

Prediction of human activity in a smart home using Machine Learning on data collected by IoT sensors

Leonardo M. R. Santos¹

¹Instituto de Computação – Universidade Federal Fluminense (UFF)
Niterói – RJ – Brazil

Abstract. *This work addresses the training and comparison of several machine-learning algorithms to attempt to predict what activity is being carried out at one specific moment by a smart home resident. More specifically, it deals with the problem of human activity recognition.*

Resumo. *Este trabalho versa sobre o treinamento e a comparação de diferentes algoritmos de Aprendizado de Máquina, de forma a tentar prever qual atividade está sendo realizada em determinado momento por um morador de uma casa inteligente. Insere-se, portanto, no domínio dos problemas de reconhecimento de atividades humanas.*

1. Introdução

O problema de reconhecimento de atividades humanas – *Human Activity Recognition* (HAR), em inglês – constitui um dos tópicos de pesquisa mais atuais devido à maior disponibilidade de sensores a baixo custo hodiernamente, com menos consumo de energia, bem como aos avanços em Aprendizado de Máquina e em Internet das Coisas (JOBANPUTRA; BAVISHI; DOSHI, 2019).

Diversas aplicações podem se beneficiar do uso de HAR. Na área da saúde, tem especial importância no monitoramento de grupos de risco. Em segurança e vigilância, destaca-se o avanço na detecção e prevenção de crimes. Inúmeras outras aplicações podem igualmente ser favorecidas, como o setor de entretenimento.

As casas inteligentes, por sua vez, são uma tendência em grande crescimento no âmbito da Internet das Coisas (IoT), com vistas a ampliar a automação de tarefas quotidianas, introduzindo outras aplicações interessantes, com base no uso de sensores e atuadores instalados no ambiente residencial.

Este trabalho integra o conjunto de requisitos práticos demandados na disciplina de Aprendizado de Máquina 2020.1 do IC/UFF. O interesse do aluno/autor ao cursar a disciplina decorre do objetivo de se ter um contato inicial com um assunto de grande relevância atualmente, de forma a investigar possibilidades de integração de Aprendizado com sua área de pesquisa, Internet das Coisas.

Buscando conjugar ambas as áreas, foi selecionado um conjunto de dados referente ao aprendizado sobre dados coletados por sensores no âmbito de residências inteligentes.

2. Referencial Teórico e Revisão da Literatura

Segundo (YANG; LEE; CHOI, 2011), o reconhecimento de atividades pode ser definido como a capacidade de reconhecer ou detectar a atividade atualmente em curso com base nas informações recebidas de diferentes sensores. Esses sensores podem ser câmeras, sensores vestíveis (*wearables*), sensores conectados a objetos do cotidiano ou implantados no ambiente.

Paralelamente, as casas inteligentes tem avançando consideravelmente como uma tendência de grande impacto na literatura. Nos últimos anos, constata-se um notável progresso na aplicação de casas inteligentes em vários campos: automação residencial (TOSCHI; CAMPOS; CUGNASCA, 2017), otimização do uso de energia (SERRA *et al.*, 2014), segurança doméstica (DAHMEN *et al.*, 2017) e, dentre inúmeros outros, reconhecimento de atividade humana (BAKAR *et al.*, 2015).

O conjunto de dados a ser estudado neste trabalho leva a denominação de CASAS, conforme seus autores (COOK, *et al.*, 2013), encontrando-se disponível no repositório da UCI para aprendizado de máquina (COOK; CRANDALL; THOMAS, 2019), e está inserido no contexto das pesquisas realizadas sobre dados reais para o reconhecimento de atividade humana em casas inteligentes. O conjunto consolida uma ampla base de dados, obtida pelos pesquisadores de Washington supramencionados mediante sensoramento contínuo de dados ambientais, em residências de voluntários, enquanto estes estavam a realizar normalmente suas atividades rotineiras. Foram distribuídos diferentes tipos de sensores, como detectores de movimento, abertura e fechamento de porta, sensores de temperatura e de acionamento de interruptores de luz, espalhados por toda a casa dos voluntários, de forma a investigar atividades específicas do cotidiano. Um aspecto interessante é que o conjunto não inclui câmeras, de forma que o reconhecimento de atividades não se baseia no aprendizado sobre imagens e vídeos.

O CASAS pode fornecer uma boa base de comparação entre diferentes algoritmos de aprendizado de máquina. Analogamente, existem outros conjuntos de dados reais disponíveis ao público e coletados por sensores em residências inteligentes, como os abordados em (KASTEREN; ENGLEBIENNE; KRÖSE, 2011), (ORDÓÑEZ; TOLEDO; SANCHIS, 2013) e (ALEMDAR *et al.*, 2013). Alguns desses conjuntos restringem-se a residências com apenas um inquilino. O conjunto de dados CASAS refere-se a residências que podem ter múltiplos moradores.

Quanto aos artigos publicados com base no CASAS, a equipe de (COOK, *et al.*, 2013) buscou projetar uma máquina de vetores de suporte (SVM) para realizar, em tempo real, o reconhecimento de atividades. Em seguida, realizaram testes com outros modelos de aprendizado de máquina, como o Naive Bayes, modelos Markov ocultos e Campos Aleatórios Condicionais. Segundo os autores, as SVMs alcançaram um desempenho consideravelmente melhor do que as outras abordagens, além de quantificar o grau de ajuste entre os dados e um rótulo de atividade, facilitando recursos adicionais como a detecção de anomalias. A acurácia média ponderada de todas as classes foi de **84%**, havendo robustez dos modelos mesmo quando usados em configurações domésticas distintas.

Por fim, ressalta-se o crescente uso de algoritmos de *deep learning* no reconhecimento de atividade humana, como os baseados em CNN – *Convolutional Neural Networks*. O estudo realizado por (SINGH *et al.*, 2017) compara o CNN com

outros algoritmos de aprendizado de máquina, realizando experimentos com diversos conjuntos de dados reais disponíveis ao público, tendo constatado bom desempenho com CNN. De todo modo, algoritmos de *deep learning* fogem ao escopo deste Relatório por não terem sido abordados na disciplina.

3. Metodologia da Análise Experimental

Com vistas a exercitar os conhecimentos adquiridos na disciplina Aprendizado de Máquina 2020.1, IC/UFF, foi utilizado um conjunto de dados reais, disponíveis no repositório da UCI para aprendizado de máquina (COOK; CRANDALL; THOMAS, 2019).

O conjunto de dados original contava com 31 voluntários, 13,9 milhões de exemplos e cerca de 64 GB de dados. Por questões de escala e de viabilidade prática, decidiu-se focar nos dados do voluntário csh101, reduzindo a quantidade de exemplos totais para cerca de 321 mil, considerado ainda um número razoável e suficiente para o propósito deste trabalho.

A tarefa de classificação pretendida consiste em prever a atividade que está ocorrendo em uma casa inteligente observada pelos sensores do ambiente.

O conjunto possui um total de 37 atributos, incluindo o de saída, sendo que os atributos de entrada caracterizam dados pré-processados após captura crua dos diversos sensores, e o atributo de saída representa o rótulo das atividades realizadas, os quais foram inseridos manualmente pelo voluntário para efeitos de treinamento e teste dos algoritmos de aprendizado.

Para exercitar os conhecimentos desenvolvidos na disciplina, buscou-se aplicar ao menos um algoritmo de cada técnica de aprendizado de máquina abordada em aula, a saber, as árvores de decisão, regras de decisão, k-NN, métodos bayesianos, redes neurais e máquinas de vetores de suporte.

O treinamento e o teste sobre o conjunto de dados foram realizados no WEKA, software gratuito e de código aberto mantido pela Universidade de Waikato, Nova Zelândia. A máquina utilizada para execução consistiu em um processador Intel Core i5 8250U, com 8.00 GB de memória RAM e sistema Windows 10 Home.

Como forma de particionamento, foi usado inicialmente o método *holdout*, dividindo o conjunto total em 2/3 dos dados para treinamento e o 1/3 restante para teste. Foi testada também a validação cruzada mediante o método *k-fold*, que naturalmente elevou o tempo total de execução dos algoritmos, por sua vez trazendo uma acurácia mais confiável.

Haja vista que alguns dos algoritmos estouraram a memória disponível durante a execução, partiu-se para uma nova estratégia, utilizando aprendizado em fluxo (*stream learning*) por meio do framework MOA, também mantido pela Universidade de Waikato.

Foi criado um endereço para um diretório no GitHub contendo a localização do conjunto de dados original, o efetivo conjunto utilizado nos experimentos e um resumo da execução com as respectivas imagens das telas. Disponível em: <<https://github.com/lmrsantosUFF/ML>>.

4. Resultados Obtidos

Logo de início, ao analisar as propriedades estatísticas do conjunto de dados, verificou-se um grande desbalanceamento no atributo alvo, denominado *activity*. Enquanto há uma classe minoritária *Work_At_Table* apresentando apenas 184 exemplos, a classe majoritária *Other_Activity* dispõe de 90824 instâncias. A distribuição em todas as classes é apresentada no relatório complementar disponível no GitHub.

Além disso, chamou atenção o fato de a classe *default* ser justamente aquela de menor interesse para o reconhecimento de atividades, uma vez que *Other_Activity* representa qualquer outra atividade que não pôde ser enquadrada em nenhuma daquelas mais rotineiras rotuladas no estudo. Presume-se que, eventualmente, um número excessivo de exemplos nessa classe em determinada distribuição poderia inclusive inviabilizar o estudo pretendido, uma vez que um algoritmo que predissesse apenas a classe majoritária apresentaria uma falsa conclusão de elevada acurácia, sem sequer ter reconhecido a atividade realizada, já que essa estaria sendo classificada genericamente como uma atividade qualquer (*Other_Activity*).

Segundo (COOK, *et al*, 2013), esse é um problema que ocorre frequentemente na área de reconhecimento de atividades, sobretudo quando o reconhecimento ocorre em tempo real, pois em algum momento podem surgir dados que não pertencem a nenhuma das classes de atividades catalogadas. A segmentação de dados não rotulados em classes menores poderia melhorar o desempenho do reconhecimento de atividades, à medida que a classe *Other_Activity* deixasse de ser a dominante em termos de tamanho. De qualquer forma, a frequência da classe *Other_Activity* no conjunto de dados CASAS foi de 28%, o que mitiga relativamente o problema, já que será buscada uma acurácia muito superior a esse valor.

O primeiro algoritmo utilizado no experimento foi do tipo **k-NN**, com o valor de *k* igual a 1, ou seja, baseando a classificação somente no vizinho mais próximo. Para isso, foi selecionado o algoritmo *IBk*, disponível no WEKA. O **k-NN** caracteriza um algoritmo preguiçoso, em que a computação é adiada até a fase de classificação. Assim, como era de se esperar, o tempo decorrido para construir o modelo no experimento foi praticamente nulo (0.1s de acordo com o WEKA), com o treinamento consistindo em apenas carregar os exemplos na memória.

A acurácia obtida com o 1-NN foi de 94,20%, a qual pode ser considerada elevada. Já o tempo gasto para testar o modelo foi de 2450,91 segundos, ou seja, em torno de 41 minutos. Esse tempo de teste refletiu o custo crescente em que o **k-NN** incorre à medida que aumentam os exemplos, pois o algoritmo necessita comparar a instância que se está tentando prever com todos os demais exemplos. Ficou nítida a desvantagem no **k-NN** por não ter construído um modelo para posterior predição rápida. Além disso, o conjunto CASAS possui 37 atributos, e à proporção que se aumenta o número de atributos presentes no conjunto de dados, o espaço dimensional do problema cresce exponencialmente, já que o 1-NN se baseia no cálculo da distância euclidiana entre os objetos, e gera superfícies de decisão bastante complexas, as quais, segundo o livro adotado como referência na disciplina (FACELI *et al.*, 2011), acabam formando um conjunto de poliedros que se denomina diagrama de Voronoi.

Assim, cogitou-se avaliar a possibilidade de redução de dimensionalidade no **k-NN** para medir experimentalmente os efeitos dessa medida no tempo de teste, e mensurar

o impacto na acurácia resultante. Uma hipótese automática de redução de dimensionalidade surgiria caso outros algoritmos, em especial as árvores de decisão, produzissem modelos acurados sem utilizar todos os 37 atributos presentes no CASAS.

Prosseguindo o experimento com um algoritmo de **regra de decisão**, foi utilizado o OneR presente no WEKA, o qual consiste em um método extremamente simples, que escolhe uma única variável para predição como sendo aquela que permitir o maior número de acertos no subconjunto de treinamento (no caso em tela, foi selecionado o atributo lastMotionLocation). O algoritmo parte do pressuposto de que alguns problemas podem ser relativamente simples, sendo potencialmente reduzidos por aproximação sem grandes perdas, com a vantagem de diminuir consideravelmente o tempo de geração do modelo. De fato, o tempo decorrido para construir o modelo e testá-lo no CASAS foi de menos de 4s ao todo.

Não obstante, a acurácia obtida com o OneR foi de apenas 46,50%, o que sugere que o reconhecimento de atividades no CASAS representa um problema mais complexo, sendo inviável reduzi-lo a uma única regra de decisão.

Assim, partiu-se para a abordagem das **árvores de decisão**, tendo sido empregado o algoritmo J48 no WEKA. O algoritmo teve uma resposta rápida, construindo o modelo em apenas 126 segundos, e com uma etapa de teste imediata – característica das árvores de decisão –, a qual durou somente 1,29 segundo. Ao mesmo tempo, a acurácia obtida por essa árvore foi de 94,83%, consolidando uma excelente relação de custo benefício de tal algoritmo para o problema CASAS.

Por outro lado, teve de ser considerado igualmente o tamanho da árvore, que resultou em um total de 16973 nós, o que faz alusão novamente a uma noção de complexidade do problema proposto.

O resultado do J48 traz mais um aspecto interessante. Apesar do grande número de nós em comparação com a quantidade de atributos, investigou-se se a árvore teria baseado sua decisão levando em conta todos os atributos disponíveis no conjunto. O que se constatou é que foram utilizados apenas 26 de um total de 36 atributos de entrada, tendo obtido elevada acurácia, surgindo uma hipótese de redundância, ou ao menos de acoplamento, nos sensores acionados concomitantemente em cada atividade realizada pelo voluntário.

Nessa acepção, foi gerado um novo conjunto de dados visando à redução de dimensionalidade, mediante a eliminação dos atributos não utilizados pelo J48 para formar a árvore de decisão, e então proceder com uma nova execução do 1-NN com vistas a medir seu novo tempo de execução e se a nova acurácia ficaria degradada com essa perda de informação.

A nova execução do IBk com o conjunto reduzido, sob os mesmos parâmetros, consumiu 2294 segundos para testar o modelo, ou seja, tendo respondido 6% mais rápido que o teste no conjunto integral. Por sua vez, a acurácia foi de 94,38%, praticamente coincidente com a do conjunto integral (94,20%), se for considerada alguma margem de erro intrínseca à forma como são particionados os dados em determinada distribuição. Por conseguinte, avalia-se que a redução de dimensionalidade logrou sucesso, tendo removido dados que se mostraram pouco relevantes para a classificação, ao menos no que concerne a esta distribuição específica. Não obstante, para efeitos de uma comparação mais justa com os

algoritmos anteriores, este trabalho continuou executando os métodos seguintes sobre o conjunto integral de atributos.

Quanto aos **algoritmos bayesianos**, partiu-se inicialmente para a experimentação com o NaiveBayes. Trata-se de um classificador probabilístico simples, baseado na aplicação do teorema de Bayes com um pressuposto muitas vezes ingênuo de que haveria total independência entre os atributos do conjunto de dados. O tempo somado de treinamento e teste foi bastante rápido, girando em torno de 20s. Porém, o algoritmo obteve uma acurácia precária de 38,62%, pouco acima do simples chute na classe majoritária (28,26%), demonstrando que a ingenuidade do algoritmo é inadequada para tratar do conjunto CASAS. Alternativamente, foram investigadas as redes bayesianas, usando o BayesNet no WEKA. Esse algoritmo forma uma rede de crenças representada por um grafo acíclico dirigido (DAG), sendo que cada aresta indica uma correlação causal entre os nós de atributos. Com um tempo de treino e teste em torno de 20s, a acurácia melhorou significativamente em comparação com o anterior método probabilístico mais ingênuo. Ainda assim, foram corretamente classificados apenas 69,03% dos exemplos, muito aquém do resultado da árvore de decisão.

Uma abordagem diferente consiste em usar métodos baseados em otimização, como as redes neurais e as máquinas de vetores de suporte.

Para construir uma **rede neural**, foi usado o MultilayerPerceptron, o qual realiza transformações para que os dados se tornem linearmente separáveis em hiperplanos, e usando uma função degrau para ativação dos neurônios, e foi configurada uma taxa de aprendizado de 0,3 no experimento. Foram necessários 5274,21 segundos para construir o modelo, cerca de uma hora e meia. Ainda assim, o perceptron atingiu uma acurácia de 76,88%, inferior à obtida pelo J48.

Por fim, como **máquina de vetores de suporte**, foi utilizado o algoritmo SMO no WEKA. Após mais de duas horas e meia treinando o modelo, o heap de memória estourou.

Para lidar com essa dificuldade, foi adotada uma nova estratégia, baseando o aprendizado em um fluxo contínuo de dados, característica do *stream learning*. Segundo o livro do MOA (BIFET; KIRKBY, 2009), esse framework, desenvolvido pela Universidade de Waikato, foca nos problemas da classificação, sendo especialmente adequado quando o número de exemplos é muito grande, ao mesmo tempo em que há um número limitado de classes (geralmente inferior a 10) e, sobretudo, quando a quantidade de memória disponível para um algoritmo de aprendizado é consideravelmente menor que o tamanho dos dados de treinamento.

Inicialmente, foi usado o classificador SMO do WEKA no MOA, para concluir a lista pretendida de algoritmos para a experimentação. Contudo, a acurácia média obtida no fluxo de dados foi de 26,38%. Ou seja, o SMO nesse fluxo errou mais do que se tivesse simplesmente chutado na classe majoritária. Uma hipótese para explicar esse resultado seria que os algoritmos do WEKA estão mais voltados para o aprendizado tradicional em batch, não se adequando ao aprendizado em fluxo.

Da mesma forma, o uso do J48 do WEKA em fluxo no MOA apresentou uma acurácia média de 25,12%, ao passo que sua acurácia executando-o em *batch* foi uma das melhores obtidas no experimento, como já apresentado.

Utilizando então algoritmos no MOA desenvolvidos especificamente para tratar do *stream learning*, notou-se uma melhora no desempenho resultante. O Naive Bayes finalizou com acurácia de 45,57%, e o Perceptron atingiu 64,97%; ambos porém continuando aquém do razoável.

Apenas o k-NN do MOA, entre os algoritmos avaliados, obteve um resultado elevado, finalizando com uma acurácia média de 96,55% para k igual a 1. Essa foi, inclusive, superior à obtida na execução em *batch*, fazendo supor-se uma correlação maior entre as instâncias do CASAS temporalmente mais próximas entre si, possivelmente com pequenas mudanças de conceito, tendo sido o k-NN no MOA configurado para lembrar apenas os 1000 exemplos mais recentes do fluxo.

As Tabelas 1 e 2 resumem as acurácias obtidas por cada algoritmo, respectivamente, no WEKA e no MOA, bem como o tempo decorrido para treinar e testar os modelos:

Tabela 1. Algoritmos ordenados por acurácia no WEKA

Algoritmo	Acurácia	Treino	Teste
J48	94,82%	126,35 s	1,29 s
1-NN (lbk)	94,20%	0,1 s	2450,91 s
MultilayerPerceptron	76,88%	5274,21 s	2,08 s
BayesNet	69,03%	16,08 s	4,84 s
OneR	46,50%	3,19 s	0,41 S
NaiveBayes	38,62%	2,18 s	17,86 s

Tabela 2. Algoritmos ordenados por acurácia no MOA

Algoritmo	Acurácia	Treino	Teste
1-NN	96,55%	N/A	N/A
Perceptron	64,97%	N/A	N/A
NaiveBayes	45,57%	N/A	N/A
Weka.SMO	26,38%	N/A	N/A
Weka.J48	25,12%	N/A	N/A

5. Conclusões e Trabalhos Futuros

Este trabalho atingiu seu objetivo de experimentar e comparar diversos algoritmos de aprendizado de máquina abordados na disciplina, tendo incidido sobre dados reais em uma área com grande potencial de crescimento nos próximos anos, que consiste no reconhecimento de atividades humanas e o uso de sensores inteligentes no âmbito da IoT.

Cada algoritmo de aprendizado de máquina tem vantagens e desvantagens, não existindo um algoritmo universalmente superior. Assim, deve ser investigado em cada caso quais têm a melhor aplicabilidade para o domínio sob análise.

Este experimento demonstrou um bom resultado para a árvore de decisão J48, quando executada em *batch* nos dados do primeiro voluntário do conjunto CASAS, tendo sido obtida uma acurácia de 94,83%. Foi também o algoritmo com a melhor relação de custo-benefício, por ter gerado rapidamente um bom modelo. Entretanto, esse

desempenho decorreu de uma análise global envolvendo todos os dados do conjunto e múltiplas passadas pelos dados, o que nem sempre é viável, sobretudo quando se trata de dados gerados por sensores em tempo real.

Já o 1-NN, apesar de ter sofrido em modo *batch* no quesito de tempo durante a etapa de classificação, conseguiu obter acurácia de 94,20%, próxima à da árvore de decisão supramencionada. O algoritmo foi além, pois o 1-NN conseguiu manter um excelente desempenho ao mudar-se de estratégia para o *stream learning*, tendo obtido nesse caso a melhor acurácia (96,55%) entre todas as constatadas neste experimento.

Registra-se que os algoritmos probabilísticos mais simples ou de decisão baseada em uma única regra não se apresentaram adequados para tratar a complexidade do problema investigado.

Também se observou que, embora gerem modelos aproximados e tenham de lidar com restrições inerentes de memória e de tempo, os algoritmos de aprendizado em fluxo podem eventualmente obter uma acurácia superior à dos tradicionais em determinados casos, sobretudo quando há uma dinâmica surgindo em tempo real, como geralmente ocorre com fluxos contínuos de dados com sensores IoT.

Os conhecimentos práticos adquiridos mediante este experimento e teóricos desenvolvidos durante a disciplina permitirão, como trabalho futuro, uma investigação mais sólida de possibilidades de pesquisa conjugando Aprendizado de Máquina com Internet das Coisas, área de interesse do autor. Também podem ser desenvolvidos novos algoritmos de aprendizado de máquina para melhorar o desempenho da execução de forma distribuída e colaborativa na borda da rede IoT, contribuindo para a área promissora de *Edge Intelligence*.

6. Referências

- JOBANPUTRA, C.; BAVISHI, J.; DOSHI, N. Human Activity Recognition: A Survey. **Procedia Computer Science**, v. 155, p. 698–703, 2019.
- COOK, D. J.; CRANDALL, A. S.; THOMAS, B. L. *Human Activity Recognition from Continuous Ambient Sensor Data Data Set*. UCI, 20 de set. de 2019. Disponível em: <https://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+from+Continuous+Ambient+Sensor+Data>. Acesso em: 1 de mai. 2020.
- YANG, J.; LEE, J.; CHOI, J. Activity Recognition Based on RFID Object Usage for Smart Mobile Devices. **Journal of Computer Science and Technology**, v. 26, n. 2, p. 239–246, 2011.
- TOSCHI, G. M.; CAMPOS, L. B.; CUGNASCA, C. E. Home automation networks: A survey. **Computer Standards & Interfaces**, v. 50, p. 42–54, 2017.
- DAHMEN, J. et al. Smart secure homes: a survey of smart home technologies that sense, assess, and respond to security threats. **Journal of Reliable Intelligent Environments**, v. 3, n. 2, p. 83–98, 2017.

- U. A. B. U. A. BAKAR et al. Activity and Anomaly Detection in Smart Home: A Survey. **Smart Sensors, Measurement and Instrumentation Next Generation Sensors and Systems**, p. 191–220, 2015.
- COOK, D. J. et al. CASAS: A Smart Home in a Box. **Computer**, v. 46, n. 7, p. 62–69, 2013.
- KASTEREN, T. L. M. V.; ENGLEBIENNE, G.; KRÖSE, B. J. A. Human Activity Recognition from Wireless Sensor Network Data: Benchmark and Software. **Activity Recognition in Pervasive Intelligent Environments Atlantis Ambient and Pervasive Intelligence**, p. 165–186, 2011.
- ORDÓÑEZ, F. J.; TOLEDO, P. D.; SANCHIS, A. Activity Recognition Using Hybrid Generative/Discriminative Models on Home Environments Using Binary Sensors. **Sensors**, v. 13, n. 5, p. 5460–5477, 2013.
- SINGH, D. et al. Convolutional and Recurrent Neural Networks for Activity Recognition in Smart Environment. **Towards Integrative Machine Learning and Knowledge Extraction Lecture Notes in Computer Science**, p. 194–205, 2017.
- FACELI, Katti; LORENA, Ana Carolina; GAMA, João; CARVALHO, André Carlos Ponce de Leon Ferreira de. **Inteligência artificial: uma abordagem de aprendizado de máquina**, p. 78. GEN, 2011.
- BIFET, A.; KIRKBY, R. **Data stream mining: A Practical Approach**. Centre for Open Software Innovation, 2009.