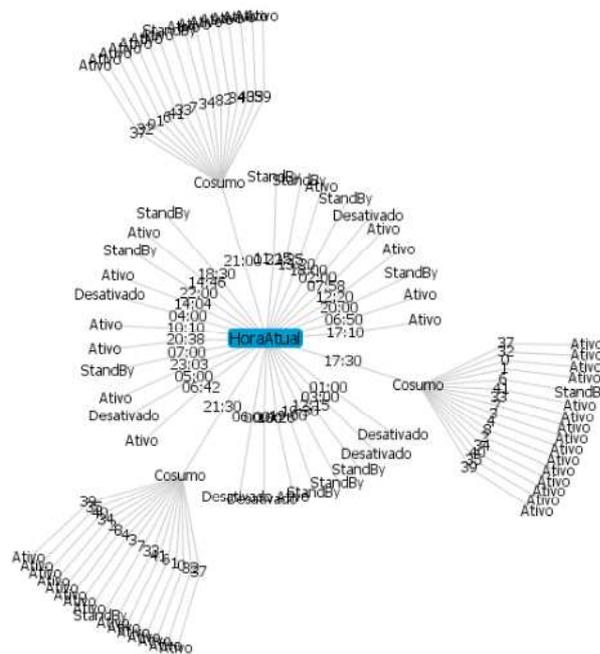
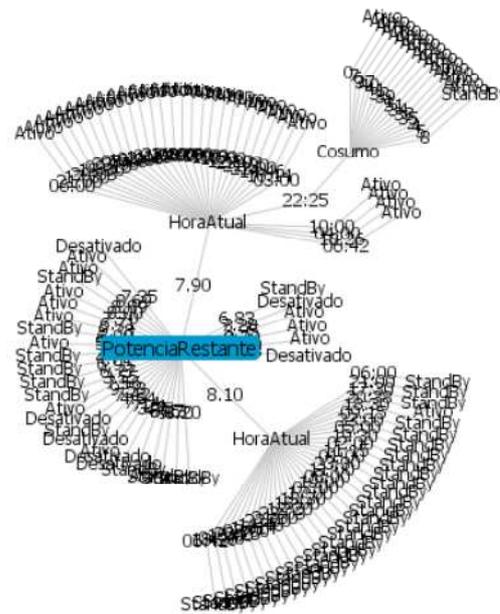


Figura 6.5: Árvore gerada com maior ganho de informação no atributo “PotenciaRestante”.



Portanto, a partir da análise do objetivo geral, o qual era a construção de uma Revisão bibliográfica e de um arranjo experimental relacionado ao estudo da AM, e também pelos resultados apresentados, é possível perceber que o método de 4 pares de atributos foi o que melhor se apresentou na resolução da problemática da pesquisa, pois, demonstrou uma variância de tempo satisfatória em comparação aos métodos 5 e 6 pares. E além disso o método de 4 pares possui uma mediana de precisão acima de 90%, conforme Figura 6.2. Sendo assim, é o método 4 pares que deve ser levado em consideração para resolução da problemática tendo em vista que os *smartphones* possuem recursos escassos e que o principal objetivo da presente pesquisa é economia de bateria. Visto isso, foi alcançado o objetivado da presente pesquisa.

7 CONSIDERAÇÕES FINAIS

Por meio da pesquisa realizada, foi possível diagnosticar a necessidade pela real otimização do consumo das baterias dos dispositivos móveis, os quais são amplamente utilizados pela sociedade moderna, fazendo parte do cotidiano das pessoas e consequentemente das atividades que estas exercitam.

Neste aspecto, a inteligência artificial tem-se mostrado uma importante ferramenta na descoberta de meios a serem utilizados no gerenciamento da energia, ou seja, apesar das dificuldades em termos de pesquisa, mostra-se que esta área tão inovadora das ciências da computação venha a alçar voos ainda maiores nas descobertas voltadas a economia dos recursos energéticos, consequentemente, gerando benefícios não só para os consumidores desta tecnologia, mas também ao próprio meio ambiente.

No que diz respeito ao manejo das árvores de decisão como método organizativo da estrutura de atributos a serem avaliados no estudo do gerenciamento de energia dos dispositivos, foi possível perceber que por meio de uma técnica simples de classificação de dados mais relevantes, gera-se o conhecimento necessário para que a máquina possa definir as tarefas a serem feitas e consequentemente prever as suas ações.

De conseguinte, sabe-se que a utilização dos algoritmos vem a ser o meio a quantificar os dados e o seu valor de relevância para análise a ser feita, é a partir destes cálculos que são definidos os melhores atributos a serem identificados pela máquina e como estes reagem a cada experimentação de aplicabilidades do suposto dispositivo móvel a utilizar de um programa a ser fabricado nestes moldes.

Em contrapartida, é de se registrar que os outros métodos avaliados ao decorrer desta pesquisa também geram relevância de análise, pois, o método de *the timeout policy*, por exemplo, é mais simples em termos de ações, pois este não corporifica as várias aplicabilidades do dispositivo, apenas opera em atividades simples, como a permanência do dispositivo em estado ligado caso a duração de um dado intervalo inativo for menor que o tempo limite, ultrapassado este tempo limite, o aparelho deve ficar em estado desligado.

Dessa forma, como já foi relatado em momento oportuno, o presente estudo não teve o objetivo de concretizar teses ou mesmo criar uma nova, mas sim de fazer uma comparação entre alguns métodos, dando maior ênfase a análise das árvores de decisão e da pesquisa dos algoritmos a serem aproveitados.

7.1 TRABALHOS FUTUROS

Como possíveis trabalhos futuros, pode-se apontar:

- Criação de aplicação que execute como serviço no *android*, automatizando a coleta de dados para construção da árvore de decisão.
- Criação de uma aplicação *android* e/ou *desktop* baseada no ID3, cujo qual teria o papel de identificar pessoas com risco de diabetes, cujo sistema seria alimentado pelos prontuários médicos de atendimento.
- Publicação de artigo científicos com os dados coletados e resultados alcançados nessa pesquisa.

REFERÊNCIAS

ALENCAR, Marcelo S. **Telefonia Celular Digital**. 2a ed., São Paulo: Érica, 2012.

ANATEL. **Brasil registra 241 milhões de linhas móveis em operação em setembro de 2017**. 2017. In: Administração Nacional de Telecomunicações. Disponível em: <<http://www.anatel.gov.br/dados/component/content/article?id=283>>. Acesso em: 23 nov. 2017.

BAGULEY, Richard. **Entenda como funcionam as baterias de seus dispositivos móveis**. 2013. In: Computer world. Disponível em: <<http://computerworld.com.br/tecnologia/2013/03/22/entenda-como-funcionam-as-baterias-de-seus-dispositivos-moveis>>. Acesso em: 21 jan. 2017.

BARANAUSKAS, José Augusto. **Aprendizado de Máquina Conceitos e Definições**. S/D. Disponível em: <<http://dcm.ffclrp.usp.br/~augusto/teaching.htm>>. Acesso em: jan. 2017.

DAQUINO, Fernando. **10 apps para aumentar a duração de bateria do seu Android**. 2014. In: TECMUNDO. Disponível em: <<https://www.tecmundo.com.br/android/54202-10-apps-para-aumentar-a-duracao-de-bateria-do-seu-android.htm>>. Acesso em: jan. 2017.

GARCIA, Simone C.; et.al. **Árvores de decisão - algoritmos ID3 e C4.5**. 2000. In: Revista Cadernos de Informática. Disponível em: <<http://seer.ufrgs.br/index.php/cadernosdeinformatica/article/view/v1n1p52-55/8809>>. Acesso em: jan. 2017.

GOMES, Helton Simões. **Tablet ultrapassa vendas de desktop e notebook pela 1ª vez no Brasil**. 2014. In: Globo.com (G1). Disponível em: <<http://g1.globo.com/tecnologia/noticia/2014/03/tablet-ultrapassa-vendas-de-desktop-e-notebook-e-pela-1-vez-no-brasil.html>>. Acesso em: jan. 2017.

JORDÃO, Fábio. **História: a evolução do celular**. 2009, In: Tecmundo. Disponível: <<https://www.tecmundo.com.br/celular/2140-historia-a-evolucao-do-celular.htm>>. Acesso em jan. 2017.

LEAL, Israel Aires Costa. **Aplicação de Algoritmos Bio-inspirados para Otimização de Bloqueio de Chamadas em Sistema Móvel Celular**. 2015. Dissertação (mestrado em Telemática). Programa de Pós-graduação em engenharia de sistemas. Universidade de Pernambuco.

LORENA, Ana Carolina; et.al. **Uma Introdução às Support Vector Machines**. 2007. In: Revista de Informática Teórica e Aplicada (RITA). Disponível em: <http://www.seer.ufrgs.br/index.php/rita/article/view/rita_v14_n2_p43-67/3543>. Acesso em: jan. 2017.

LUIZ Saulo O.D.; et.al. **Feasibility of a timeout power management policy**. 2013. IEEE Third International Conference on Consumer Electronics-Berlin (ICCE-Berlin).

OGURI, Pedro. **Aprendizado de máquina para o problema Sentiment Classification**. 2006, In: PUC – Rio. Disponível em: <https://www.maxwell.vrac.puc-rio.br/9947/9947_1.PDF>. Acesso em jan. 2017.

MENDES, José Renato Ribeiro Mendes. **5G: A quinta Geração**. 2014. Disponível em:<<http://repositorio.roca.utfpr.edu.br/jspui/handle/1/3866>>. Acesso em Jan. 2017.

MICROSOFT AZURE. **Como Escolher algoritmos de Aprendizagem de Máquina do Microsoft Azure**. 2016. Disponível em: <<https://docs.microsoft.com/pt-br/azure/machine-learning/machine-learning-algorithm-choice>>. Acesso em: jan. 2017.

NGUYEN, Tien-Dung; et. al. **Prediction-based energy policy for mobile virtual desktop infrastructure in a cloud environment**. 2015. ELSEVIER. Information Sciences. Departamento f computer. Engineering Kyung Hee University, Republico f Korea.

OLIVEIRA, Stanley R. M. **Introdução à Aprendizagem de Máquina**. S/D. In:UNICAMP. Disponível em:<<http://www.ime.unicamp.br/~wanderson/Aulas/Aula9/MT803-AprendizadoMaquina-ArvoreDecisao.pdf>>. Acesso em: jan. 2017.

PALEOLOGO. G. A. Policy optimization for dynamic power management. S/D, In: IEEE xplore Digital Library. Disponível em:<<http://ieeexplore.ieee.org/document/724463>>. Acesso em jan. 2017.

PINTO, Tasso Tirapani Silva apud VEJA. **Evolução das memórias, baterias e processadores para dispositivos móveis**. 2012. In: Grenoble. Disponível em: <http://grenoble.ime.usp.br/~gold/cursos/2012/movel/mono-1st/2006-2_TassoTirapani.pdf>. Acesso em: jan. 2017.

PISTORI, H.; SOUZA, K.P. de. **Tecnologia Adaptativa à Biotecnologia: Estudo de caso e oportunidades**. 2010. In: Quarto Workshop de Tecnologia Adaptativa. Disponível em: <<http://lta.poli.usp.br/lta/publicacoes/artigos/2010/pistori-e-souza-2010-tecnologia-adaptativa-aplicada-a-biotecnologia-estudos-de-caso-e-oportunidades/view+>>. Acesso em: jan. 2017.

SEROWY, Stephan. **Estes são os aplicativos que mais consomem bateria e o plano de dados do seu celular. 2015**, In: Androidpit. Disponível em:<<https://www.androidpit.com.br/aplicativos-que-mais-consoem-bateria-dados>>. Acesso em jan. 2017.

VALINNA-RODRIGUEZ, Narseo; et. al. **Exhausting Battery Statistics: understanding the energy demands on mobile handsets**. NewDelhi,India. 2010. ACM- DL- Digital library. New York, NY, USA ©2010. ISBN: 978-1-4503-0197-8 doi>[10.1145/1851322.1851327](https://doi.org/10.1145/1851322.1851327)

VEIGA, Daniel da. **Bateria: Uma investigação detalhada (Android)**. 2013. In: Showmetech. Disponível em: <<https://www.showmetech.com.br/investigacao-a-bateria/>>. Acesso em: jan. 2017