Reconhecimento de atividades em Casas Inteligentes

Um comparativo entre técnicas de aprendizagem de máquina

Maria Luiza Nascimento Rodrigues

CENTRO DE INFORMÁTICA
UNIVERSIDADE FEDERAL DE PERNAMBUCO

Recife, 2017

Maria Luiza Nascimento Rodrigues

Reconhecimento de atividades em Casas Inteligentes

Um comparativo entre técnicas de aprendizagem de máquina

Trabalho de Conclusão de Curso apresentado no
Centro de Informática, da Universidade Federal de Pernambuco,
como requisito para a obtenção do grau de Bacharel em Engenharia da Computação

Orientador: George Darmiton da Cunha Cavalcanti

Julho de 2017

Centro de Informática
Universidade Federal de Pernambuco

Trabalho de Conclusão de Curso de Engenharia da Computação intitulado ***Reconhecimento de atividades em Casas Inteligentes***: **Um comparativo entre técnicas de aprendizagem de máquina** de autoria de Maria Luiza Nascimento Rodrigues, aprovada pela banca examinadora constituída pelos seguintes professores:

Prof. Dr. George Darmiton da Cunha Cavalcanti
Universidade Federal de Pernambuco

Prof. Dr. Kiev Santos da Gama
Universidade Federal de Pernambuco

Coordenador do curso de Engenharia da Computação do Centro de Informática
Renato Mariz de Moraes
CIN/UFPE

Recife, 13 de Julho de 2017

Centro de Informática, Universidade Federal de Pernambuco
Av. Jorn. Aníbal Fernandes, s/n - Cidade Universitária, Recife - PE, 50740-560
Fone: +55 (81) 2126-8430

AGRADECIMENTOS

À Deus, Criador da minha vida. **Aquele que era, que é e que sempre será**.

À meus Pais, Fernanda e Plínio, que dedicaram todos os esforços e paciência para que eu chegasse até aqui. Suas jornadas foram difíceis e cansativas, mas com muito ânimo se dispuseram a viver os meus sonhos. Hoje, a vitória não é exclusivamente minha.

À meus Irmãos, Erick e Rodolfo, que sempre me incentivaram e estiveram ao meu lado, sendo os meus primeiros amigos. O companheirismo deles foi fundamental nessa etapa da minha vida.

À minha Igreja, por todas as orações e ensinamentos, pois através deles aprendi que devemos sempre amar à Deus sobre todas as coisas e ao próximo como a nós mesmos.

À este Centro, seu corpo docente, direção e administração que oportunizaram esta janela na qual vislumbro meu futuro.

À meu orientador, George Darmiton, pelo incentivo, suporte e atenção que foram dados. A sua excelência no trabalho são inspiradores.

À meus amigos de infância, Sílvia Cristina, Ariel Carvalho, no qual pude viver momentos sublimes em minha vida. De cada momento pude extrair um aprendizado singular.

À meus amigos do Colégio da Polícia Militar, em especial Ana Carolina, Paloma Farias, Thays Divilart, Carlos Tito, Wandreson Rocha, que aspiraram este momento e, hoje, compreendem que minhas ausências foram justificáveis.

À meus amigos da turma de Engenharia da Computação 2012.2, em especial Larissa Camila, Jilmar Almeida, Luã Lázaro, Vinícius Sanguinete, Artur Franco e Gabriel Lima, que trilharam a árdua caminhada do curso ao meu lado, dispondo de horas preciosas para que nosso objetivo fosse alcançado. Obrigada por todo o apoio.

À meus amigos do Centro de Informática, em especial Gabriel Albuquerque, Heitor Araújo, Isabela Góes, Thaís Alves, Artur Montenegro, Yago Zacarias e Sílvio Santana, que, apesar de turmas diferentes, sempre demonstraram afeto e cuidado. A presença de vocês mudou significativamente a minha vida.

À meus amigos do LIVE, em especial Bianca Lisle, Caio Ferreira, Maria Cireno, Eron Neto e Marcus Felipe que durante todo o tempo de trabalho me apoiaram a prosseguir nos meus objetivos. Muito obrigada por tudo.

Por fim, a todos que, direta ou indiretamente, fizeram parte da minha formação. A todos, o meu muito obrigada.

RESUMO

Para levar uma vida funcionalmente independente faz-se necessário que as pessoas sejam capazes de realizar atividades do cotidiano, tais como: comer, dormir, tomar banho, etc. Tais atividades ocorrem de maneira intercalada e, na grande maioria das vezes, independente. Assim, automatizar o reconhecimento e o rastreamento dessas atividades diárias é um passo significativo para monitorar a saúde funcional de um morador, além do seu conforto e a economia à residência. Casas cujo monitoramento é feito por meio de sensores em tempo real e permite que sejam acessadas ou controlada remotamente são ditas casas inteligentes. Os dados captados nessas casas são rotulados e utilizados no treinamento de máquinas de aprendizagem supervisionada, como *Naive Bayes*, *Support Vector Machine* e *Random Forest*. Este trabalho tem por objetivo comparar o desempenho destas máquinas sobre as atividades realizadas nas casas inteligentes. Dispondo de dados do projeto CASAS: *Center for Advanced Studies in Adaptive Systems*, da Universidade do Estado de Washington, as informações, contidas em bases cuja residência acomoda apenas um morador, foram pré-processadas e suas características extraídas para que sejam classificadas por máquinas de aprendizagem. Dentre os classificadores utilizados, a *Random Forest* obteve uma acurácia média sobre todas as bases de, aproximadamente, 92% , mensurada por meio de uma validação cruzada de 5 folds em detrimento a aproximadamente 53% da *Support Vector Machine* e a apenas 10% da Naive Bayes.

Palavras-chave: Casas Inteligentes; Internet das coisas; Aprendizagem de máquina;

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS

ADLs – *Activities of daily living*

CASAS - *Center for Advanced Studies in Adaptive Systems*

IA - *Inteligência Artifical*

IoT – *Internet of Things*

ML - *Machine Learning*

SVM – *Support Vector Machine*

TAE - *Teoria de Aprendizado Estatístico*

Conteúdo

1 INTRODUÇÃO

1.1 Motivação

A maior parte das interações na Internet é realizada entre seres humanos. No entanto, em um futuro próximo, qualquer "coisa" (*thing*) poderá ser hospedada na grande rede. As comunicações serão concebidas não apenas entre humanos mas também entre humanos e coisas e entre coisas sem a interação com seres humanos. Esse novo paradigma chamado *Internet of Things*, IoT, será amplamente utilizado [18]. Sob essa perspectiva, surge a ideia de Casas Inteligentes: Casas onde podem ser encontradas uma gama vasta de soluções integradas a tecnologia da informação. O desenvolvimento de sensores mais eficazes, de baixo custo e fácil usabilidade tem suscitado o interesse no desenvolvimento de ambientes inteligentes. Quando integrados, esses sensores tornam-se uma ferramenta poderosa no monitoramento e rastreamento de atividades. O envelhecimento da população, os custos associados ao tratamento de saúde e o advento de tecnologias cada vez mais robustas fomentaram a necessidade do indivíduo de permanecer independente em suas próprias casas.

Para o funcionamento independente da casa é necessário que *Activities of daily living*, ADLs, como comer, dormir, tomar banho, sejam realizadas pelos indivíduos. Por isso, automatizar o reconhecimento destas atividades é fundamental. Tomando como referência o cotidiano de pacientes de Alzheimer, classificar e rastrear as suas atividades são uma necessidade latente aos familiares e cuidadores a fim de monitorar a saúde funcional do enfermo [2]. O uso de sensores, em situações como estas, torna-se de grande auxílio para reconhecer as atividades. Uma casa, porém, poderia comportar centenas ou milhares de sensores que, por sua vez, geram dados complexos e volumosos, com isso o processo de aprendizado é desafiador.

A aprendizagem da atividade desempenha um papel fundamental na concepção de agentes inteligentes. Russell e Norvig (1994) definem um agente como uma entidade que percebe o seu ambiente através de sensores e atua sobre o ambiente através de atuadores. Em um ambiente controlado, as informações brutas de indivíduos são capturadas e armazenadas em uma base de dados. O agente inteligente, por sua vez, analisa essas informações para prever e reconhecer atividades que estejam sendo executados pelo residente. Com isso, decisões podem vir a serem tomadas a fim de alcançar o objetivo almejado. As casas inteligentes, neste contexto, passam a desempenhar o papel de agentes inteligentes.

Máquinas de aprendizagem supervisionada utilizam de métodos para gerar uma função preditiva $F : X \rightarrow Y$ que mapeia X – atributos de uma dada instância – para uma predição Y – classe de uma instância. Uma vez treinada, um conjunto de testes, com

instâncias que não foram utilizadas no conjunto de treinamento, é utilizado para validar a função F gerada. Um classificador é dito apto para o problema se as variáveis-alvo geradas pela função preditiva forem compatíveis com o vetor de características do teste [12].

A fim de otimizar a predição realizada pelas máquinas de aprendizagem, faz-se necessário o uso de técnicas que extraiam informações relevantes dos dados utilizados. Para tal, utiliza-se a mineração de dados: análise de grandes conjuntos de dados a fim de resumir os dados de uma forma que eles sejam tanto úteis quanto compreensíveis ao dono dos dados. O uso desta técnica permite que os especialistas concentrem esforços apenas em partes mais significativas dos dados. Porém, utiliza-se de grande esforço computacional se forem mal selecionados.

Trabalhos recentes, como o CASAS [5], o House_n[6], do MIT, e o *The Aware Home* [7] da Georgia Tech, que utilizam diversas abordagens de aprendizagem de máquina, tais como *Support Vector Machine* para o projeto CASAS, têm mostrado que há uma vasta área ainda a ser explorada para aperfeiçoar o reconhecimento de atividades. Aumentar a robustez do sistema garantindo que, se adapte a diferentes ambientes inteligentes impulsiona as atuais pesquisas.

1.2 Objetivos

1.2.1 Objetivo

Este trabalho tem o objetivo de avaliar a média e o desvio padrão de máquinas de aprendizagem supervisionada como a *Naive Bayes*, *Support Vector Machine* e *Random Forest* na classificação de atividades em casas inteligentes através do banco de dados público do projeto CASAS, cujas bases utilizadas serão: hh106, hh110, hh123, hh124 e hh130.

1.3 Estrutura do trabalho

A fim de atingir os objetivos anteriormente apresentados, este trabalho será estruturado da seguinte maneira: No Capítulo 2 será abordada a fundamentação teórica do projeto detalhando as máquinas de aprendizagem supervisionada que foram utilizadas, bem como as técnicas de mineração de dados aplicadas. No Capítulo 2, também, será apresentado as abordagens de avaliação de classificadores que foram utilizadas. No Capítulo 3 é apresentada a metodologia ao qual o projeto foi desenvolvido. No Capítulo 4, os resultados obtidos são apresentados. Por fim, no Capítulo 5 temos as considerações finais e os trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Internet das Coisas

Definir Internet das coisas tem sido uma temática discutida há muitos anos. Apresentada como revolução tecnológica, o paradigma tem por objetivo conectar diferentes dispositivos por meio da internet. Essa conexão extrapola as fronteiras de *smartphones* e computadores, e tem por finalidade atingir, em larga escala, os objetos mais comuns em nossas residências, tais como: Geladeiras, carros, monitores de saúde, etc. Com o crescimento da internet nos últimos 10 anos, a interação entre os dispositivos tem ficado cada vez mais frequentemente [18].

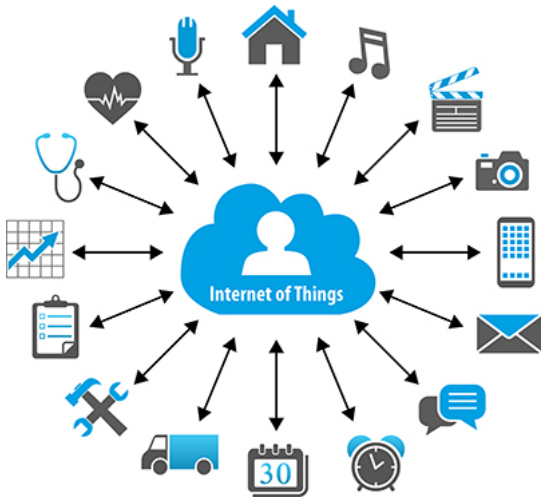


Figura 1: Ilustração da Internet da Coisas

Fonte: http://www.decom.ufop.br/imobilis/iot-coletando-dados/

Em 1991 começou-se a discutir sobre a conexão dos dispositivos uma vez que a conexão TCP/IP e a Internet tornaram-se acessíveis. Bill Joy, cofundador da Sun Microsystems foi o precursor da ideia. Em 1999, porém, Kevin Ashton, do Massachusetts Institute Technologym propôs o termo "Internet das Coisas"sendo disseminada por meio do aritgo "A coisa da Internet das Coisas"para o RFID Journal. Segundo Ashton, a falta de tempo na rotina das pessoas fará com que necessitem se conectar a internet de várias maneiras. Com a mobilidade e tecnologia avançando, será possível acumular dados e até o movimento dos corpos com precisão. Esses registros poderão servir para otimizar e economizar recursos naturais e energéticos, por exemplo, além de infinitas facilidades pessoais e de saúde.

2.1.1 Características da Internet das Coisas

Sete características embasam a internet das coisas: Inteligência, arquitetura, sistema complexo, considerações de tamanho, considerações de tempo, considerações de espaço, e tudo como serviço. Essas características devem ser consideradas no desenvolvimento de soluções de Internet das coisas ao longo de todas as fases de concepção, desenvolvimento, implementação e avaliação[18].

- **Inteligência:** Aplicação do conhecimento, ou seja, transformar os dados brutos em conhecimento. Pode ser feito através da coleta, modelagem e raciocínio do contexto.
- **Arquitetura:** A arquitetura híbrida deve suportar múltiplas arqutiteturas anexadas.
- **Sistemas Complexos:** Por compreender uma vasta variedade de objetos que interagem de forma autônoma. Ou seja, a complexidade do sistema para assimilar as mudanças constantes no sistema é elevada.
- **Tamanho:** Estima-se que em meados de 2020 já existam cerca de 100 bilhões de dispositivos conectados à internet. A internet das coisas deve facilitar a interação entre esses dispositivos comportando essa expansão do sistema.
- **Tempo:** Processar os dados em tempo real é essencial para este paradigma tendo vista que bilhões de eventos paralelos e simultâneos estarão ocorrendo.
- **Espaço:** Possuir a localização específica de um dado dispositivo será fundamental já que desempenha um papel significativo no contexto.
- **"Tudo como serviço":** O modelo "Tudo como serviço"é altamente eficiente, escalável e fácil de usar. Internet das coisas exige uma quantidade significativa de infra-estrutura a ser postas em prática a fim de tornar a sua visão uma realidade, onde ele iria seguir uma abordagem baseada na comunidade ou multidão.

2.1.2 Casas Inteligentes

Diante do crescimento deste paradigma, ambientes cada vez mais conectados tem se tornado presente atualmente. Assim, surge o conceito de casas inteligentes: Uma casa que incorpora sistemas de automação avançados para fornecer aos habitantes monitoramento e controle sofisticados sobre as funções da residência. Por exemplo, uma casa inteligente pode controlar as operações de iluminação, temperatura, multimídia, segurança, janela e porta, bem como muitas outras funções.

Em 2003, o Departamento de Comércio e Indústria do Reino Unido (DTI) apresentou a seguinte definição para uma casa inteligente: *"Uma habitação que incorpora uma rede de comunicações que conecta os principais aparelhos elétricos e serviços, e permite que sejam controlados remotamente, monitorados ou acessados"*.

Figura 2: Representação dos elementos de uma casa inteligente

Fonte: http://www.creativebits.cc/

A figura 2 ilustra o cenário das casas inteligentes de tal maneira que todos os dispositivos estarão integrados entre si disponibilizando informações sobre o ambiente para o morador.

2.2 Mineração de dados

Mineração de dados é a exploração e a análise, por meio automático ou semiautomático, de grandes quantidades de dados, a fim de descobrir padrões e regras significativos.

Utiliza sistemas matemáticos, por meio de algoritmos sofisticados, para segmentar os dados e avaliar a probabilidade de ocorrência de determinados eventos no futuro [20].

As principais propriedades da mineração de dados são:

- Detecção automática de padrões
- Predição de eventos
- Geração de informação para ações práticas
- Foco em grandes conjuntos e bancos de dados

Para gerar resultados significativos, a mineração de dados exige um processo constante de coleta, processamento e análise de informações. Por se tratar de um método de análise profunda e que abrange uma enorme quantidade de dados, o sistema responsável por organizar e estudar as informações colhidas precisa apresentar algumas características fundamentais, como contar com diferentes formas de classificação, clusterização e preparação de dados.

Coleta dos dados

Por lidar com uma grande quantidade de informações, o sistema deve contar com uma capacidade robusta para coletar e processar os dados.

Classificação de informações

Após a coleta dos dados, a análise só é possível com a organização eficiente das informações. Esta organização começa com a classificação dos dados. A tarefa de classificação consiste em construir um modelo de algum tipo que possa ser aplicado a dados não classificados visando categorizá-los em classes. Um objeto é examinado e classificado de acordo com uma classe definida (Harrison, 1998).

Clusterização dos dados

A segmentação é um processo de partição de uma população heterogênea em vários subgrupos ou clusters mais homogêneos. Na segmentação, não há classes predefinidas, os registros são agrupados de acordo com a semelhança, o que a diferencia da tarefa de classificação.

Preparação dos dados

Segundo Fayyad (1996), a tarefa de sumarização envolve métodos para encontrar uma descrição compacta para um subconjunto de dados.

2.3 Aprendizagem de máquina

A aprendizagem de máquina tem por objetivo desenvolver algoritmos que façam a aprendizagem automaticamente sem intervenção ou assistência humana. O paradigma de aprendizado da máquina pode ser visto como "programação por exemplo". Por ser uma subárea central da inteligência artificial, IA, é muito improvável que possamos construir qualquer tipo de sistema inteligente, como linguagem ou visão, sem usar o aprendizado pois são tarefas de difícil compreensão. Além disso, não consideramos um sistema realmente inteligente se fosse incapaz de aprender, pois o aprendizado é o cerne da inteligência. Embora seja uma subárea de IA, a aprendizagem automática também se intercepta amplamente com outros campos, tal como estatística, física, ciência computacional teórica e entre outros [8].

No sentido mais amplo, qualquer método que incorpore informações de amostras de treinamento no desenvolvimento de um classificador emprega aprendizado visto que, quase todos os problemas de reconhecimento de padrões práticos ou interessantes são tão difíceis que não podemos adivinhar a decisão de classificação antes do tempo. A criação de classificadores envolve, então, a postura de uma forma geral de modelo ou forma do classificador e o uso de padrões de treinamento para aprender ou estimar os parâmetros desconhecidos do modelo. A aprendizagem refere-se a alguma forma de algoritmo para reduzir o erro em um conjunto de dados de treinamento.

2.3.1 Aprendizagem Supervisionada

Dado um espaço de entrada X, representação dos dados, e um espaço de saída Y, rótulos ou valores associados aos dados, o principal objetivo consiste em encontrarmos um mapeamento f, que associa cada entrada em X a respectiva saída em Y [13]. Simplificando podemos definir que cada valor no espaço de entrada X é um vetor de números d-dimensional. Esse vetor é chamado de vetor de características. É importante notar que seus elementos podem ser mais complexos, como uma imagem, um gráfico, etc. Similarmente, cada valor no espaço de saída Y pode ser, a princípio, qualquer coisa. Entretanto, a maior parte dos métodos assume que cada elemento de Y é uma categoria de um conjunto finito ou um escalar de valor real. No primeiro caso, o problema é chamado de classificação ou reconhecimento de padrões e no segundo é chamado de regressão.

Nesta seção serão apresentados os principais algoritmos de aprendizagem utilizados para o reconhecimento de atividades em casas inteligentes. São eles: *Naive Bayes, Support Vector Machine* e *Random Forest*.

2.3.2 Naive Bayes

O classificador Naive Bayes é um algoritmo de aprendizagem de máquina supervisionada, da família dos classificadores probabilísticos, baseado no Teorema de Bayes com a suposição "ingênua"de que há independência entre cada par de características da entrada do sistema [14].

Suponha o seguinte problema: Conforme Figura 6, queremos definir se uma nova entrada fará parte do grupo das bolas verdes ou das bolas vermelhas.

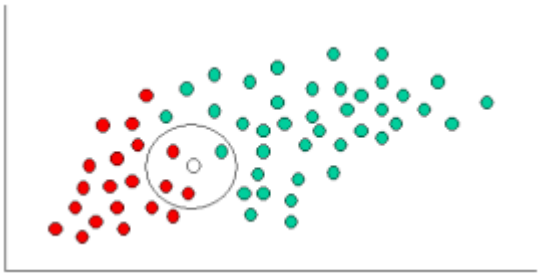


Figura 3: Problema de decisão: Naive Bayes

Fonte: https://stackoverflow.com/questions/10059594/a-simple-explanation-of-naive-bayes-classification

É possível avaliar que há mais bolas verdes a bolas vermelhas, com isso é razoável imaginar que a entrada seja mais provavelmente verde que vermelha. Na análise bayesiana isto é conhecido como probabilidade a priori. As probabilidades a priori são baseadas na experiência anterior, isto é, as porcentagens de objetos verdes e vermelhos.

Assim:

$$P(\omega_j) : \frac{\#\omega_j}{\Omega} \tag{1}$$

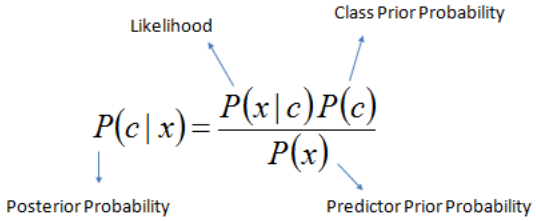
No qual ω_j indica a cor da bola, verde ou vermelha, e Ω o espaço amostral.

Uma vez calculada a probabilidade a priori, podemos classificar um novo objeto. Os objetos estão bem agrupados, sendo razoável supor que os objetos mais verdes (ou vermelhos) na vizinhança de X, mais provável que os novos casos pertençam a essa cor particular. A partir disso, calculamos a probabilidade:

$$P(\text{Vizinhança de X dado que é vermelho}) = \frac{\text{Número de vizinhos vermelhos}}{\text{Número total de vermelhos}} \tag{2}$$

$$P(\text{Vizinhança de X dado que é verde}) = \frac{\text{Número de vizinhos verdes}}{\text{Número total de verdes}} \tag{3}$$

Na análise bayesiana, a classificação final é produzida combinando ambas as fontes de informação, isto é, a probabilidade a priori e a probabilidade da entrada, para formar uma probabilidade posterior usando a chamada regra de Bayes:



$$P(c \mid \mathbf{X}) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

Figura 4: Probabilidade Bayesiana

Fonte: http://chem-eng.utoronto.ca/ datamining/dmc/naive_bayesian.htm

No qual:

- $P(c|x)$ - é a probabilidade da classe c dada o preditor x.
- $P(c)$ - é a probabilidade da classe
- $P(x|c)$ - é a probabilidade do preditor x dada a classe c.
- $P(x)$ - é a probabilidade do preditor

Finalmente podemos classificar a entrada como verde ou vermelha para o cenário descrito.

2.3.3 Support Vector Machine

A *Support Vector Machine*, SVM, são máquinas de aprendizagem embasadas pela teoria de aprendizado estatístico, desenvolvida por Vapnik. Esta teoria pode ser compreendida da seguinte maneira:

Seja f um classificador e F o conjunto de todos os classificadores que um determinado algoritmo 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 $\hat{f} \in F$ [10].

A Teoria de Aprendizado Estatístico, TAE, estabelece condições matemáticas que auxiliam na escolha de um classificador particular $\hat{f}$ a partir de um conjunto de dados de treinamento. Essas condições levam em conta o desempenho do classificador no conjunto de treinamento e a sua complexidade, com o objetivo de obter um bom desempenho também para novos dados do mesmo domínio.

Quando nos referimos a bom desempenho de um classificador f estabelecemos que este obtenha o menor erro durante o treinamento, onde o erro pode ser mensurado pelas

predições erradas de f. Sendo assim, definimos erro médio empírico $R_{emp}(f)$ como sendo a medida de perda entre a resposta desejada e a resposta real. A equação (4) demonstra, matematicamente, a definição de risco empírico[9].

$$R_{emp}(f) = \frac{1}{n}\sum_{n=1}^{n} c(f(\mathbf{x_i}, y_i)) \tag{4}$$

No qual $c(\cdot)$ é a função de custo relacionada a previsão $f(\mathbf{x_i})$ com a saída desejada y_i, onde um tipo de função de custo é a "perda 0/1"definida pela equação abaixo. O processo de busca por uma função f' que represente um menor valor de R_{emp} é denominado de Minimização do Risco Empírico.

$$c(f(\mathbf{x_i}, y_i)) = \begin{cases} 1, & y_i f(x_i) \leq 0 \\ 0, & \text{Caso contrário} \end{cases} \tag{5}$$

Sobre a hipótese de que os padrões de treinamento (x_i, y_i) são gerados por uma distribuição de probabilidade $P(x, y)$ em $\mathbb{R}^N \times \{-1, +1\}$ sendo P desconhecida. A probabilidade de classificação incorreta do classificador f é denominada de Risco Funcional, que quantifica a capacidade de generalização, conforme é mostrado pela equação (6).

$$R(f) = \int c(f(\mathbf{x_i}, y_i))\, dP(\mathbf{x_i}, y_i) \tag{6}$$

Durante processo de treinamento, $R_{emp}(f)$, pode ser facilmente obtido, ao contrário de $R(f)$, pois em geral a distribuição de probabilidades P é desconhecida.

A partir disto, dado um conjunto de dados de treinamento (x_i, y_i) com $x_i \in \mathbb{R}^N$ e $y_i \in \{-1, +1\}$, $i = 1, 2, \cdots, n$, sendo x_i o vetor de entrada e y_i o rótulo da classe.

O objetivo então é estimar uma função f: $\mathbb{R}^N \to \{-1, +1\}$. Caso nenhuma restrição seja imposta na classe de funções em que se escolhe a estimativa f, pode ocorrer que a função obtenha um bom desempenho no conjunto de treinamento, porém não tendo o mesmo desempenho em padrões desconhecidos, sendo este fenômeno denominado de "overfitting". Em outras palavras, a minimização apenas do risco empírico $R_{emp}(f)$ não garante uma boa capacidade de generalização, sendo desejado um classificador f^* tal que $R(f^*) = min_{f \in F} R(f)$, onde F é o conjunto de funções f possíveis.

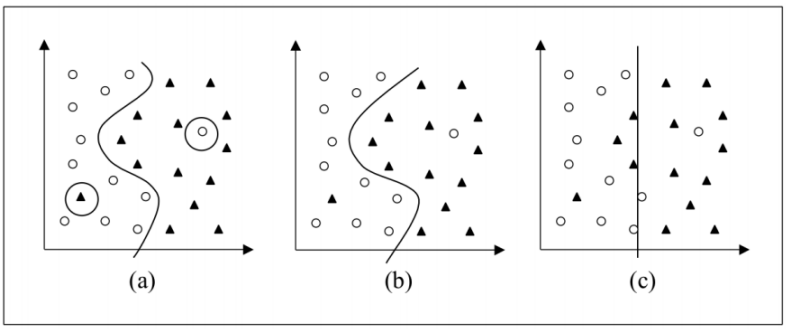


Figura 5: Classes de hiperplanos com um hiperplano ótimo

Fonte: HEARST. 1998.

A figura 8 mostra um exemplo onde uma classe de funções pode ser utilizada para separar padrões linearmente separáveis. É necessário determinar uma função que minimize o R_{emp}, representado na figura 8b.

A TAE provê formas de limitar a classe de funções (hiperplanos), com o intuito de prevenir modelos ruins, ou seja, que levem ao "overfitting", implementando uma função com a capacidade adequada para o conjunto de dados de treinamento. Estas limitações são impostas ao risco funcional da função [9].

2.3.4 Random Forest

Considerando que haja uma compreensão prévia do algoritmo de árvores de decisão, podemos definir o *Random Forest* como segue:

"Uma Random Forest é um classificador composto por uma coleção de árvores de decisão $\{h(\mathbf{x}, \Theta_k), k = 1, \cdots\}$ *onde* $\{\Theta_k\}$ *são vetores aleatórios independentemente distribuídos de forma idêntica e cada árvore lança um voto de unidade para a classe mais popular na entrada* x*"* [12].

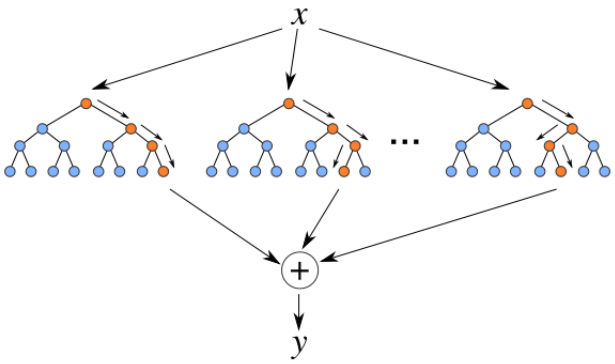


Figura 6: Esquema do algoritmo Random Forest

Fonte: https://kgpdag.wordpress.com/

Simplificando, podemos dizer que a *Random Forest* é uma combinação de árvores classificadoras, de modo que cada árvore depende dos valores de um vetor aleatório amostrado independentemente e com a mesma distribuição para todas as árvores da floresta. Essas idéias também são aplicáveis à regressão. O *Random Forest* possui por características:

2.4 Técnicas de Avaliação de Combinadores

Dado um classificador, como estimar a taxa de erro? Dado dois classificadores eles são iguais? A análise de classificadores é, sobretudo, experimental. Algumas métricas são usuais para classificação, tais como: taxa de erro, taxa de acerto, matriz de confusão. Usar toda amostra para treinamento e computar a taxa de erro no conjunto de treino não é um bom indicador em relação ao que vai ser observado no futuro. Uma solução simples para dados abundantes é dividir os dados em dois subconjuntos mutuamente excludentes: treinamento e teste.

Abaixo, algumas técnicas utilizadas quando há uma limitação nos dados:

1. **Estimação Holdout**: Usa $\frac{2}{3}$ para treinamento e $\frac{1}{3}$ para teste. Esta técnica pode ocasionar problemas porque uma classe pode estar ausente no conjunto de teste.

2. **Holdout Repetido**: Utiliza-se do processo da estimação holdout porém com diferentes subamostras, tal que: Seleciona uma amostra com ou sem estratificação para treinamento.

3. **Validação Cruzada**: São divididos em k conjuntos com mesmo tamanho onde cada subconjunto é usado como teste e os demais como treino.

4. **Validação cruzada leave-one-out**: número de folds é número de exemplos. Classificador é construído n vezes. Não tem estratificação.

5. **Bootstrap**: É um método de estimação que usa amostragem com reposição para formar o conjunto de treinamento. É indicado quando o conjunto de dados é pequeno. Retira uma amostra aleatória de tamanho n de um conjunto de n exemplos com reposição.

3 MÉTODO PARA RECONHECIMENTO DE ATIVIDADES

Neste capítulo é descrita a metodologia utilizada no desenvolvimento do software de reconhecimento de atividades em casas inteligentes. As etapas do processo são apresentadas na Figura 8 e o respectivo detalhamento é feito nas subseções seguintes. Estas etapas foram baseadas na metodologia proposta em [1].

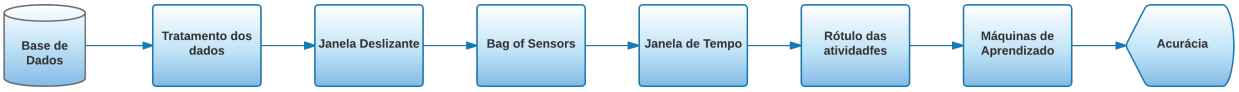


Figura 7: Fluxograma do Método de Reconhecimento de Atividades

Antes de detalhar as etapas da metodologia se faz necessária a definição de algumas notações que serão utilizadas para descrever o processo. Seja $s_1, s_2, \cdots, s_n$ a sequência de sensores. O dia, a hora e o modo de operação do sensor estão associados juntamente a cada evento de sensor presente na casa, tais como temperatura, umidade, e etc. O evento, por sua vez, pode ser descrito como a leitura realizada pelo sensor naquele instante. Com essas informações, temos por objetivo classificar cada evento do sensor a uma atividade que esteja sendo realizada.

Para tornar clara as etapas que sucederão, a Figura 9 apresenta o formato da base de dados:

```
1  2011-06-15 00:03:09.817697 LS006 0
2  2011-06-15 00:17:44.211833 LS005 4
3  2011-06-15 00:23:09.72767 LS006 1
4  2011-06-15 00:32:12.027662 BATV103 3180
5  2011-06-15 00:37:44.063746 LS005 5
6  2011-06-15 00:43:09.615905 LS006 0
7  2011-06-15 00:47:43.992647 LS005 4
8  2011-06-15 01:03:09.498286 LS006 1
9  2011-06-15 01:07:43.887796 LS005 5
10 2011-06-15 01:17:43.846326 LS005 4
11 2011-06-15 01:23:09.380385 LS006 0
12 2011-06-15 01:27:43.776285 LS005 5
13 2011-06-15 01:37:43.727408 LS005 4
14 2011-06-15 01:43:09.302783 LS006 1
15 2011-06-15 01:43:14.277702 M015 ON Sleep="begin"
16 2011-06-15 01:43:17.068053 M015 OFF
17 2011-06-15 01:47:43.659496 LS005 5
18 2011-06-15 01:53:09.245733 LS006 0
19 2011-06-15 01:57:43.601974 LS005 4
20 2011-06-15 02:01:31.402581 M015 ON
```

Figura 8: Trecho extraído da Base hh123 mostrando a formatação dos dados

3.1 Tratamento dos dados

Como dito, a entrada possui vários campos: Data, hora, sensor e modo de operação do sensor. Essas informações precisam ser tratadas a fim de facilitar a compreensão para a máquina de aprendizado.

Para tal, os modos de operação são processados conforme a natureza dos seus dados: analógicos e digitais. Se analógicos, serão normalizados em valores entre $[0, 1]$ de acordo com o alcance dos dados. Por exemplo, seja a entrada a seguir:

```
5   2011-06-15 00:37:44.063746 LS005 5
6   2011-06-15 00:43:09.615905 LS006 0
7   2011-06-15 00:47:43.992647 LS005 4
8   2011-06-15 01:03:09.498286 LS006 1
9   2011-06-15 01:07:43.887796 LS005 5
```

Figura 9: Base de dados: Informações analógicas

Tem-se então que:

$$[5, 0, 4, 1, 5] \rightarrow [1.0, 0, 0.8, 0.2, 1.0]$$

Caso sejam discretos, os dados serão convertidos em valores numéricos binários. Por exemplo:

```
15  2011-06-15 01:43:14.277702 M015 ON Sleep="begin"
16  2011-06-15 01:43:17.068053 M015 OFF
```

Figura 10: Base de dados: Informações digitais

Com isso, podemos definir então:

$$[ON, OFF] \rightarrow [1, 0]$$

3.2 Janela deslizante

Assumimos que as atividades são modeladas por uma sequência de eventos de sensor. O janelamento assumido aqui, considera que em um dado intervalo de capturas o contexto será definido pela última atividade que precedeu o processo.

Formalmente, uma janela é definida como W_i e representada pela sequência $\left[s_{i \rightarrow \Delta s}, s_i\right]$, tal que s_i refere-se ao sensor. O parâmetro Δs define o tamanho da janela e varia do contexto do problema. Δs pode ser obtido considerando o número médio de eventos de sensores que abrangem diferentes atividades [1]. Encontrar o tamanho ótimo das janelas é um problema desafiador, pois informações que definem uma dada atividade podem ser desconsideradas se o tamanho for muito pequeno ou informações desnecessárias podem ser associadas se for muito grande [4]. Duas abordagens podem ser consideradas para redimensionar o tamanho da janela:

1. Dividir sequências inteiras em intervalos de tempo iguais;

2. Dividir em sequências de eventos de sensor.

Devido aos dados obtidos a partir dos sensores apresentarem formatação discreta, a primeira abordagem foi desconsiderada. Porém, nas bases consideradas, existem sensores digiais. Além disto, o tamanho do intervalo de tempo varia entre os eventos dos sensores. A segunda abordagem, no entanto, apresenta-se como conveniente tendo em vista a pluralidade dos sensores. No mais, não depende de eventos futuros para classificar eventos pretéritos.

O tamanho da janela, definido para o problema, é fixo pois, fornece uma informação de quanto movimento e interação estão ocorrendo dentro da janela. Assim, seja $\{w_1, w_2, \cdots, w_L\}$ tal que $w_1 = min\{w_s(A_1), w_s(A_2), \cdots, w_s(A_M)\}$ e $w_L = median$ $\{w_s(A_1), w_s(A_2), \cdots, w_s(A_m)\}$, e $w_s(A_m)$, por sua vez, corresponde a média do tamanho da janela de atividade A_m. O tamanho intermediário da janela entre w_1 e w_L é obtido dividindo o intervalo entre eles igualmente. Assim:

$$w^* = argmax_{w_l}\{P(w_l|A_m)\} \tag{7}$$

Para tal, estamos estimando o tamanho mais provável da atividade A_m. Com isso, devemos calcular, também, a probabilidade $P(A_m|S_i)$. Assim, dado o identificador de sensor s_i para o evento em consideração, podemos determinar a atividade A^* mais provável associada como:

$$A^* = argmax_{A_m}\{P(A_m|s_i)\} \tag{8}$$

Assim, para o evento do sensor s_i o tamanho ótimo da janela pode ser determinado combinando as duas equações de acordo com a seguinte fatoração:

$$\begin{aligned} w^* &= max_{w_l} P(w_l|s_i) \\ &= max_{w_l}\left[P(w_l|A_m) \times P(A_m|s_i)\right] \end{aligned} \tag{9}$$

No cenário proposto vimos, conforme :

```
2011-06-15 00:03:09.817697 LS006 0
2011-06-15 00:17:44.211833 LS005 4
2011-06-15 00:23:09.72767 LS006 1
2011-06-15 00:32:12.027662 BATV103 3180
2011-06-15 00:37:44.063746 LS005 5
2011-06-15 00:43:09.615905 LS006 0
2011-06-15 00:47:43.992647 LS005 4
2011-06-15 01:03:09.498286 LS006 1
2011-06-15 01:07:43.887796 LS005 5
2011-06-15 01:17:43.846326 LS005 4
```

Atividade: "Other"

```
2011-06-15 00:17:44.211833 LS005 4
2011-06-15 00:23:09.72767 LS006 1
2011-06-15 00:32:12.027662 BATV103 3180
2011-06-15 00:37:44.063746 LS005 5
2011-06-15 00:43:09.615905 LS006 0
2011-06-15 00:47:43.992647 LS005 4
2011-06-15 01:03:09.498286 LS006 1
2011-06-15 01:07:43.887796 LS005 5
2011-06-15 01:17:43.846326 LS005 4
2011-06-15 01:23:09.380385 LS006 0
```

Atividade: "Other"

```
2011-06-15 00:43:09.615905 LS006 0
2011-06-15 00:47:43.992647 LS005 4
2011-06-15 01:03:09.498286 LS006 1
2011-06-15 01:07:43.887796 LS005 5
2011-06-15 01:17:43.846326 LS005 4
2011-06-15 01:23:09.380385 LS006 0
2011-06-15 01:27:43.776285 LS005 5
2011-06-15 01:37:43.727408 LS005 4
2011-06-15 01:43:09.302783 LS006 1
2011-06-15 01:43:14.277702 M015 ON Sleep="begin"
```

Atividade: Sleep

Figura 11: Bases de dados: Janelas deslizantes

3.3 Bag of Sensors

Similar ao modelo *bag of words*, comumente utilizado em mineração de texto e imagens, no qual um conjunto de características são representadas em um vetor que desconsidera a sequência de sua aparição. Por exemplo, na mineração de texto, um *bag of words* pode ser representado pelo conjunto de palavras e sua frequência de aparição em um documento. Sob essa perspectiva, um *bag of sensors* pode ser definido como um *bag of words* para sensores baseados em eventos. Assim, cada entrada pode ser representada como uma tupla (s_i, t_i), em que s_i representa o sensor e t_i a sua frequência de aparição [4].

Definimos então, o *bag of sensors* para este projeto, como uma lista cujas entradas são as contagens de uso de cada sensor para a dada janela. Podemos expressar como:

$$bag = [[a_1, a_2, \cdots, a_n], [b_1, b_2, \cdots, b_m], \cdots, [z_1, z_2, \cdots, z_t]] \tag{10}$$

Como ilustra na Equação (10), podemos definir então que: Seja a, b, z os sensores presentes na base de dados, tais como temperatura, bateria da TV, e sensor de movimento, por exemplo, logo os índices refletem a qual sensor em específico estamos nos referindo. Por exemplo, o sensor a_1 refere-se ao primeiro sensor de temperatura, o sensor b_1 ao primeiro sensor de bateria da TV e o z_1, por sua vez, ao primeiro sensor de movimento. E, assim, o *bag of sensors* contém a frequência de ativação dos sensores em cada janela.

3.4 Janela de tempo

Os seres humanos são seres de hábitos. Como os indivíduos seguem horários bastante semelhantes, a noção de tempo nos permite coordenar as atividades em grupo. Por este motivo, o tempo da sequência do evento representa uma característica importante.

Para o nosso problema, consideramos que a janela de tempo limita-se a janela atual. Com isso, convertemos a hora em milisegundos conforme descrito a seguir:

$$time = ((((((hour \times 60) + minutes) \times 60) + seconds) \times 1000) + milliseconds) \tag{11}$$

Logo, para cada janela teremos: o tempo inicial da janela, o tempo final e a duração da janela em milisegundos, respectivamente. A Figura 13 ilustra o passo descrito:

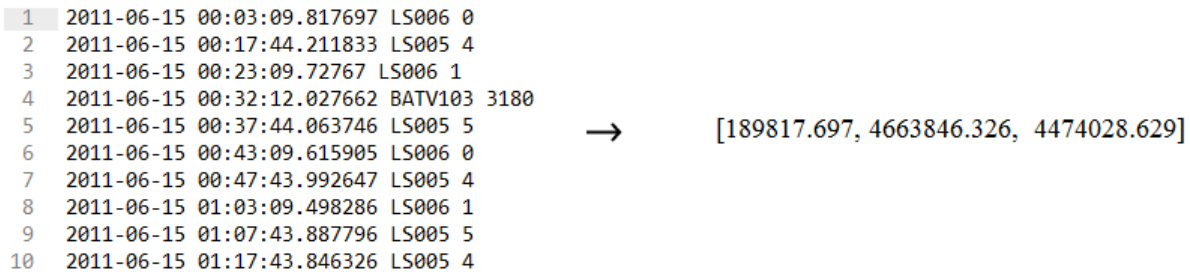


Figura 12: Bases de dados: Janela de Tempo

3.5 Rótulo das atividades

As atividades necessitam de um estilo de formatação específica antes que sejam utilizadas para o treinamento e teste das máquinas de aprendizagem. Cada atividade é expressa por extenso na base de dados, ou seja, não utilizam de valores numéricos para a sua representação. Esse mapeamento foi realizado para que a cada atributo preditor, *bag of sensors* e janela de tempo, tivesse um valor associado indicando a sua atividade. Assim:

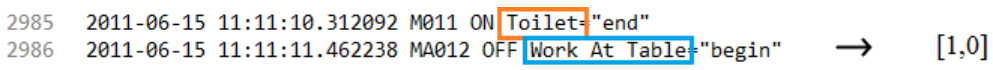


Figura 13: Bases de dados: Rótulo das Atividades

No qual, o vetor $[1, 0]$ representa os rótulos que serão utilizados pelas máquinas de aprendizagem.

3.5.1 Máquinas de Aprendizagem

Para o reconhecimento das atividades optou-se por utilizar máquinas de aprendizagem supervisionada tal como descritas no capítulo anterior. Apesar dos diferentes algoritmos, a formatação dos dados é a mesma para todos os casos. Assim:

$$input = [bag\ of\ sensors + \text{janela de tempo}] \tag{12}$$

$$labels = [1, 2, 1, \cdots, 10] \tag{13}$$

No qual, o *input* é representado por um vetor cujos elementos referem-se ao *bag of sensors* e as informações da janela de tempo de cada janela. Além disto, as *labels* informam qual atividade estava sendo executada durante cada janela. Com isso, podemos treinar as máquinas sob as respectivas bases de dados e, por meio de validação cruzada, estimar a acurácia do classificador em questão.

4 RESULTADOS E ANÁLISE

Nesta seção são apresentados os resultados obtidos usando cinco bases de dados distintas disponibilizadas pelo projeto CASAS[5]: hh106, hh110, hh123, hh124, hh130, utilizando a metodologia anteriormente proposta no Capítulo 3. Para cada experimentação foi considerada uma validação cruzada estratificada de 5 *folds* e, para cada *fold*, foi considerado sua acurácia. Ao término de cada execução, a média e o desvio da acurácia foram calculadas entre todos os *folds*.

Será feita uma descrição da tecnologia utilizada a fim de apresentar características que proporcionaram a sua escolha, bem como das bases de dados utilizadas.

4.1 Python

Python é uma linguagem de programação de alto nível, interpretada, multiparadigma, desenvolvida por Guido Van Rossum, em 1991. Seus objetivos de projeto eram legibilidade e produtividade. Possui como principais características:

- Baixo uso de caracteres especiais
- Identação para marcar blocos
- Poucas palavras-chave
- Uso de coletor de lixo

Por ser uma linguagem multiparadigma, Python suporta os conceitos de programação orientada a objetos, imperativa e funcional. Possui uma tipagem dinâmica e forte, facilitando a leitura e entendimento do código desenvolvido. Além disto, possui a capacidade de meta-programação, permitindo a criação de linguagens de domínio específico [17]. Sua biblioteca padrão é imensa definindo-a como uma *linguagem de baterias inclusas*, ou seja, sem adição de pacotes externos, tem a capacidade para desenvolver uma vasta variedade de projetos.

Atualmente, possui um desenvolvimento de modelo comunitário o que permite que as contribuições sejam constantes. Com isso, a documentação da linguagem torna-se abundante e existam inúmeros módulos para executar virtualmente a tarefa necessária. Por esse motivo a linguagem passa por constantes alterações, correções e melhorias aumentando a sua confiança.

Scikit-learn

Scikit-learn, *Sklearn*, é um *framework open-source* de aprendizagem de máquina escrito em Python que utiliza a plataforma dos pacotes Numpy/Scipy e Matplotlib. O pacote Numpy/Scipy é básico da linguagem e permite a utilização de arranjos, vetores, e matrizes de n dimensões de forma semelhante a utilizada na linguagem MATLAB. O pacote Matplotlib, porém, é utilizada na geração de gráficos 2D a partir de vetores [16]. A *Sklearn* fornece o estado da arte da implementação de muitos algoritmos bem conhecidos em aprendizagem de máquina, dispondo de uma interface simples. Difere de outros pacotes de aprendizagem de máquina por: ser distribuído com a licença BSD, incorporar código com eficiência comprovada e focar em programação imperativa. Deste framework foram utilizados os algoritmos: *Naive Bayes* e *Random Forest*.

LibSVM

LIBSVM é uma biblioteca para o algoritmo SVM. A biblioteca foi desenvolvida em meados de 2000 na Universidade Nacional de Taiwan, cujo principal objetivo era facilitar o uso e aumentar a eficiência para a classificação e regressão SVM. Apesar de ser aplicável a linguagens como JAVA, MATLAB e C, a LIBSVM, neste projeto, foi utilizada como uma biblioteca da linguagem Python [15].

4.2 Base de dados

As bases de dados utilizadas foram extraídas do projeto CASAS coordenado pela Dra. Diane J. Cook da *Washington State University*. O projeto CASAS trata as residências utilizadas na pesquisa como agentes inteligentes, no qual o estado dos residentes e os arredores são percebidos por meio de sensores e o ambiente atuado por controladores que proporcionam conforto, segurança, e produtividade na residência [5].

Dentre as residências disponibilizadas pelo projeto foram optadas por bases que existam apenas um morador. Tal escolha foi feita para garantir que as atividades que foram rotuladas pelo sistema não sejam correlacionadas com ações realizadas por outros moradores.

Figura 14: Layout das residências do CASAS [KRISHNAN; COOK. 2014. 10 p]

Antes de discutir os parâmetros utilizados no experimento e os resultados obtidos precisamos ter uma visão geral das bases utilizadas para o desenvolvimento deste projeto. A Tabela 1 detalha quais as atividades são executadas em cada base, bem como a suas respectivas frequências.

Tabela 1: Perfil das atividades desempenhadas nas Bases de dados

Atividades	HH106	HH110	HH123	HH124	H1130
Bathe	1491	1444	2394	0	249
Bed_Toilet_Transition	807	1677	2657	1020	2393
Cook	5401	0	560	0	285
Cook_Breakfast	9154	2379	6697	0	160
Cook_Dinner	16398	225	15952	216	0
Cook_Lunch	9458	0	2195	0	86
Dress	6845	8631	9987	500	5722
Eat	5119	0	93	0	387
Eat_Breakfast	6586	1685	2997	84	118
Eat_Dinner	6807	479	1672	125	0
Eat_Lunch	5139	0	400	0	24
Enter_Home	11628	8499	7421	2418	6787
Entertain_Guests	8152	0	1727	6624	46223
Evening_Meds	420	2799	2090	0	0
Groom	2737	3586	3897	463	5025
Leave_Home	20245	5103	10768	58561	4912
Morning_Meds	1057	2509	1490	0	0
Personal_Hygiene	31229	4109	6659	147	19369
Phone	4659	0	4144	146	6216
Read	8476	640	585	0	0
Relax	2463	18515	2779	0	1140
Sleep	4484	15449	14150	1612	14365
Sleep_Out_Of_Bed	421	4081	99	398	1841
Take_Medicine	3406	5609	0	0	0
Toilet	8105	18081	10527	2119	9968
Wash_Breakfast_Dishes	7368	2730	2188	152	0
Wash_Dinner_Dishes	4928	110	8348	32	0
Wash_Dishes	3331	0	9909	37	313
Wash_Lunch_Dishes	4671	0	1039	0	0
Watch_TV	22570	0	20546	364	26358
Work	1236	349	0	60	743
Work_At_Table	4681	12287	69	205	676
Work_On_Computer	33559	17246	0	438	5323

4.3 Parâmetros do Experimento

Naive Bayes

O algoritmo utilizado implementa a classificação Gaussiana do *Naive Bayes*. Nesta abordagem supomos que os dados se comportam conforme uma distribuição gaussiana.

Possui como única configuração as probabilidades a priori das classes. Para avaliar o comportamento do algoritmo em questão, foi utilizada a configuração padrão.

Support Vector Machine

A SVM, da biblioteca LIBSVM, possui múltiplos parâmetros de configuração. Nos nossos experimentos, escolhemos alterar apenas:

- Tipo SVM: C-SVC (Classificação de múltiplas classes)
- Tipo de Kernel: Base radial
- Heurística de encolhimento: 0 (Desativada)
- Tamanho da Cache: 500MB

A heurística de encolhimento reduz o tamanho do problema eliminando temporariamente as variáveis que provavelmente não serão selecionadas no conjunto de trabalho. Porém, dependendo do formato dos dados essa técnica pode piorar o tempo de execução. No caso do nosso problema, foi abordado as duas condições apresentando melhor desempenho aquela no qual a técnica de encolhimento não havia sido aplicada.

Random Forest

A *Random Forest*, por sua vez, apesar de apresentar, também, múltiplos parâmetros para configuração, sofreu poucas alterações. Apenas o parâmetro da quantidade de árvores foi alterado. Devido à dimensionalidade das bases e os atributos apresentarem independência entre si, no teste foram utilizadas apenas 10 árvores.

4.4 Análise dos dados

Comparamos os resultados obtidos entre os diferentes algoritmos a fim de estimar qual seria a melhor abordagem para o reconhecimento de atividades. Porém, antes de avaliar a performance para cada máquina de aprendizagem, se faz necessária definir qual o tamanho da janela que melhor se adequa as bases de dados.

Inicialmente realizamos os testes com os parâmetros anteriormente definidos. A janela de tamanho 10 fora utilizada a fim de avaliar o desempenho dos algoritmos escolhidos.

Tabela 2: Média e Desvio padrão da acurácia obtida para a *Naive Bayes*, *Support Vector Machine* e *Random Forest* para os testes iniciais cujo tamanho da janela era 10. Os melhores resultados estão em negrito.

Banco de Dados	**Naive Bayes(%)**	**Support Vector Machine(%)**	**Random Forest(%)**
hh106	4,64 (0,17)	36,34 (0,14)	**52,78 (0,02)**
hh110	5,00 (0,00)	42,50 (0,02)	**58,68 (0,03)**
hh123	5,16 (0,02)	32,41 (0,01)	**44,31 (0,02)**
hh124	13,11 (0,01)	84,69 (0,0)	**92,52 (0,01)**
hh1130	2,75 (0,07)	49,62 (0,02)	**59,23 (0,13)**
Média	6,13 (0,05)	49,17 (0,04)	**61,50 (0,04)**

A Figura 16 ilustra como a variação do tamanho da janela infere no desempenho obtido pelas máquinas de aprendizagem. Os testes foram efetuados em todas as bases de dados.

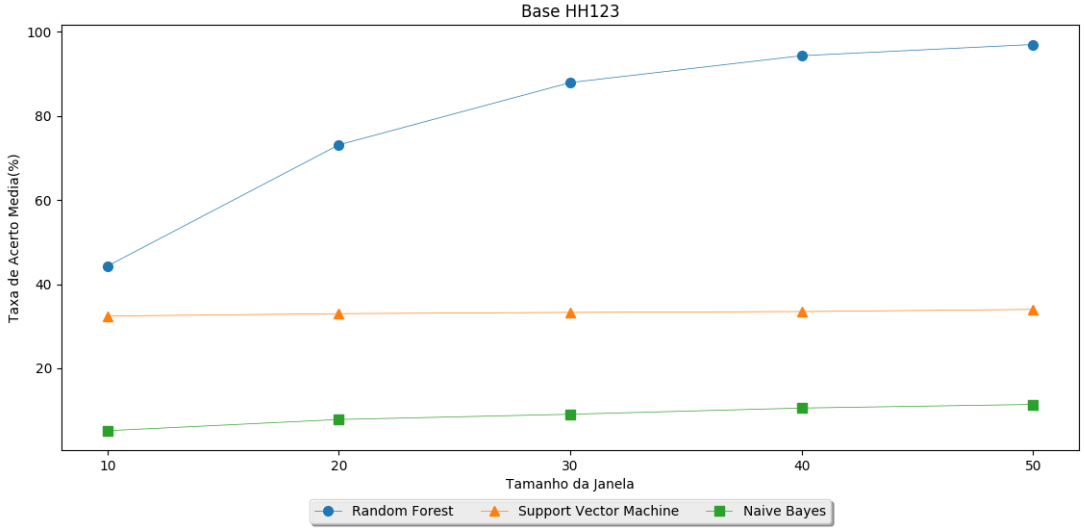


Figura 15: Variação do tamanho da janela deslizante utilizada para compôr o *Bag of sensors*

Para tal, nota-se que a dimensão 50 apresentou melhores resultados. Os resultados obtidos pela variação do tamanho da janela sobre as demais bases de dados encontram-se em Anexo.

De acordo com o tamanho da janela definido, vimos, conforme Tabela 2, que a *Random Forest* apresenta os melhores resultados para diferentes bases de dados segundo a nossa metodologia proposta. A SVM, por sua vez, apresenta valores consideráveis, uma vez que pouco dos seus parâmetros foi alterado. A *Naive Bayes*, porém, em diferentes bases de dados permaneceu com dificuldades para atingir a finalidade proposta.

Tabela 3: Média e Desvio padrão da acurácia obtida para a *Naive Bayes*, *Support Vector Machine* e *Random Forest* com janela de tamanho 50. Os melhores resultados estão em negrito.

Banco de Dados	**Naive Bayes(%)**	**Support Vector Machine(%)**	**Random Forest(%)**
hh106	7,32 (0,11)	39,12 (0,18)	**96,34 (0,08)**
hh110	9,31 (0,14)	46,3 (0,15)	**98,06 (0,10)**
hh123	11,4 (0,17)	33,99 (5,58)	**96,97 (0,15)**
hh124	23,46 (0,35)	85,06 (15,51)	**99,43 (0,06)**
hh1130	5,10 (0,09)	53,38 (19,30)	**96,64 (0,10)**
Média	11,32 (0,17)	51,57 (8,14)	**97,49 (0,10)**

O desvio padrão estimado nos dá o aporte para avaliar a configuração das estimativas. Além disto, as matrizes de confusão, utilizada para avaliar o desempenho das máquinas de aprendizagem por, encontram-se no Anexo. Com isso, vemos que a *Random Forest* ainda permanece como melhor abordagem.

Por fim, decidimos avaliar, para o tamanho de janela 50, diferentes quantidades de árvores utilizadas na *Random Forest* para o reconhecimento das atividades. Os valores definidos foram: $[10, 25, 50, 75, 100]$. A figura abaixo demonstra os resultados obtidos:

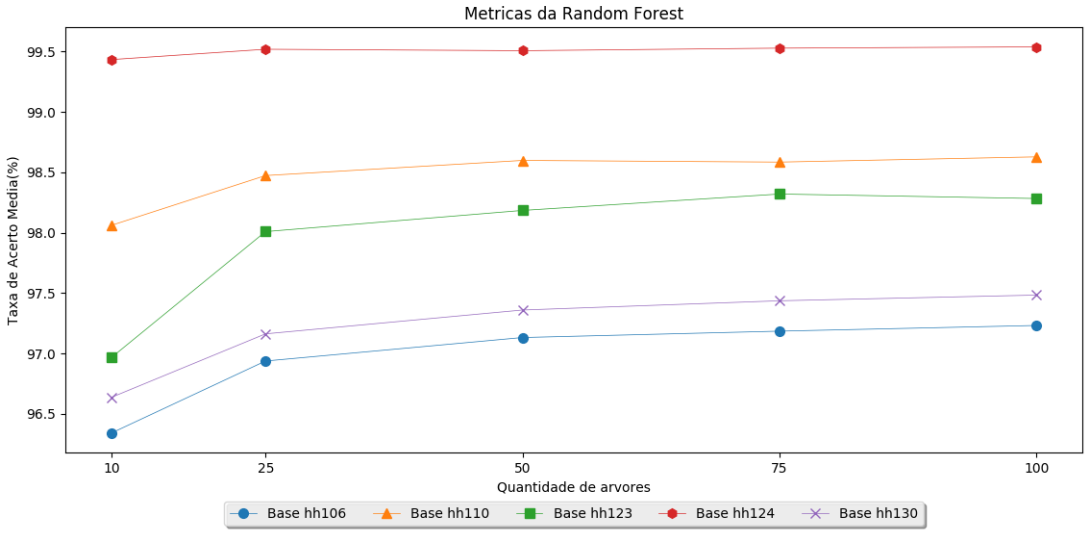


Figura 16: Variação da quantidade de árvores na *Random Forest*

Assim, vemos que há um aumento na acurácia quando alteramos a quantidade de árvore utilizadas pela *Random Forest*. Porém, este ganho se torna irrisório para valores superiores a 25 árvores. Logo, a configuração que apresenta-se ideal para o nosso cenário pode ser definida como:

Random Forest

- Quantidade de árvores: 25
- Tamanho da Janela: 50

5 CONSIDERAÇÕES FINAIS

O objetivo deste trabalho foi avaliar o desempenho de máquinas de aprendizagem no reconhecimento de atividades em casas inteligentes utilizando a metodologia proposta no artigo *Activity recognition on Streaming Sensor data* [1].

Para tal foram aplicados os dados processados das cinco bases de dados do projeto CASAS: hh106, hh110, hh123, hh124, hh130 nas máquinas de aprendizagem: *Naive Bayes*, *Support Vector Machine*, e *Random Forest*. Com isso, foram avaliadas a média e o desvio padrão para cada algoritmo, no qual a *Naive Bayes* apresentou uma média de 11%, a *Support Vector Machine* apresentou, aproximadamente, 52% e *Random Forest* com 97%.

Além disto, avaliamos para a *Random Forest*, a alteração da quantidade de árvores de decisão utilizadas. Para tal, valores superiores a 25 árvores apresentaram ganhos irrisórios.

Trabalhos como os realizados no projeto CASAS[5], House_n [6] e *The Aware Home* [7], apesar de apresentarem uma proposta de monitoramento de atividades, diferem do trabalho proposto quanto aos algoritmos utilizados para o reconhecimento ou, ainda, por técnicas de mineração de dados abordadas.

5.1 TRABALHOS FUTUROS

Como possíveis trabalhos futuros, pode-se apontar:

- Alterar a normalização dos dados, utilizando uma abordagem mais estatística, tal como a Distância de Mahalanobis;
- Considerar a correlação entre os sensores localizados na casa;
- Utilizar técnicas de combinação de classificadores a fim de otimizar o desempenho obtido.

6 REFERÊNCIAS

[1] Krishnan, Narayanan C. ; Cook, Diane J. **Activity recognition on streaming sensor data**. 2014. 10 f. Pervasive and Mobile Computing.

[2] Sánchez, Dairazalia; Tentori, Monica; Favela, Jesús. **Activity Recognition for the Smart Hospital**. 2008. 8 f. IEEE Intelligent Systems.

[3] Cook, Diane J. **Learning Setting Generalized Activity Models for Smart Spaces**. 2012. 7 f. IEEE Intelligent Systems.

[4] Krishnan, Narayanan C.; Cook, Diane J. **Activity Learning: Discovering, Recognizing, and Predicting Human Behavior from Sensor Data**. Vol 1. 2015, Wiley. pp. 282.

[5] **CASAS: Center for Advanced Studies in Adaptive Systems**. Disponível em: http://casas.wsu.edu/

[6] **House n**. Disponível em: http://web.mit.edu/cron/group/housen/

[7] **Aware Home Research Initiative**. Disponível em: http://www.awarehome.gatech.edu/

[8] Mitchell, Tom Michael. **Machine Learning**. 1997, McGraw-Hill. pp 414.

[9] LORENA, A. C.; CARVALHO, A. C. P. L. de. **Introdução aos Classificadores de Margens Largas**. 2003, São Carlos - SP.

[10] SMOLA, A. J. et al. **Advances in Large Margin Classifiers**. [S.l.]: Morgan-Kauffman, 1999.

[11] HEARST, M. A. et al. **Support vector machines**. IEEE Intelligent Systems, IEEE Computer Society, Los Alamitos, CA, USA, v. 13, n. 4, p. 18–28, 1998. ISSN 1094-7167.

[12] L. Breiman, **Random forest**, Machine Learning, 2001, no. 45, pp. 5–32.

[13] Duda, Richard O.; Hart, Peter E.; Stork, David G. **Pattern Classification**. Second Edition. 2001, Wiley. pp 654.

[14] Smola, Alex; Vishwanathan, S.V.N. **Introduction to Machine Learning**. 2008, Cambridge University Press. pp 234.

[15] **LIBSVM: A Library for Support Vector Machines**. Disponível em: https://www.csie.ntu.edu.tw/ cjlin/libsvm/

[16] **Scikit-learn: Machine learning in Python**. Disponível em: http://scikit-learn.org/stable/

[17] **Python Brasil**. Disponível em: http://python.org.br/

[18] **IoT: Coletando Dados**. Disponível em: http://www.decom.ufop.br/imobilis/iot-coletando-dados/

[19] Xia, Feng; Yang, Laurence T; Wang, Lizhe Wang; Vinel, Alexey. **Internet of Things**. International Journal of Communication. 2012. 2 p.

[20] Camilo, Cássio Oliveira; Silva, João Carlos da; **Mineração de Dados: Conceitos, Tarefas, Métodos e Ferramentas**. 2009.

ANEXOS

Devido ao grande número de atividades desempenhadas nas bases e a fim de exibir os resultados na matriz de confusão, optou-se por representá-las em formato numérico de $[0, 1, \cdots, m-1]$, no qual m é a quantidade de atividades distintas que ocorreram na casa. Para melhor compreensão temos que:

Tabela 4: Atividades Base hh106

Enter Home	0	Read	9	Toilet	18	Bathe	27
Personal Hygiene	1	Phone	10	Groom	19	Work	28
Wash Lunch Dishes	2	Evening Meds	11	Sleep out of bed	20	Entertain Guests	29
Leave Home	3	Eat Breakfast	12	Work at table	21	Sleep	30
Cook Dinner	4	Watch TV	13	Morning Meds	22	Work on Computer	31
Eat Dinner	5	Cook	14	Cook Breakfast	23	Dress	32
Cook Lunch	6	Wash Breakfast Dishes	15	Take Medicine	24		
Eat Lunch	7	Eat	16	Bed Toilet Transition	25		
Relax	8	Wash Dishes	17	Wash Dinner Dishes	26		

Tabela 5: Atividades Base hh110

Enter Home	0	Read	7	Groom	14	Toilet	21
Personal Hygiene	1	Drink	8	Sleep out of bed	15	Sleep	22
Leave Home	2	Evening Meds	9	Work at table	16	Work on computer	23
Cook Dinner	3	Eat Breakfast	10	Morning Meds	17	Dress	24
Eat Dinner	4	Wash Breakfast Dishes	11	Cook Breakfast	18		
Wash Dinner Dishes	5	Work	12	Take Medicine	19		
Relax	6	Bathe	13	Bed Toilet Transition	20		

Tabela 6: Atividades Base hh123

Enter Home	0	Relax	8	Eat	16	Bed Toilet Transition	24
Personal Hygiene	1	Read	9	Wash Dishes	17	Wash Dinner Dishes	25
Eat Lunch	2	Phone	10	Bathe	18	Toilet	26
Leave Home	3	Evening Meds	11	Groom	19	Entertain Guests	27
Cook Dinner	4	Eat Breakfast	12	Sleep out of Bed	20	Sleep	28
Eat Dinner	5	Watch TV	13	Work at Table	21	Dress	29
Cook Lunch	6	Cook	14	Morning Meds	22		
Wash Lunch Dishes	7	Wash Breakfast Dishes	15	Cook Breakfast	23		

Tabela 7: Atividades Base hh124

Enter Home	0	Wash Dinner Dishes	6	Work	12	Toilet	18
Personal Hygiene	1	Started	7	Wash Dishes	13	Entertain Guests	19
Other	2	Phone	8	Groom	14	Sleep	20
Leave Home	3	Eat Breakfast	9	Sleep out of Bed	15	Work on Computer	21
Cook Dinner	4	Watch TV	10	Work at Table	16	Dress	22
Eat Dinner	5	Wash Breakfast Dishes	11	Bed Toilet Transition	17	Work on Computer	23

Tabela 8: Atividades Base hh130

Enter Home	0	Eat Breakfast	7	Sleep out of Bed	14	Entertain Guests	21
Personal Hygiene	1	Watch TV	8	Work at Table	15	Sleep	22
Eat Lunch	2	Cook	9	Cook Breakfast	16	Work on Computer	23
Leave Home	3	Eat	10	Bed Toilet Transition	17	Dres	24
Cook Lunch	4	Groom	11	Bathe	18		
Relax	5	Toilet	12	Work	19		
Phone	6	Wash Dishes	13	Steep out	20		

A matriz de confusão de uma hipótese oferece uma medida efetiva do modelo de classificação, pois apresenta a proporção de classificação correta versus a classificação prevista pela máquina de aprendizagem. Quanto maior o percentual de acerto, maior representatividade é dada a diagonal da matriz de confusão. Assim, as figuras a seguir

ilustram o desempenho obtido pelas diferentes máquinas de aprendizagem sob as mesmas bases de dados.

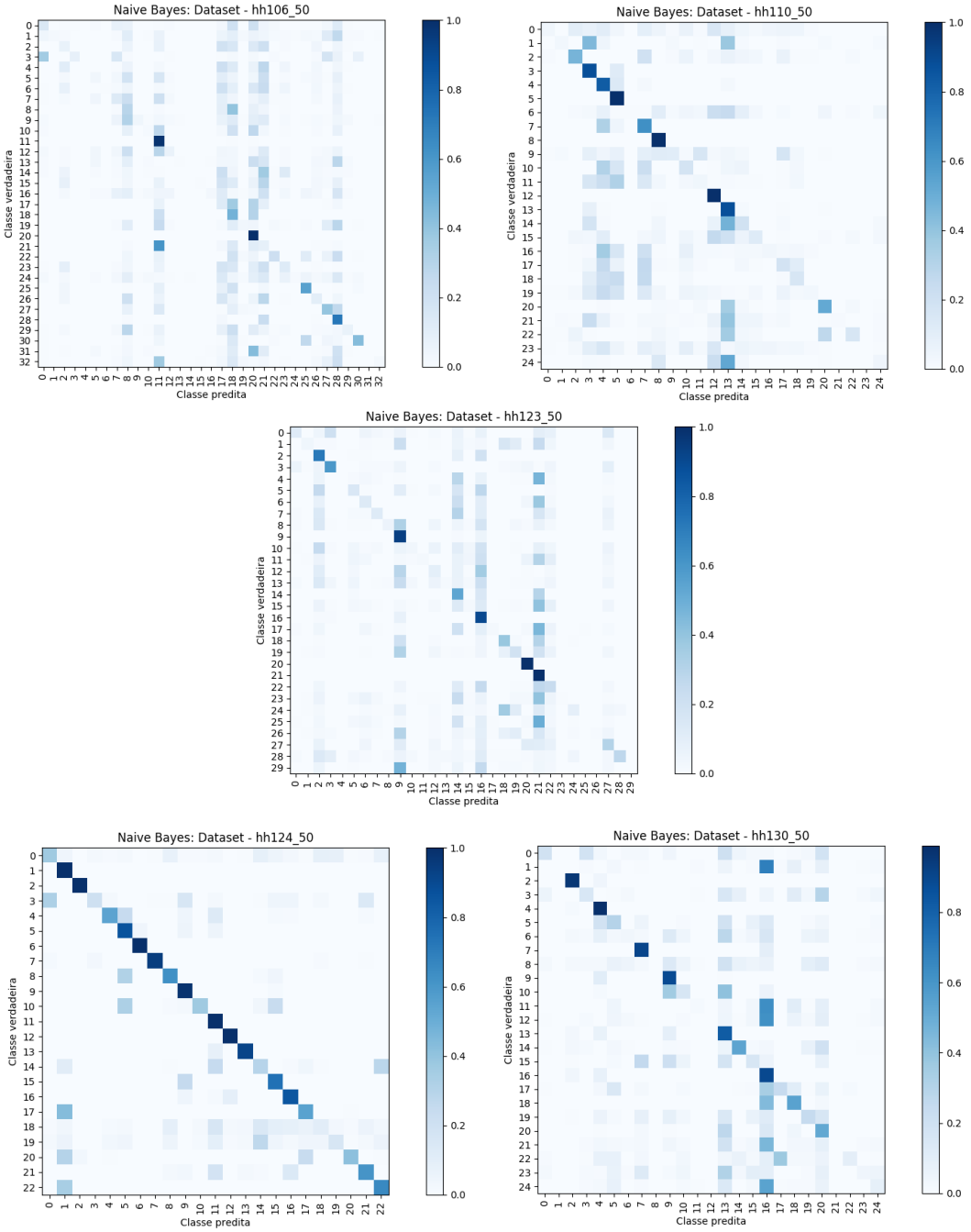


Figura 17: Matriz de Confusão: Naive Bayes

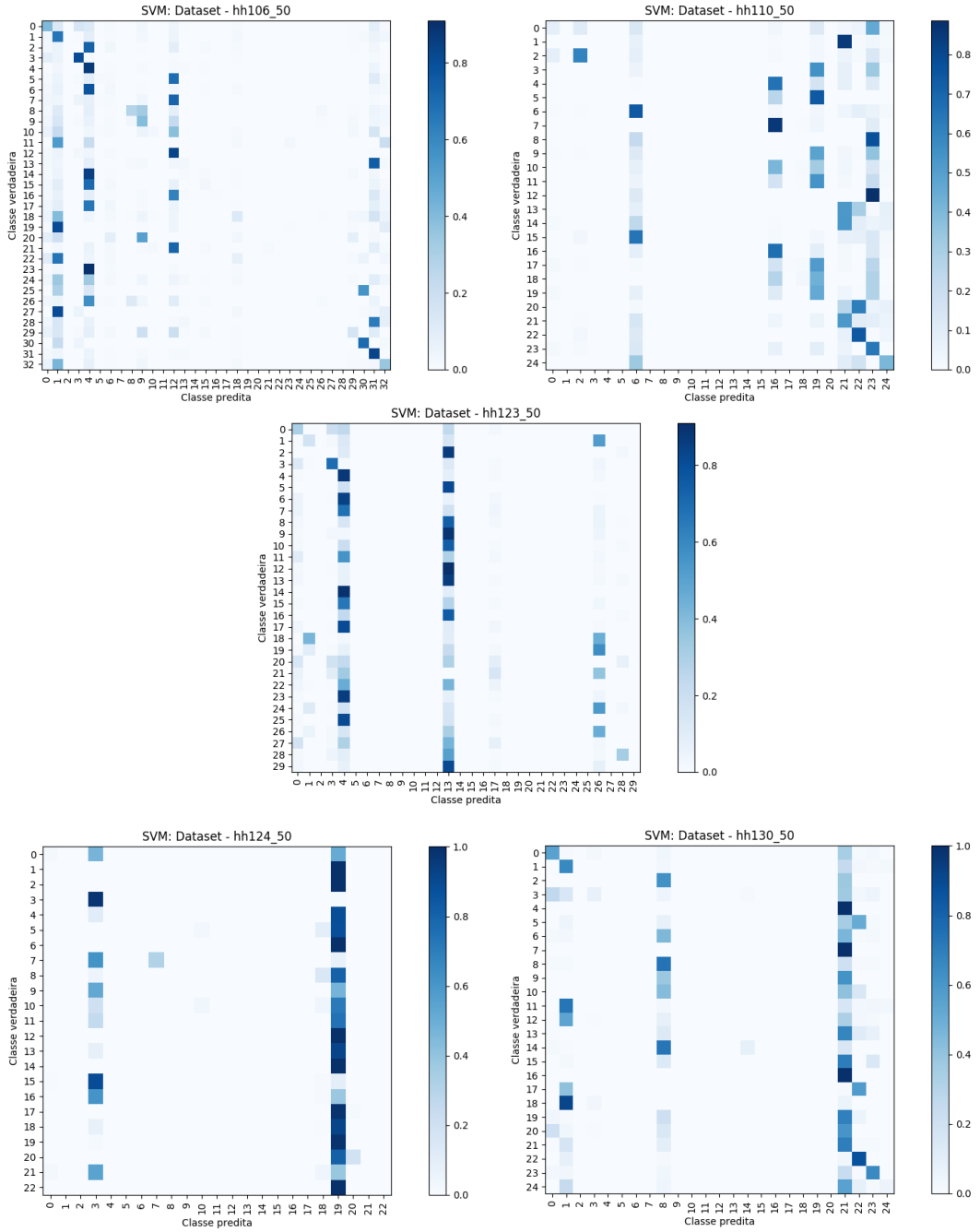


Figura 18: Matriz de Confusão: Support Vector Machine

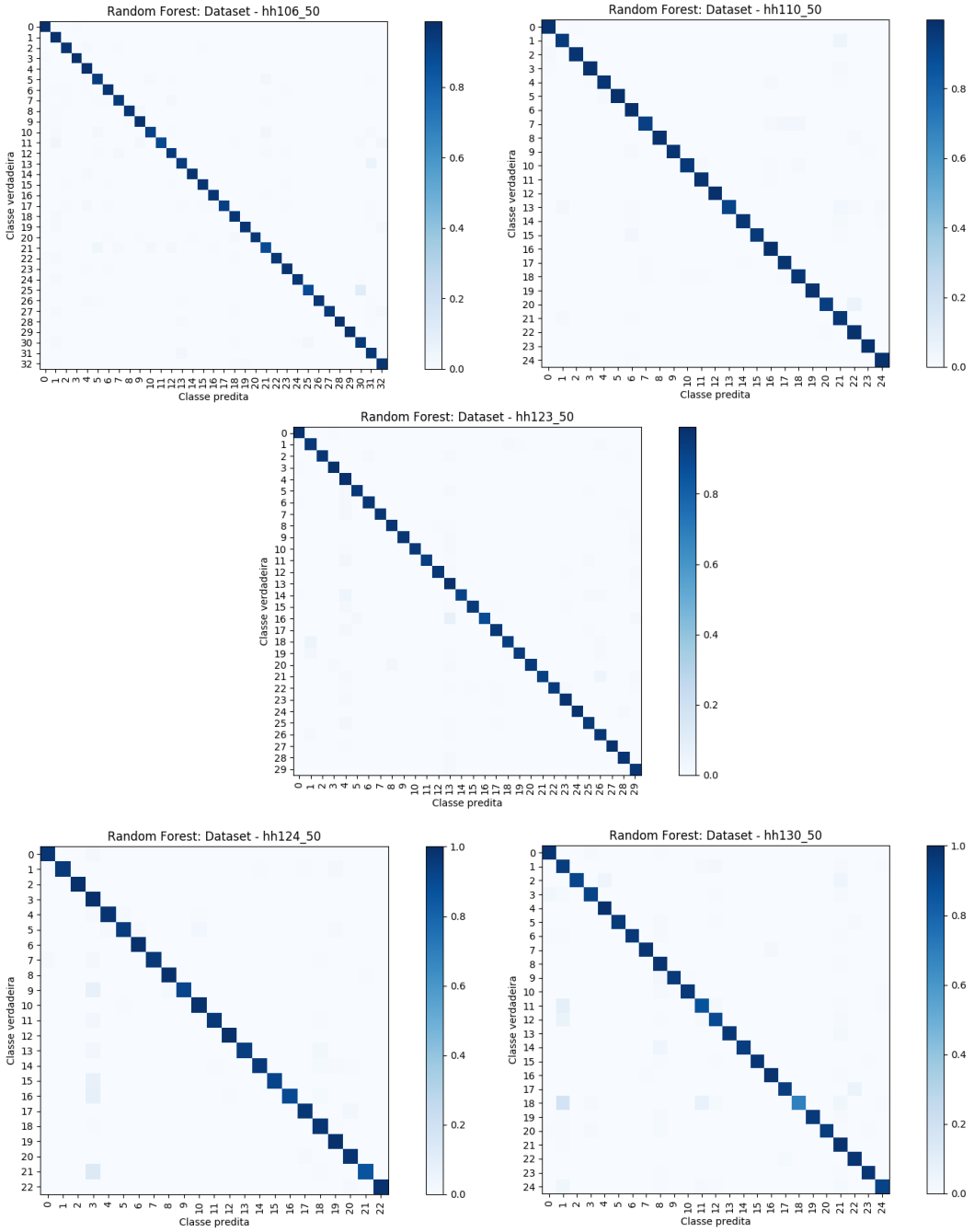


Figura 19: Matriz de Confusão: Random Forest

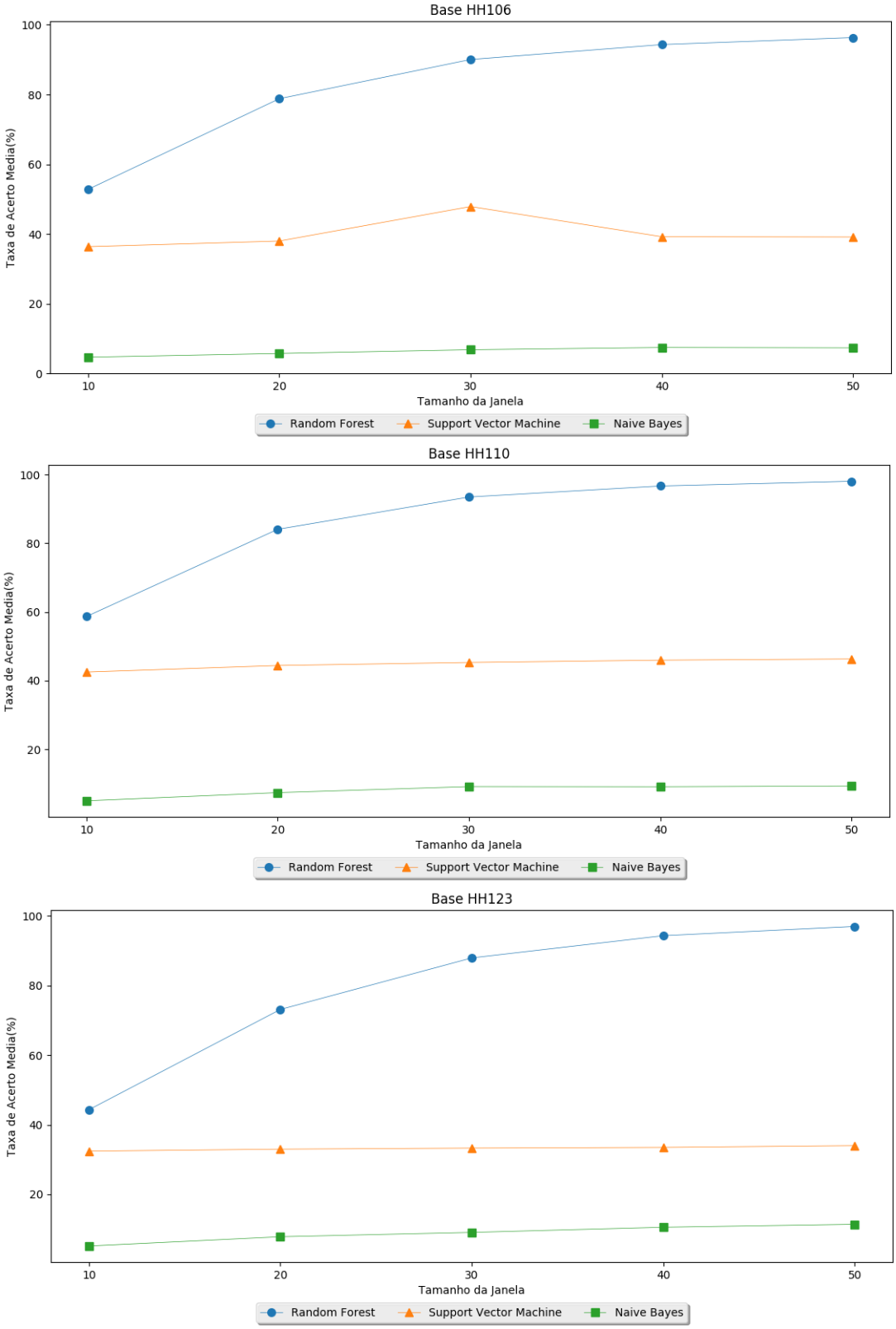


Figura 20: Variação do Tamanho da Janela[1]

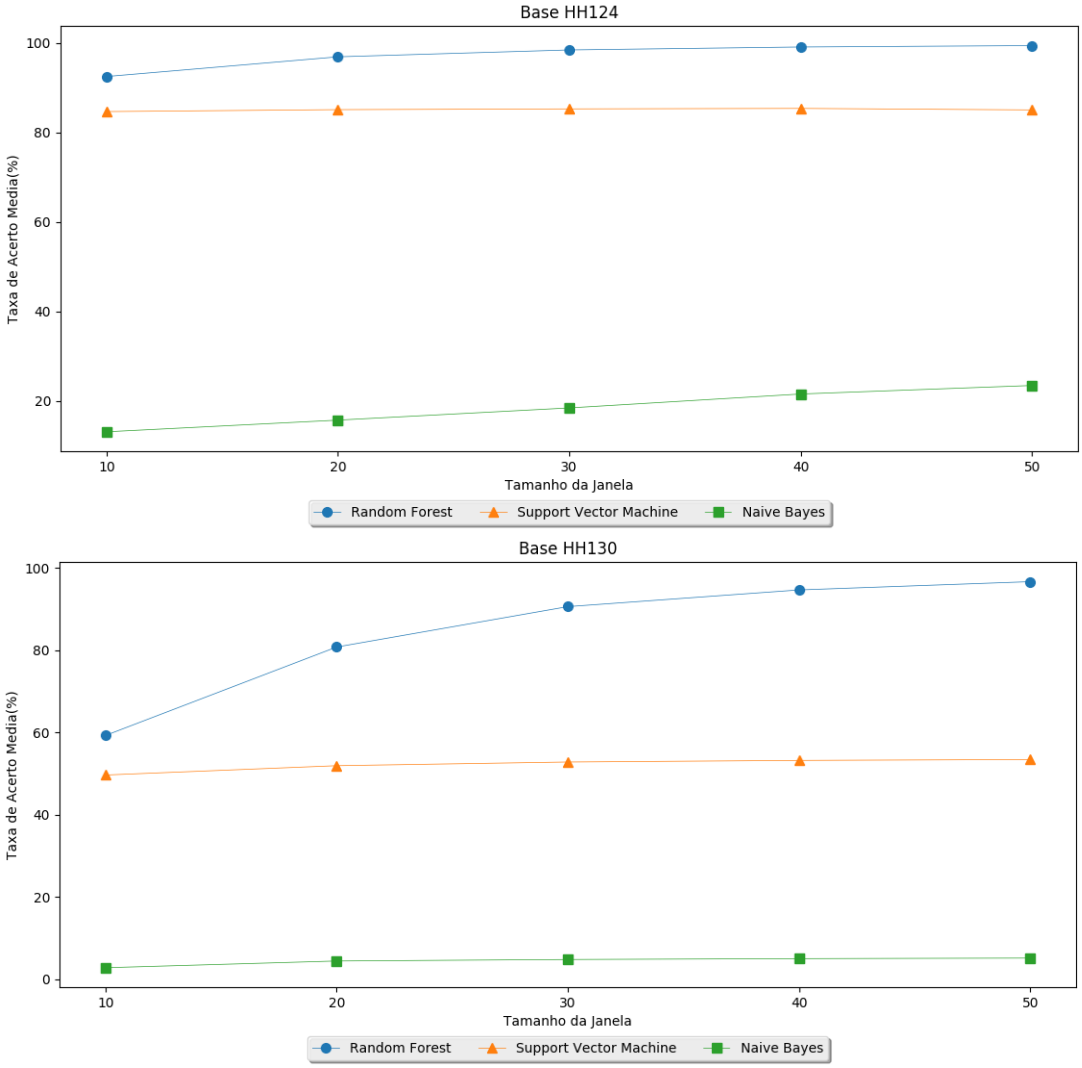


Figura 21: Variaçãp do Tamanho da Janela[2]