UNIVERSIDADE FEDERAL DE MINAS GERAIS
Escola de Engenharia
Programa de Pós Graduação em Engenharia Elétrica

Igor Pereira Gomes

Previsão de Eventos em Ambientes Inteligentes com Extração de Características Sequenciais e Localização de Usuários por Modelos Ocultos de Markov

Belo Horizonte

2019

Igor Pereira Gomes

Previsão de Eventos em Ambientes Inteligentes com Extração de Características Sequenciais e Localização de Usuários por Modelos Ocultos de Markov

Dissertação de Mestrado apresentada à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais, como requisito para obtenção do Título de Mestre em Engenharia Elétrica.

Orientador: Prof. Antônio de Pádua Braga

Belo Horizonte

2019

Gomes, Igor Pereira.
G633p Previsão de eventos em ambientes inteligentes com extração de características sequenciais e localização de usuários por modelos ocultos do Markov [recurso eletrônico] / Igor Pereira Gomes. - 2019.
1 recurso online (54 f. : il., color.) : pdf.

Orientador: Antônio de Pádua Braga.

Dissertação (mestrado) - Universidade Federal de Minas Gerais, Escola de Engenharia.

Bibliografia: f. 51-54.
Exigências do sistema: Adobe Acrobat Reader.

1. Engenharia elétrica - Teses. 2. Automação residencial - Teses. 3. Modelo escondido de Markov - Teses. I. Braga, Antônio de Pádua. II. Universidade Federal de Minas Gerais. Escola de Engenharia. III. Título.

CDU: 621.3(043)

Ficha catalográfica elaborada pelo bibliotecário Reginaldo César Vital dos Santos CRB/6 2165
Biblioteca Prof. Mário Werneck, Escola de Engenharia da UFMG

"Previsão de Eventos em Ambientes Inteligentes com Extração de Características Sequenciais e Localização de Usuários por Modelos Ocultos de Markov"

Igor Pereira Gomes

Dissertação de Mestrado submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais, como requisito para obtenção do grau de Mestre em Engenharia Elétrica.

Aprovada em 13 de fevereiro de 2019.

Por:

Prof. Dr. Antônio de Pádua Braga
DELT (UFMG) - Orientador

Prof. Dr. Cristiano Leite de Castro
DEE (UFMG)

Prof. Dr. Carmela Maria Polito Braga
DELT (UFMG)

Este trabalho é dedicado à Escola de Engenharia da UFMG
e a todos que lutam por sua excelência.

Agradecimentos

Agradeço enormemente ao professor Antônio de Pádua Braga pelo apoio, orientação e auxílio que me foram preciosos. Também essencial foi apoio dos professores Cristiano Leite de Castro e Hani Camille Yehia, da UFMG, e de Marco Túlio Sousa e Jullierme Dias, da empresa Neocontrol. O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001.

"Casa não é o lugar onde você vive,
e sim, o lugar onde te entendem.
(Christian Morgenstern)

Resumo

Este trabalho apresenta estudos realizados sobre os problemas de previsão de eventos e de localização de usuários em ambientes inteligentes, tendo como base o sistema de automação residencial *Minibox*, da empresa brasileira *Neocontrol*. É proposta e validada uma arquitetura para um sistema dedicado a resolver tais problemas, tendo como entrada uma sequência de ativações e mudanças de estado de sensores e outros dispositivos do ambiente. O problema de previsão de eventos é formulado como um problema de classificação, sendo apresentada uma nova abordagem (AM-CHIP-CLAS) para classificação sem parâmetros de amostras pequenas ou desbalanceadas, baseada no Classificador por Arestas de Suporte (CLAS) e em Aprendizado de Métrica. Também são apresentados três métodos de seleção de características para sequências temporais discretas, com adaptações para serem utilizados no problema, de forma a alimentar o classificador. Para solução do problema de localização de usuários, é desenvolvido um Modelo Oculto de Markov Fatorial (FHMM) cujos estados ocultos representam a localização de cada residente e cujas emissões representam as observações de sensores e dispositivos. Os parâmetros de transição e emissão são estimados a priori através de regras simples baseadas na planta da casa e na localização dos sensores. Nos experimentos feitos para o problema da previsão de eventos, verifica-se bons resultados para os métodos implementados para seleção de características, utilizando como base o classificador SVM. Para os algoritmos de classificação baseados no CLAS, apesar do desempenho equivalente ou superior ao das SVMs em *benchmarks* de classificação, estes mostram desempenho inferior aos das SVMs para os dados do banco de testes devido ao grande desbalanceamento e pequena quantidade de amostras das classes minoritárias. Ainda assim, verifica-se que o aprendizado de métrica melhora consistentemente o desempenho do Classificador por Arestas de Suporte nestes dados. Para o modelo FHMM para localização de usuários, os resultados obtidos foram validados através de inspeção manual com dados de casas inteligentes presentes na literatura e os resultados foram coerentes com as anotações dos dados.

Palavras-chaves: Smart Home. Previsão de Eventos. Reconhecimento de Padrões Sequenciais. Extração de Características Sequenciais. Classificador por Arestas de Suporte. Grafos de Gabriel. Classificadores sem Parâmetros. Aprendizado de Métrica. Modelo Oculto de Markov.

Abstract

This thesis presents studies done on the Event Prediction problem and the unsupervised User Tracking problem in smart environments, using the data collected by the *Minibox* home automation system, developed by the Brazilian company *Neocontrol*. An architecture for solving those problems is proposed and validated, with a sequence of device state changes as its input. The event prediction problem is formulated as a supervised classification problem. A new approach for small and imbalanced sample classification without hyperparameters is proposed, based on Support Edge Classifiers (CHIP-CLAS) and metric learning. Three sequential feature extraction methods were adapted and validated for the event prediction problem, based on the presented smart home data, as a preprocessing step for the classification. For the user tracking problem, a Factorial Hidden Markov Model is employed with its hidden states representing the location for the individuals and its emissions representing the observed device activations. Its parameters are estimated *a priori* through simple rules based on the floor plan and location for the devices. In the experiments done for the event prediction problem, good results were obtained with the SVM classifier using all three feature extraction methods. With the CLAS classifiers, although results were equivalent to SVMs for a benchmark consisting on 15 different datasets for both CHIP-CLAS and AM-CHIP-CLAS, the observed performance for the event prediction data was far behind, due to small sample size and imbalanced data. Still, the metric learning step proposed for AM-CHIP-CLAS significantly improved the performance comparing to CHIP-CLAS. For the user tracking FHMM model, validation was done through manual inspection with smart home data from the literature and results were consistent with data annotation.

Key-words: Smart Home. Event Prediction. Sequence Pattern Recognition. Sequence Feature Extraction. Support Edge Classifier. Gabriel Graphs. Classification without Hyperparameters. Metric Learning. Hidden Markov Model

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

SVM	Máquina de Vetor de Suporte
MQTT	Message Queue Telemetry Transport
CLAS	Classificador por Aresta de Suporte
RBF	Função de Base Radial
HMM	Modelo Oculto de Markov
FHMM	Modelo Oculto de Markov Fatorial

Sumário

1 Introdução

1.1 Motivação

Em 1998, já eram delimitados os problemas a serem resolvidos para o desenvolvimento de um sistema inteligente e adaptativo integrado com residências, com previsões de que não demorariam a surgir utensílios domésticos equipados com processadores que se comunicariam entre si e tomariam decisões, como por exemplo, uma lavadora de louças que se comunica com o aquecedor de água ou aparelhos de entretenimento que reagissem à presença do morador (MOZER, 1998). Mesmo antes, uma proto-inteligência já estava sendo concebida, com sistemas de controle de energia para utensílios domésticos já sendo desenvolvidos por pesquisadores (HUNT; HOLMES, 1986) e até mesmo presentes em patentes (CARR et al., 1987).

Na atualidade, as barreiras tecnológicas para a concepção e implementação de *Smart Homes* já tem sido rompidas por diversas universidades, com diversos experimentos de sucesso. No contexto brasileiro, porém, ainda resta para ser rompida a barreira que separa academia e indústria, impedindo a adoção de sistemas inteligentes em larga escala nas residências do país.

Empresas de Automação Residencial já possuem tecnologias integradas a residências e outros ambientes que geram *streaming* de dados oriundos de dispositivos automatizados e sensores. O presente trabalho visa explorar alternativas de para inserção de inteligência a tais tecnologias já existentes na indústria nacional. Para isso, efetuou-se a parceria com a companhia *Neocontrol*, de automação residencial, para delimitar problemas e criar soluções para implementação de ambientes adaptativos utilizando as tecnologias da companhia, de forma a viabilizar, em pouco tempo, a criação de um produto acessível para este propósito.

1.2 Objetivos

Este trabalho visa explorar os dados gerados pelo sistema de automação residencial *Minibox*, da companhia *Neocontrol*, e propor arquiteturas para sistemas inteligentes baseados no mesmo. Endereçam-se os problemas da Previsão de Eventos e da Localização de Usuários em ambientes inteligentes.

A necessidade de se produzir e instalar os sistemas em série com custos razoáveis leva a uma série de restrições tecnológicas e outras dificuldades, discutidas e validadas com a *Neocontrol* como sendo barreiras na implementação de sistemas inteligentes baseados

em seus sistemas atuais de automação. São as principais:

- Alto custo de implementação de uma malha de sensores de tamanho considerável.
- Dificuldade na obtenção de dados anotados para reconhecimento de atividades e eventos.

Neste trabalho, são explorados métodos encontrados na literatura e desenvolvidos novos métodos para a solução dos problemas endereçados (Previsão de Eventos e Localização de Usuários), a partir dos sistemas de automação já existentes e considerando as dificuldades tecnológicas listadas.

1.3 Contribuições

Este trabalho apresenta as seguintes contribuições ao campo de pesquisa e à indústria:

- Um estudo exploratório de viabilidade e desempenho de diversos métodos presentes na literatura para extração de características de sequências temporais de dados categóricos aplicados à previsão de eventos através de *streaming* de dados de ambientes inteligentes, com dados publicados na literatura e dados disponibilizados pela companhia *Neocontrol*.
- Abordagens para tais métodos de seleção de características que levam em conta o custo computacional dos mesmos e validação das abordagens através de experimentos, utilizando Máquinas de Vetor de Suporte como classificador.
- Uma nova abordagem envolvendo aprendizado de métrica para os classificadores CHIP-CLAS visando classificação supervisionada sem hiperparâmetros de conjuntos desbalanceados e/ou com pouca representação da classe minoritária, com estudos sobre seu desempenho em dados presentes na literatura e para o problema tratado neste trabalho.
- Uma proposta de modelo estocástico baseado em estados para localização não-supervisionada de usuários através de dados sequenciais de sensores, com testes para validação através de inspeção.
- O artigo: Gomes, I.P.; Bambirra,L.C.; Braga, A.P.; Aprendizado de Métrica Supervisionado para Classificador por Arestas de Suporte. Publicado no *XIII Congresso Brasileiro de Inteligência Computacional*.

1.4 Estrutura do Trabalho

O trabalho é organizado em sete seções, descritas a seguir:

- Introdução: Explicita a motivação, os objetivos e a contribuição que este trabalho apresenta à área da automação residencial e da inteligência computacional.
- Contextualização: Mostra o estado da arte na área, enumerando trabalhos anteriores, suas contribuições e como estes se relacionam com o presente trabalho.
- Fundamentação Teórica: Mostra os fundamentos teóricos sobre o qual este trabalho é construído e que são necessários para sua compreensão.
- Metodologia: Descreve os métodos desenvolvidos e as abordagens adotadas para solução dos problemas apresentados.
- Experimentos e Resultados: Descreve os experimentos realizados para validação e comparação dos métodos desenvolvidos e propostos, apresenta os resultados dos experimentos e os discute.
- Conclusões e Trabalhos Futuros: Sumariza e discute os resultados obtidos com o trabalho e apresenta propostas para trabalhos futuros.

2 Contextualização

O trabalho se insere no contexto de previsão de atividades e localização (*tracking*) de usuários em ambientes inteligentes.

Como são utilizados apenas sensores simples, comuns e não-invasivos para o sistema criado pela *Neocontrol*, considera-se para o contexto apenas projetos onde são empregados sensores com essas características. Não se insere no contexto, portanto, o reconhecimento de atividades utilizando aparatos como câmeras, *wereables* e informações de sensores inerciais de *smartphones*.

O problema do reconhecimento de atividades, apesar de bastante presente na literatura, faz extenso uso de dados anotados, ou seja, dados cujas atividades a qual pertencem, iniciam ou terminam são registradas (KASTEREN et al., 2008). Como os dados utilizados neste trabalho não possuem anotações, este problema não será endereçado.

2.1 Bases de Dados de *Smart Homes*

Conjuntos de dados originados de sensores simples em *Smart Homes*, incluindo detalhes sobre os dispositivos utilizados na coleta dos dados, estão amplamente disponíveis na literatura. São citados alguns exemplos a seguir, com descrições dos dados que estes contém.

Alguns conjuntos de dados contém longos períodos da vida cotidiana, com ou sem anotações (especificação de início ou fim de atividades específicas, vinculado aos seus correspondentes acionamentos de sensores). Em (COOK, 2010) e (COOK; SCHMITTER-EDGECOMBE, 2009) são gerados dados com anotações, correspondentes a uma residência com apenas um morador. Anotações nos dados são úteis para métodos supervisionados de reconhecimento de atividades e previsão de eventos, além de também servirem para validação de métodos não-supervisionados.

Tratando de residências com vários residentes, também encontramos conjuntos de dados com anotações, como os utilizados em (COOK; SCHMITTER-EDGECOMBE, 2009) e (SZEWCYZK et al., 2009), ambos correspondentes a residências com dois moradores.

Em (SZEWCYZK et al., 2009), são discutidos também os métodos utilizados para produzir anotações para os dados, elencando vantagens e desvantagens para cada um. Métodos onde o próprio morador registra o momento de início e fim de suas próprias atividades podem ser facilmente implementados, mas geram grande incômodo ao morador. Métodos em que sensores específicos são instalados em determinados equipamentos para

detectar seu uso não geram incômodo, mas tais sensores são custosos, dificultando a implementação em série. Métodos em que a anotação é feita através de visualização e rotulação manual não possuem este custo com equipamento, mas exigem uma elevada carga de tempo dos pesquisadores para visualização e rotulação.

Alguns conjuntos de dados focam no reconhecimento de atividades específicas, com determinadas atividades sendo repetidas e registradas, fora do cotidiano. Exemplos são os dados utilizados em (SINGLA; COOK; SCHMITTER-EDGECOMBE, 2010) e (KASTEREN et al., 2008).

2.1.1 Representação dos Dados

Os dados oriundos de sensores em ambientes inteligentes são normalmente apresentados como uma ou mais sequências de acionamentos de sensores ou dispositivos, sendo cada acionamento descrito por uma *timestamp* com data e hora, o código identificador do sensor e o novo estado do sensor. Para os dados com anotações, também é incluso para cada acionamento se este representa o início ou o fim de alguma atividade. A título de exemplo, mostram-se na Figura 1 as amostras iniciais do conjunto de dados utilizado em (COOK; SCHMITTER-EDGECOMBE, 2009). Também é disponibilizada a planta do ambiente de onde foram coletados os dados, sendo mostrada a localização de cada sensor.

id	timestamp	sensor	smpval	annot	status
1	1251072000	M046	OFF		
2	1251072001	M048	ON		
3	1251072019	M050	ON	R1_Wandering_in_room	begin
4	1251072019	M044	ON		
5	1251072021	M046	ON		
6	1251072021	M044	OFF		
7	1251072022	M050	OFF		
8	1251072025	M046	OFF		
9	1251072041	M044	ON		
10	1251072043	M044	OFF		

Figura 1 – Visualização no terminal do PostgreSQL das amostras iniciais de conjunto de dados de múltiplos residentes com anotações.
Fonte: (COOK; SCHMITTER-EDGECOMBE, 2009), adaptado.

2.2 Previsão de Eventos em Residências Inteligentes

Em (TAX, 2018), são apresentados três problemas envolvendo a previsão de atividades em *Smart Homes*: Previsão da próxima atividade a ser executada, tempo até a próxima atividade e sequência das N próximas atividades executadas.

Para a solução destes problemas, o artigo (TAX, 2018) foca em técnicas baseadas em Redes Neurais Artificiais, como Redes Neurais Recorrentes (RNNs), Long Short-Term Memory (LSTM) e Gated Recurrent Units (GRUs). Para comparação de resultados, também são avaliadas outras técnicas para predição de sequências discretas. As técnicas *Prediction by Partial Matching* (PPM) (CLEARY; WITTEN, 1984), *Dependency Graph* (DG) (PADMANABHAN; MOGUL, 1996) e *All k-order Markov Chains* (AKOM) (PITKOW; PIROLLI, 1999) baseiam-se em Modelos de Markov, interpretando as sequências temporais como oriundas de um sistema baseados em estados. Também se testa o algoritmo LZ78 (ZIV; LEMPEL, 1978), um conhecido algoritmo de compressão de sequências através de codificação utilizando dicionário, representando sequências que se repetem com mais frequência por códigos de menor tamanho. Testam-se os algoritmos mencionados para todos os três problemas. Os resultados indicaram desempenho superior para as LSTMs no problema de previsão da próxima atividade e do tempo até a próxima atividade. Porém, as LSTMs não apresentam desempenho consistentemente superior ao se tentar prever uma sequência de atividades, sendo inferior a uma ou mais técnicas baseadas em modelos de Markov em parte dos casos.

Em (DAS et al., 2002), é discutida a importância do problema da previsão de atividades em *Smart Homes* e são descritos dois algoritmos para realizar esta atividade: *LeZi Update* e SHIP (*Smart Home Inhabitant Prediction*). Vê-se no artigo que a previsão de eventos é necessária para automação de rotinas e tarefas repetitivas para o usuário e que a previsão de sua próxima localização auxilia o ambiente a localiza-lo quando o sistema necessita entrar em contato com o mesmo. Para este problema, o artigo apresenta o algoritmo *LeZi Update*, baseado no algoritmo de compressão LZ78 (ZIV; LEMPEL, 1978), e SHIP (*Smart Home Inhabitant Prediction*), que funciona através de comparação com um dicionário de sequências, priorizando-se a correspondência de sequências maiores e/ou mais frequentes.

Em (JAKKULA; COOK, 2007), são identificadas relações temporais entre eventos, por exemplo, quando eventos sempre acontecem em sequência, um evento acontece durante outro, um evento acontece sempre ao início ou ao fim de outro, dois eventos acontecem sempre simultaneamente, entre outras relações. Tais relações podem ser, então, utilizadas para previsão.

2.3 Localização de Usuários

O processo de localização de usuários, apesar de trivial no caso de apenas um residente e conhecendo-se a localização dos sensores, se torna um desafio e uma necessidade no caso de mais de um residente. Nestes casos, a localização possibilita a separação de uma sequência única de ativações equivalente a todos os usuários em diferentes sequências

correspondentes ao comportamento individual de cada um (CRANDALL; COOK, 2013), facilitando as tarefas do reconhecimento de atividades e da identificação de usuário.

O artigo (CRANDALL; COOK, 2013) apresenta ainda dois métodos para realizar o tracking. O método mais simples, *GR/ED*, é baseado em um grafo onde dois sensores são interconectados somente se são próximos entre si, possíveis de se haver acionamento consecutivo de ambos por apenas uma pessoa. Baseado neste grafo, criam-se regras simples para a atualização da localização de cada residente. A atualização é realizada caso seja acionado o mesmo sensor de sua última localização ou um sensor conectado a ele no grafo. Apesar de intuitivo e simples, este método leva a estados incoerentes quando um sensor erroneamente não é acionado e quando um sensor reconhece dois acionamentos diferentes como apenas um devido ao tempo que leva para reconhecer uma mudança, problema frequente em sensores infravermelhos de movimento. Para resolver este problema, é apresentado o método *BUG/ED*, em que a atualização é feita de forma probabilística. Ao invés de um grafo onde cada par de sensores é conectado ou desconectado baseado na localização, existe uma matriz de probabilidades de transição de todos os sensores para todos os sensores. Esta matriz deve ser treinada de forma supervisionada através de Inferência Bayesiana utilizando dados anotados.

2.4 Conclusões do Capítulo

Conjuntos de Dados de *Smart Homes* estão disponíveis em diversos trabalhos na literatura, seja do cotidiano de residências ou de diversas atividades realizadas várias vezes. Os dados normalmente são representados como sequências de ativações ou mudanças de estado de sensores, contendo para cada uma a *timestamp*, a identificação do sensor e identificação do novo estado. Alguns conjuntos de dados possuem as chamadas anotações, marcações em determinados acionamentos de sensores indicando o início ou o fim de determinadas atividades. Estas anotações podem ser utilizadas para treinamento supervisionado de algoritmos de reconhecimento de atividades, porém são de difícil obtenção em larga escala.

O trabalho (TAX, 2018) define três diferentes problemas envolvendo previsão de eventos para ambientes inteligentes e os formula como problemas de aprendizado de máquina, apresentando um *benchmark* testando diversos métodos para solução. Métodos para solução destes problemas normalmente são baseados em modelos de Markov, compressão de dados ou diversas topologias recorrentes de Redes Neurais Artificiais.

O problema da localização de usuários, trivial para um único residente, é um desafio no caso de múltiplos residentes e é um passo necessário para separação das ativações em sequências únicas por usuário, facilitando a identificação do residente e o reconhecimento de atividades. Métodos para sua solução normalmente envolvem atualização de um modelo

em espaço de estados, que pode ser construído baseado em regras ou aprendido de forma supervisionada com dados anotados.

3 Fundamentação Teórica

Informações baseadas em *streaming* de dados de sensores podem ser interpretadas como sequências temporais discretas, onde em cada instante do tempo um determinado símbolo, categórico, é emitido.

Segundo (XING; PEI; KEOGH, 2010), métodos para reconhecimento de padrões nestas sequências podem ser divididos em três grandes famílias:

- Extração de características da sequência, gerando amostras que podem ser utilizadas com algoritmos convencionais de classificação, regressão ou agrupamento.
- Métodos de classificação baseados em distância, onde são obtidas medidas de similaridade entre diferentes sequências.
- Métodos baseados em modelos inerentemente sequenciais, como o Modelo Oculto de Markov (HMM) ou Redes Neurais Recorrentes (RNN).

3.1 Classificação Supervisionada de Sequências Temporais

3.1.1 Extração de Características de Sequências Temporais Discretas

Para alimentar algoritmos de classificação, transforma-se uma janela que representa uma sequência de tamanho fixo de W símbolos em um vetor numérico de características. Os símbolos são elementos de um conjunto de S símbolos possíveis. Esta notação será utilizada no restante desta seção.

Idealmente, a extração de características deve considerar a separação temporal entre os símbolos da sequência, a ordem de ocorrência dos mesmos e ser tolerante a pequenas translações ou mudanças de escala.

3.1.2 Contagem de Elementos

Como mostrado em (COOK et al., 2013), um método simples para extração de características significativas de sequências de tamanho fixo de acionamentos de sensores em *Smart Homes* é a contagem das ocorrências de cada símbolo na janela a ser classificada.

Tem-se uma característica para cada um dos S possíveis símbolos da sequência. Inicialmente todas possuem valor 0 e, conforme mostrado na tabela 1, a característica correspondente a determinado símbolo é incrementada em 1 para cada vez em que o

	E	**N**	**G**	**E**	**N**	**H**	**A**	**R**	**I**	**A**
E:	2									
N:	2									
G:	1									
H:	1									
A:	2									
R:	1									
I:	1									

Tabela 1 – Contagem dos símbolos presentes na palavra "ENGENHARIA". Para os demais símbolos possíveis e não encontrados na sequência, o valor da característica correspondente é igual a 0

símbolo é encontrado na sequência. O espaço de características resultante tem, portanto, S dimensões.

3.1.3 N-Gramas

O método das N-gramas é utilizado para extração de características de sequências de símbolos discretos, principalmente em aplicações de processamento de linguagem natural (NLP) (XING; PEI; KEOGH, 2010). Em (AIPPERSPACH; COHEN; CANNY, 2006), o método é utilizado para extrair características de sequências de ativações de sensores para modelar o comportamento humano em residências inteligentes.

N-Gramas são sequências de N símbolos consecutivos contidas dentro da sequência a ser classificada. Dado o valor de N, conta-se o número de ocorrências de cada possível *N-Grama* para cada sequência do conjunto amostral, obtendo-se para cada amostra um vetor de características conforme mostrado na tabela 2. O espaço de características resultante tem, portanto, S^N dimensões, sendo cada uma associada a um dos possíveis *N-Gramas*.

	E	**N**	**G**	**E**	**N**	**H**	**A**	**R**	**I**	**A**
ENG:	1									
NGE:	1									
GEN:	1									
ENH:	1									
NHA:	1									
HAR:	1									
ARI:	1									
RIA:	1									

Tabela 2 – Contagem dos *Trigramas* (N-Gramas com $N = 3$) contidos na palavra "ENGENHARIA". Para os demais N-Gramas possíveis, o valor da característica correspondente é igual a 0.

3.1.4 Convolução com Função Linear

Em (LUNDSTRÖM; JÄRPE; VERIKAS, 2016), realiza-se classificação de sequências temporais de acionamentos de sensores em uma *Smart Home* utilizando representações discretas de duração fixa dos sinais temporais correspondentes aos sensores.

O estado de cada sensor i é representado por um sinal temporal discreto $S_i(t)$, sendo cada amostra dada por um segmento de duração fixa de todos os sinais dos sensores de entrada. Na figura 2, temos representações de sinais correspondentes a uma janela de 10 segundos de atividade, obtidos por 4 diferentes sensores.

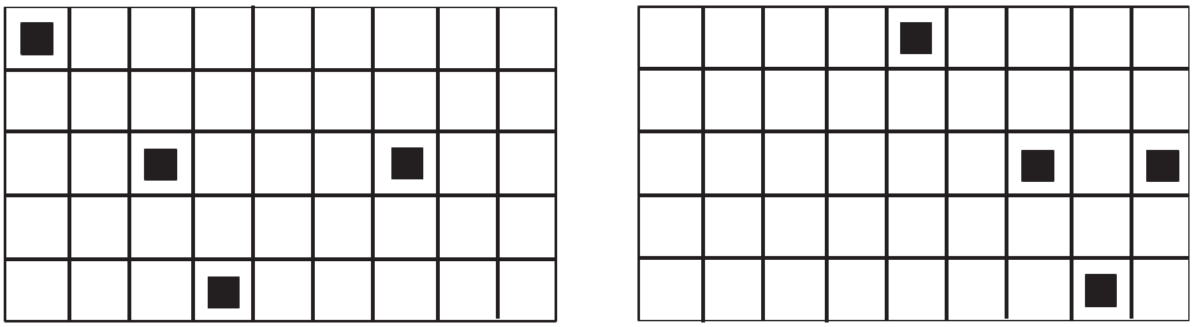

Figura 2 – Visualização dos sinais temporais correspondentes a uma mesma rota percorrida duas vezes.
Fonte: (LUNDSTRÖM; JÄRPE; VERIKAS, 2016)

Considerando-se que podem existir segmentos correspondentes à mesma atividade com pequenas translações, deslocamentos, distorções e fatores de escala, como mostrado na figura 2, é necessário que estes fenômenos não provoquem grandes alterações na série. Com este propósito, efetua-se a convolução do sinal temporal correspondente a cada sensor com uma função linear de inclinação negativa, conforme a equação 3.1. Dessa forma, conforme visualizado na figura 3, poucas amostras de cada sinal sofrerão grandes alterações com translações e mudanças de escala.

$$St_i(t) = S_t(t) * [-at + 1] \tag{3.1}$$

O vetor de características é obtido concatenando-se o sinal convoluído para cada sensor existente.

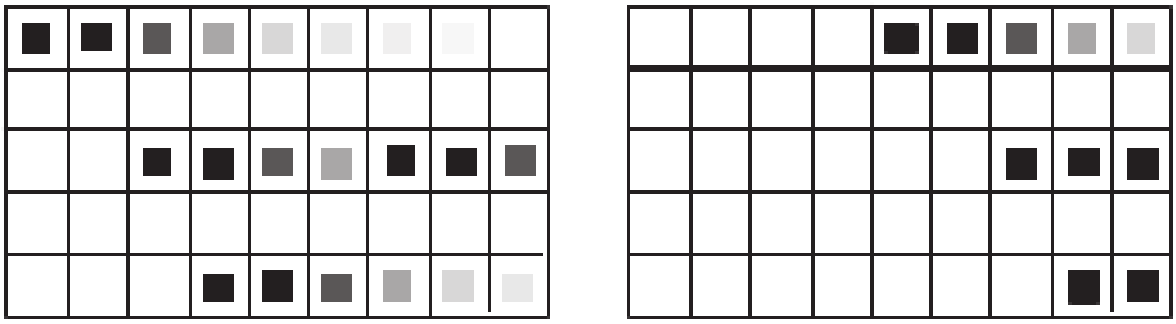

Figura 3 – Visualização dos sinais da figura 2 convoluídos com função linear.
Fonte: (LUNDSTRÖM; JÄRPE; VERIKAS, 2016)

3.2 Métodos para Classificação Supervisionada

3.2.1 Máquina de Vetores de Suporte

Propostas em (CORTES; VAPNIK, 1995), Máquinas de Vetores de Suporte (SVMs) são algoritmos de classificação que definem hiperplanos de separação maximizando a margem de classificação, ou seja, maximizando a distância do hiperplano aos pontos mais próximos ao mesmo de cada classe. Separações não-lineares são feitas através do mapeamento não-linear implícito ou explícito do espaço de entrada para um espaço de dimensão superior.

Quando existe sobreposição no conjunto de dados, com amostras de classes distintas compartilhando uma mesma área sem definir uma superfície de separação obvia (como o exemplo da figura 4), perde-se desempenho de generalização ao se reforçar a condição de margem máxima. Nas SVMs, este problema é contornado com a adição de uma variável de folga para a restrição de margem máxima na formulação do problema de otimização.

A SVM é um dos métodos mais confiáveis para problemas de classificação, com grande capacidade de generalização, sendo ainda considerada estado da arte (TAKAHASHI, 2015).

3.2.2 Classificadores por Arestas de Suporte (CLAS)

Os Classificadores por Arestas de Suporte (TORRES, 2016) constituem uma família de algoritmos de classificação de margem larga com métodos de aprendizado baseados em Grafos de Gabriel (GABRIEL; SOKAL, 1969).

Os Grafos de Gabriel são grafos não-orientados onde dois pontos a e b são vizinhos se e somente se não existe nenhum outro ponto no interior da hiperesfera cujo diâmetro é definido por estes dois pontos. Nas figuras 5 e 6, são exemplificadas as condições respectivamente para a conexão e não-conexão de dois pontos a e b no grafo.

Nos classificadores CLAS, é construído o Grafo de Gabriel correspondente ao con-

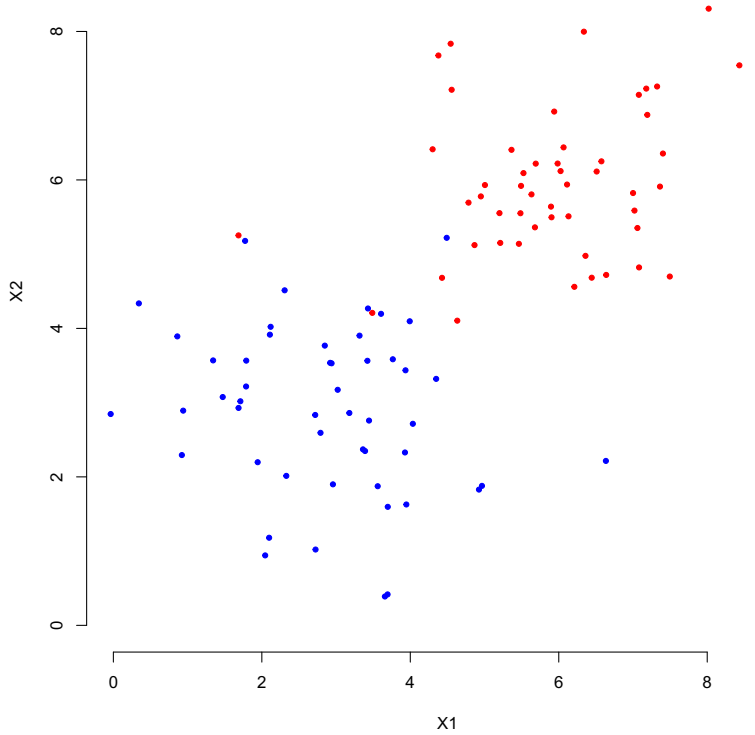

Figura 4 – Exemplo de conjunto de dados com superposição.

junto de dados e são então definidas as Arestas de Suporte, que são arestas que separam pontos de classes distintas. Através delas e de seus pontos médios, são extraídos parâmetros para configuração e construção de classificadores de margem larga (TORRES et al., 2014) (TORRES; CASTRO; BRAGA, 2015) (TORRES et al., 2015), além de um decisor (TORRES; CASTRO; BRAGA, 2012) utilizado para o método de treinamento multiobjetivo de redes neurais (ALBUQUERQUE TEIXEIRA et al., 2000).

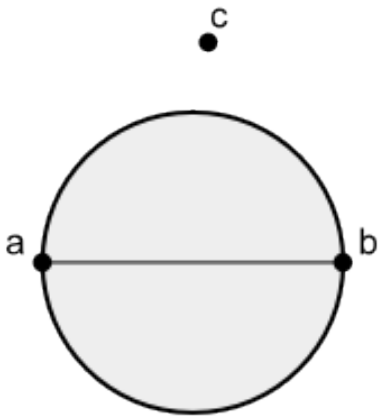

Figura 5 – Visualização de dois pontos a e b vizinhos em um Grafo de Gabriel.
Fonte: (GABRIEL. . . , 2018)

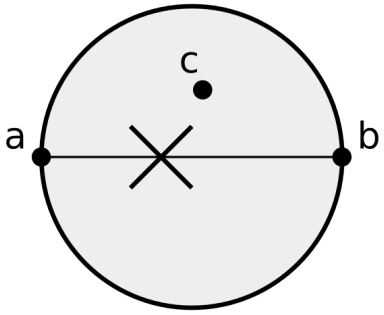

Figura 6 – Visualização de dois pontos a e b não-vizinhos em um Grafo de Gabriel devido à presença do ponto c.
Fonte: (GABRIEL. . . , 2018)

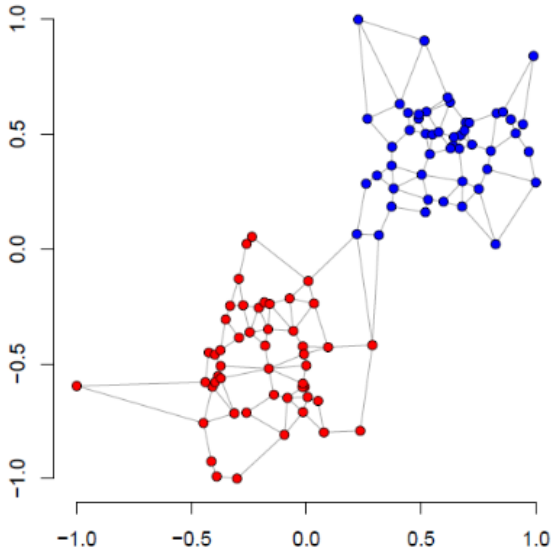

Figura 7 – Visualização do grafo de Gabriel de conjunto de dados amostrado de duas gaussianas multivariadas de diferentes médias.

Como base para este trabalho, utiliza-se o classificador CHIP-CLAS (TORRES et al., 2015). Este cria para cada aresta de suporte um hiperplano de separação que passa pelo ponto médio da mesma e maximiza a margem de separação. Obtém-se, desta forma, um classificador local de margem máxima para cada aresta de suporte.

A classificação é feita através de votação deste conjunto de hiperplanos, sendo o voto de cada hiperplano ponderado pelo inverso da distância do ponto a ser classificado ao ponto médio da aresta de suporte correspondente.

Como ocorre nos demais classificadores baseados em margem, classificadores CLAS também perdem desempenho de generalização para bases de dados com sobreposição se a condição de margem larga for rígida. O classificador deve possuir, portanto, um meio de flexibilizar tal restrição para controlar o erro de generalização. Este problema é solucionado excluindo-se do conjunto de dados as amostras caracterizam a sobreposição. Isso é feito

atribuindo-se a cada amostra um fator de qualidade dado pela razão entre o número de amostras de mesma classe conectadas ao nó correspondente do Grafo de Gabriel e o número total de amostras conectadas a ele. Exclui-se as amostras cujo fator de qualidade é menor que a média do fator de todas elas para a classe. Mostra-se, na figura 8, o conjunto de dados da figura 4 após este processo, com seu grafo de Gabriel correspondente.

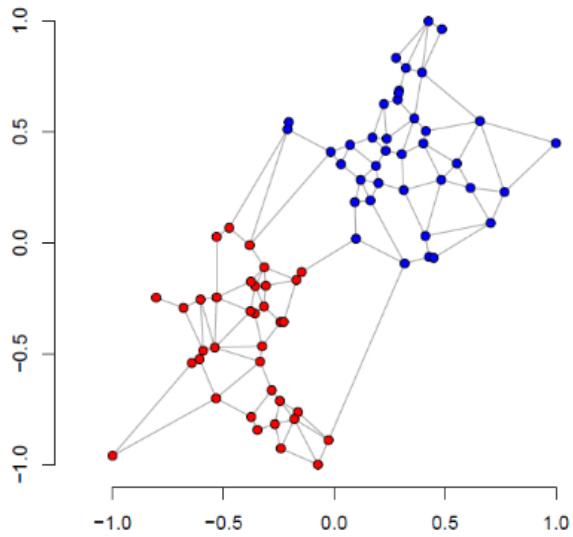

Figura 8 – Visualização do grafo de Gabriel de conjunto de dados após eliminação de sobreposição.

O método CHIP-CLAS, em particular, procura identificar se existe sobreposição nos dados antes que o processo de eliminação seja iniciado. Não há, porém, controle sobre a quantidade de dados excluída no processo, podendo a exclusão demasiada de dados levar à perda de desempenho do classificador.

3.2.3 Avaliação do Desempenho de Algoritmos de Classificação

Para avaliar e comparar o desempenho de classificadores, diversas métricas podem empregadas. A mais tradicional, a Acurácia, é definida pela quantidade de classificações corretas em relação à quantidade total de amostras classificadas.

Em conjuntos de dados altamente desbalanceados, desaconselha-se o uso da Acurácia, visto que classificações que erram sistematicamente a classe minoritária são pouco penalizadas por esta métrica. Uma métrica mais adequada para tais casos é a curva ROC, que mede independentemente a taxa de falsos positivos e falsos negativos e dá um peso igual para cada uma destas (CHAWLA, 2009).

3.3 Aprendizado de Métrica para Classificadores baseados em Distância

Classificadores baseados na distância entre dois pontos, como o SVM, o KNN e os métodos da família CLAS, foram concebidos utilizando-se a distância Euclidiana, conforme a equação 3.2.

$$D_E(\mathbf{X}, \mathbf{Y}) = \sqrt{(\mathbf{X} - \mathbf{Y})^T(\mathbf{X} - \mathbf{Y})} \tag{3.2}$$

Para alguns problemas a distância Euclidiana entre alguns pontos de mesma classe pode ser sistematicamente maior que a distância entre pontos de classes distintas, como exemplificado no conjunto de dados da figura 9.

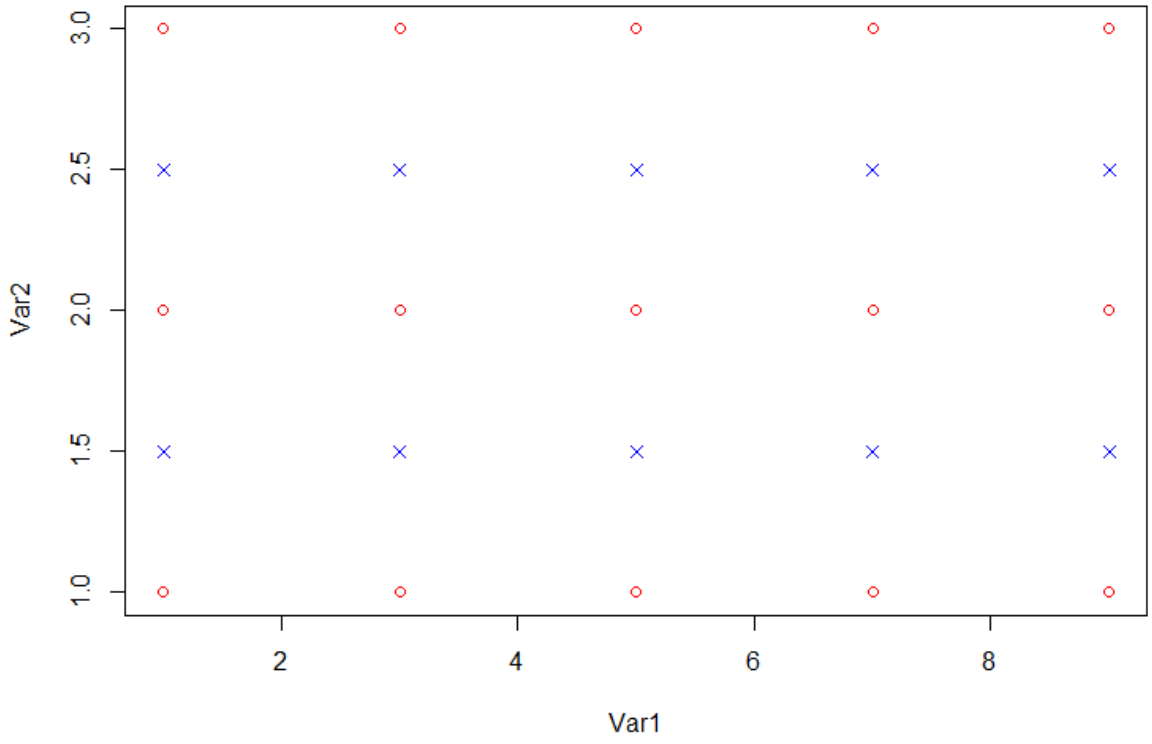

Figura 9 – Conjunto de dados ”Sanduíche”

Para solucionar este problema, pode-se usar métricas parametrizadas de distância, como a distância de Mahalanobis, mostrada na equação 3.3. Os melhores parâmetros da métrica são obtidos para cada problema através de um processo de otimização.

A distância de Mahalanobis (DUDA; HART; STORK, 2000) foi inicialmente criada como uma medida de distância entre um ponto e uma distribuição de probabilidade

multivariada. Ela é definida pela Equação 3.3, onde $\mathbf{X}$ é o vetor que indica a localização do ponto, $\mathbf{Y}$ é a média e M é o inverso da matriz covariância da distribuição de probabilidade.

$$D_M(\mathbf{X}, \mathbf{Y}) = \sqrt{(\mathbf{X} - \mathbf{Y})^T M (\mathbf{X} - \mathbf{Y})} \tag{3.3}$$

Pode-se generalizar este conceito e considerar a distância de Mahalanobis entre dois pontos $\mathbf{X}$ e $\mathbf{Y}$ quaisquer, sendo a matriz M um parâmetro para a métrica. Chamamos esta de matriz de Mahalanobis, com a restrição que M deve ser uma matriz semidefinida positiva d por d, onde d é a dimensão do espaço em que os pontos estão definidos.

A distância de Mahalanobis pode ser considerada uma generalização da distância Euclidiana, sendo esta última equivalente à distância de Mahalanobis com a matriz M igual à identidade. O conjunto dos pontos equidistantes a um centro utilizando distância de Mahalanobis gera uma superfície elipsoidal, enquanto para a distância Euclidiana, esta superfície é circular.

3.3.1 Large Margin Nearest Neighbors (LMNN)

O LMNN (WEINBERGER; SAUL, 2009) é um processo criado para aprendizado de métrica para classificadores KNN. A melhor matriz de Mahalanobis é encontrada através da minimização de uma função convexa baseada no erro Leave-One-Out (LOO) deste classificador.

O método recebe o número de vizinhos mais próximos do KNN como parâmetro (K). Pode-se definir a função objetivo para o LMNN como composta de dois termos. O primeiro penaliza a soma das distâncias de cada ponto a seus vizinhos mais próximos, tendo efeito de aproximá-los, sendo dado pela Equação 3.4, onde $j \rightsquigarrow i$ significa que j está entre os K vizinhos mais próximos de i.

$$\varepsilon_{pull}(M) = \sum_{j \rightsquigarrow i} D_M^2(\mathbf{x}_i, \mathbf{x}_j) \tag{3.4}$$

O segundo termo penaliza curtas distâncias entre cada ponto e pontos de classes distintas entre seus vizinhos mais próximos (impostores). É definido pela Equação 3.5.

$$\varepsilon_{push}(M) = \sum_{i, j \rightsquigarrow i} \sum_{l} (1 - y_{il})[1 + D_M^2(\mathbf{x}_i, \mathbf{x}_j) - D_M^2(\mathbf{x}_i, \mathbf{x}_l)] \tag{3.5}$$

Para garantir a condição de que a matriz M seja semidefinida positiva durante o processo de otimização, ela pode ser decomposta como o quadrado de uma matriz simétrica, conforme a equação 3.6. Otimiza-se, então, a matriz simétrica L. A distância de Mahalanobis pode então ser obtida através da distância Euclidiana das transformações lineares dos pontos por esta matriz simétrica, como visto na Equação 3.7.

$$M = LL \tag{3.6}$$

$$D_M = dist(L\mathbf{X}, L\mathbf{Y}) \tag{3.7}$$

Assim, a classificação utilizando a distância Euclidiana, utilizando-se esta transformação linear L nos dados de entrada, é equivalente à utilização da distância de Mahalanobis. Somando-se as penalidades, adicionando a restrição Semidefinida-Positiva para a matriz M e modificando o segundo termo da função objetivo de forma a adicionar variáveis de folga e colocá-la numa forma mais adequada para a solução, temos na Equação 3.8 a formulação final do problema de otimização:

$$\begin{aligned} L* = \arg\min_{L} \quad & \sum_{j \rightsquigarrow i} d(L\mathbf{x}_i, L\mathbf{x}_j) + \sum_{i,j \rightsquigarrow i} (1 - y_{il})\xi_{ijl} \\ \text{sujeito a} \quad & d(L\mathbf{x}_i, L\mathbf{x}_l) - d(L\mathbf{x}_i, L\mathbf{x}_j) \geq 1 - \xi_{ijl}, \\ & \xi_{ijl} \geq 0, \\ & LL \succeq 0. \end{aligned} \tag{3.8}$$

Após o aprendizado de métricas, o algoritmo LMNN toma a decisão utilizando o classificador KNN com a métrica de distância aprendida. Assim, cada amostra do conjunto de testes é classificada de acordo com seus K vizinhos mais próximos segundo a métrica de Mahalanobis obtida.

3.4 Modelos Ocultos de Markov para Sequências Temporais

Um Modelo Oculto de Markov (HMM) (RABINER; JUANG, 1986) é um modelo estatístico onde assume-se que o sistema pode ser modelado por uma Cadeia de Markov. A Cadeia de Markov é um processo estocástico que consiste em um conjunto de estados, cada um possuindo um conjunto de probabilidades de transição, que indica a probabilidade de transição de cada estado para cada estado, e uma probabilidade de emissão de símbolos, que indica a probabilidade de emissão de cada símbolo pelo sistema para cada estado.

No HMM, o estado do sistema não é diretamente visível ao observador, apenas a saída que o sistema emite. Tem-se, então, três problemas canônicos para HMMs:

- **Avaliação:** Dado um modelo λ e uma sequência de observações $\mathbf{O}$, deseja-se encontrar a probabilidade $p(\mathbf{O}|\lambda)$, da sequência $\mathbf{O}$ ser geradas pelo modelo λ.
- **Decodificação:** Dado um modelo λ e uma sequência de observações $\mathbf{O}$, deseja-se encontrar a sequência de estados $\mathbf{S}$ mais provável de ter produzido a sequência $\mathbf{O}$.

- **Aprendizado:** Dado um modelo λ e uma sequência de observações $\mathbf{O}$, deseja-se encontrar o melhor conjunto de parâmetros de transição e de emissão para λ, de forma a maximizar a probabilidade $p(\mathbf{O}|\lambda)$.

Para a aplicação neste trabalho, são importantes a solução dos problemas de Decodificação e Aprendizado.

3.4.1 Problema da Decodificação

Um algoritmo de decodificação baseado no princípio de Máxima Verossimilhança e em programação dinâmica foi proposto em (VITERBI, 1967) e é de uso canônico para encontrar a sequência de estados mais provável em modelos como HMMs, dados os parâmetros de transição e emissão do modelo e a sequência $\mathbf{O}$ de símbolos emitidos.

Também existem algoritmos aproximados, baseados em técnicas de amostragem, úteis principalmente para problemas de grandes dimensões, computacionalmente intratáveis utilizando o algoritmo de Viterbi.

3.4.2 Problema do Aprendizado

Também baseado no princípio de Máxima Verossimilhança, o algoritmo de Baum-Welch (BAUM et al., 1970) é utilizado para encontrar mínimos locais para o melhor ajuste dos parâmetros de transição e emissão de um HMM, dados os parâmetros iniciais e uma sequência $\mathbf{O}$ de símbolos observados.

3.4.3 Modelos Ocultos de Markov Fatoriais

Um Modelo Oculto de Markov Fatorial (FHMM) (GHAHRAMANI; JORDAN, 1996) consiste em vários Modelos de Markov, emitindo símbolos para uma mesma sequência $\mathbf{O}$.

Uma das formas para se tratar os problemas de avaliação, decodificação e aprendizado para FHMMs consiste em produzir um HMM equivalente ao FHMM, cujo espaço de estados equivalente ao produto cartesiano do espaço de estados de todos os Modelos de Markov que o compõe. Assim, para um FHMM composto de d Modelos de Markov, cada um com k estados, o HMM equivalente possui k^d estados, cada um representando uma possível combinação de estados para o FHMM.

4 Metodologia

4.1 Banco de Testes

O Banco de Testes que fornece os dados para este trabalho consiste em um apartamento de 7 ambientes (2 quartos de solteiro, quarto de casal, sala de estar, sala de jantar, cozinha e escritório). Por estes cômodos, tem-se o histórico em tempo real dos dados de 28 atuadores (como relés ou *dimmers* para lâmpadas e motores de cortina), 13 interfaces com o usuário (interruptores e outros comandos), 6 sensores infravermelhos de presença (sala de estar, sala de jantar, cozinha, quarto de casal e escritório) e 2 sensores de abertura de porta nas entradas do apartamento (entrada principal e entrada pela cozinha). Também são registrados comandos enviados ao sistema por dispositivos móveis, como celulares e tablets.

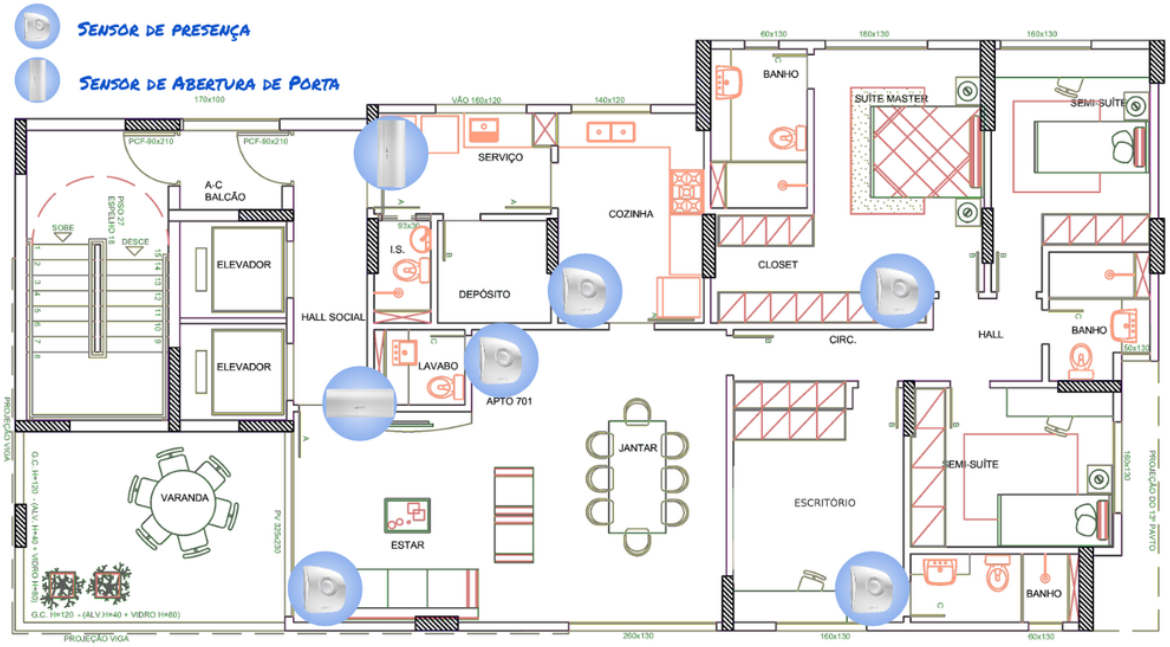

Figura 10 – Planta da residência e localização dos sensores para o banco de testes.

4.2 Coleta, Armazenamento e Representação dos Dados

Os sistemas propostos baseiam-se no sistema de automação residencial *Minibox*, da companhia nacional *Neocontrol*. Este sistema centraliza a comunicação de todos os sensores, interfaces e demais dispositivos de uma determinada residência ou imóvel em um dispositivo ligado à internet, que recebe comandos de dispositos móveis e transmite os

dados recebidos dos sensores e atuadores para um servidor na nuvem através do protocolo MQTT para comunicação com dispositivos móveis.

O protocolo MQTT é um protocolo de transmissão e recepção de mensagens muito utilizado em contextos de *Internet of Things* (IoT) que roda sobre TCP/IP. O protocolo é aberto e foi projetado para ser simples, leve e de fácil implementação, adequado às necessidades para esta aplicação (OASIS, 2014).

Foi criado um cliente em linguagem C com o trabalho de ouvir e interpretar todas as comunicações do servidor MQTT com os diversos dispositivos do banco de testes e registrá-las em um banco de dados PostgreSQL, acessível remotamente.

As informações apresentadas são uma sequência temporal de eventos de dispositivos, contendo para cada amostra uma *timestamp*, o dispositivo ativado (identificador do dispositivo e canal) e um valor descrevendo o estado do sensor, assim como a representação vista na seção 2.1.1.

Os possíveis valores para cada tipo de dispositivo presente são descritos a seguir:

- Sensores de Movimento: Estados binários ON/OFF.
- Sensores de Porta: Estados binários OPEN/CLOSE.
- Lâmpadas: Estados inteiros entre 0 e 255, descrevendo a intensidade da luz (dimmer).
- Interruptores e interfaces de comando: Estados binários ON/OFF, descrevendo o acionamento do interruptor ou da interface.

Observa-se que cada amostra carrega muito pouca informação, sendo necessário um pré-processamento dos dados de forma a condiciona-los para extrair informações significativas sobre o evento corrente.

4.3 Previsão de Eventos

Certos eventos, dos quais são obtidas informações, são disparados pelos usuários da casa, como a ativação de comandos específicos, o acionamento de interruptores e o abrir de portas. As informações contidas na *timestamp* destes eventos, tais como hora do dia, dia da semana, aliadas às informações dos dispositivos acionados nos minutos anteriores ao evento de interesse, podem servir como preditores para um possível acionamento do evento em um futuro próximo, mostrando a capacidade do sistema de encontrar padrões no comportamento de seus usuários.

Deseja-se solucionar o problema da previsão do próximo evento a ser executado, descrito na seção 2.2, com 60 segundos de antecedência.

Como se trata de um problema de classificação com mais de duas classes, pode ser adotada uma abordagem *um-contra-todos*. Treina-se um classificador por classe, especializado em diferenciar esta das demais, e escolhe-se a classe como sendo a correspondente ao classificador que apresentou maior saída, em magnitude. Também se cria uma classe negativa para eventos, caso nenhum evento ocorra no intervalo definido.

4.4 Extração de Características

É necessária para a classificação a extração de características das sequências temporais de acionamentos de dispositivos. Os métodos para Seleção de Características visam extrair da sequência temporal correspondente a uma janela de tamanho fixo de W símbolos, equivalente a M minutos, com M variável.

São S símbolos possíveis, cada um deles correspondente a um estado de um sensor ou dispositivo. Para as luzes de intensidade variável, são considerados apenas 2 estados, ligado ou desligado.

São explorados os seguintes métodos encontrados na literatura:

4.4.1 Contagem de Acionamentos

Conforme descrito na seção 3.1.2, este método produz um número S de características. Para cada símbolo possível, é adicionada uma característica contendo o número de vezes que o mesmo é presente na janela.

Esta abordagem, bastante simples, não leva em conta o momento do acionamento dos dispositivos, a ordem ou o intervalo entre acionamentos. Propõe-se uma modificação neste método para caracterizar o intervalo entre acionamentos através da inserção de um símbolo indicador de inatividade na sequência a ser classificada em momentos onde o intervalo entre dois acionamentos for maior que um *threshold*.

4.4.2 N-Gramas

Parametrizado por um valor inteiro N, este método consiste na contagem de ocorrências na janela de cada uma das possíveis combinações de N acionamentos consecutivos, produzindo um número de características S^N, assim como descrito na subseção 3.1.3.

Dado o grande número de combinações possíveis, especialmente para valores de N maiores que 3, as características se tornam esparsas, com poucas tendo valores diferentes de 0. Para reduzir o consumo de memória e o tempo de processamento, utiliza-se neste trabalho apenas as características correspondentes aos N-Gramas presentes no conjunto de treinamento, sendo cada N-Grama mapeado para um índice consecutivo na ordem de

sua primeira ocorrência através de um *hashmap*. As demais combinações, caso surjam para avaliação, são consideradas raras e não farão parte do conjunto.

Como visto na seção 3.1.3, este método não leva em conta o momento do acionamento dos dispositivos ou o intervalo entre acionamentos, mas leva em conta a ordem destes. Para caracterizar longos períodos de inatividade, insere-se um símbolo indicador de inatividade na sequência a ser classificada em momentos onde o intervalo entre dois acionamentos for maior que um determinado *threshold*, assim como foi feito para o método de contagem de símbolos.

4.4.3 Convolução com Função Linear

Como visto na seção 3.1.4, o sinal temporal correspondente a cada símbolo é convoluído com uma função linear $x = -at + 1$ de inclinação negativa. Considera-se $a = \frac{1}{M}$, dependente da largura M, em minutos, da janela.

Por razões de custo computacional, não integra o conjunto de características o sinal temporal correspondente a todo o intervalo da janela considerada discretizada em segundos, como feito em (LUNDSTRÖM; JÄRPE; VERIKAS, 2016), mas apenas o instante final. Desta forma, evita-se um espaço de características de dimensão $60 * M * S$ e obtém-se um espaço de características de dimensão S.

Este método leva em conta o momento do acionamento dos dispositivos, não sendo necessária, portanto, a inserção de símbolos indicadores de inatividade na sequência a ser classificada.

4.5 Classificação

Após a etapa de extração de características, a dimensão do espaço de características é reduzida através de Análise de Componentes Principais (WOLD; ESBENSEN; GELADI, 1987) e, então, as características resultantes podem ser diretamente utilizadas em um algoritmo de classificação. Utiliza-se como *baseline* para classificação o algoritmo SVM.

O ajuste de hiperparâmetros para o SVM é custoso computacionalmente, devendo ser feito através de busca em *grid*, com cada possível combinação de hiperparâmetros tendo seu desempenho avaliado por *10-fold Cross-Validation* (KOHAVI, 1995), de forma a obter a melhor combinação.

Classificadores da família CLAS mostraram empiricamente ter desempenho estatisticamente equivalente ao SVM com *kernels* RBF e Polinomial em um *benchmark* de 15 conjunto de dados (TORRES; CASTRO; BRAGA, 2012), com a vantagem de não possuirem hiperparâmetros a serem ajustados. Testa-se, então, o classificador CHIP-CLAS para esta aplicação, com o objetivo de evitar o processo de busca em *grid*.

O processo de eliminação de sobreposição do algoritmo CHIP-CLAS, porém, pode torná-lo mal comportado para conjuntos desbalanceados ou com pequeno tamanho amostral. Os conjuntos de dados obtidos através de extração de características do banco de testes para previsão de eventos se mostraram extremamente desbalanceados e com pouca representação da classe minoritária, possuindo eventos cuja ocorrência é rara, como o acionamento das luzes indiretas da sala de estar. Para estes, o descarte de amostras pode levar a perda de informações importantes. Justifica-se, então, o desenvolvimento de uma nova abordagem para classificação (AM-CHIP-CLAS), que inclui uma etapa de aprendizado de métrica anterior à eliminação de sobreposição, minimizando assim o descarte de dados mantendo-se o desempenho de generalização.

4.5.1 AM-CHIP-CLAS

A abordagem AM-CHIP-CLAS foi baseada no método LMNN, descrito na seção 3.3, introduzindo ao CHIP-CLAS uma etapa de aprendizado de métrica. É encontrada uma métrica de Mahalanobis que melhor se adequa ao problema.

Adapta-se o aprendizado de métrica do método LMNN para utilização em classificadores CLAS. Espera-se que o processo não possua hiperparâmetros, de forma que continue desnecessário o ajuste de parâmetros através de busca em *grid*. Para tal, modifica-se a função objetivo do LMNN. Ao invés de considerar os K vizinhos mais próximos para cada amostra, a função é calculada considerando-se os vizinhos conectados a ele em um Grafo de Gabriel construído utilizando distância Euclidiana. Elimina-se assim a necessidade de um parâmetro K e leva-se em conta no aprendizado de métrica a estrutura geométrica do problema, também utilizada na classificação.

A formulação do problema de otimização mantém-se na forma vista na Equação 3.8, alterando-se o significado da vizinhança $j \rightsquigarrow i$, significando agora que j é conectado a i no Grafo de Gabriel. Isto mantém as características de convexidade e de restrições esparsamente violadas do problema de otimização do LMNN, facilitando o processo de otimização.

Obtida a matriz L, são feitas as transformações lineares nos conjuntos de treino e teste e o problema de classificação é solucionado pelo algoritmo CHIP-CLAS.

4.6 Modelo Não-Supervisionado para Localização de Usuários na Smart Home

Busca-se extrair dos dados, de forma não-supervisionada, o número de pessoas presentes na casa e a localização delas. Para isso, foi desenvolvido um modelo FHMM

cuja sequência **O** de símbolos emitidos corresponde aos dados observados para a *Smart Home*, como descrito a seguir.

Para cada residente, é criado um modelo de Markov cujos estados correspondem à presença do residente em cada cômodo da casa ou então fora da casa. Os parâmetros iniciais de transição e de emissão dos modelos podem ser estimados *a priori* através de regras.

Para os parâmetros de transição, estima-se:

- 0 entre estados correspondentes a cômodos que não se conectam (uma pessoa não tem como transitar entre dois cômodos que não se ligam),
- 1 entre estados correspondentes a cômodos que se conectam (uma pessoa tem probabilidade de mudar de cômodo entre duas ativações de sensores),
- 50 entre os estados e eles próprios. (Uma pessoa tem uma grande probabilidade de continuar no mesmo cômodo entre duas ativações de sensores).

Para os parâmetros de emissão, estima-se:

- 100, se o dispositivo correspondente ao símbolo estiver presente em um cômodo correspondente ao estado,
- 1 caso não estiver (não-nulo, para o sistema tolerar algum nível de ruído sem se tornar inconsistente).

Os parâmetros de transição e emissão para cada estado são normalizados de forma que a soma das probabilidades resulte sempre em 1.

A composição dos modelos de Markov para todos os residentes forma então um FHMM, podendo a localização de cada indivíduo ser obtida através da solução do problema da decodificação.

O modelo de Markov, sendo discreto, não leva em conta o tempo entre emissões. Da mesma forma que nos métodos de extração de características por contagem de emissões e por N-Gramas, isto pode ser contornado com a criação de um símbolo representativo de inatividade, modificando-se a sequência emitida para incluí-lo caso o tempo de inatividade seja maior que um *threshold*.

Para este trabalho, o aprendizado do modelo não foi necessário visto que o modelo apresentou comportamento satisfatório utilizando os parâmetros estimados *a priori*.

4.7 Conclusões do Capítulo

É apresentado o banco de testes desenvolvido através de sensores e interfaces ligadas ao sistema *Minibox*, em um apartamento com 4 residentes. Os dados são transmitidos para um banco de dados remoto e apresentados na forma de sequências de mudança de estados, sendo cada linha composta de uma *timestamp*, identificadores do dispositivo e o novo estado.

Formula-se o problema da previsão de eventos em ambientes inteligentes como um problema de classificação. Em uma abordagem um-contra-todos, são treinados um classificador por evento, cujas classes são a ocorrência ou não-ocorrência do evento após um período de tempo pré-estabelecido da última amostra coletada.

Propõe-se, para a previsão de eventos, três métodos para extração de características das sequências de mudanças de estado dos dispositivos interligados, adaptados de métodos encontrados na literatura: Contagem de Emissões, N-Gramas e Convolução com Função Linear.

Para a classificação, são estudadas a utilização de Máquinas de Vetor de Suporte e do algoritmo CHIP-CLAS, baseado em grafos de Gabriel e que não possui hiperparâmetros. Considerando o desbalanceamento dos dados extraídos do banco de testes e a raridade de amostras das classes minoritárias, uma nova abordagem para classificadores CHIP-CLAS é proposta para reduzir a quantidade de dados descartados no processo de eliminação de sobreposição, consistindo em uma etapa de aprendizado de métrica antes da classificação. São encontrados nesta etapa os parâmetros da métrica de Mahalanobis que melhor agrupam dados de mesma classe e afastam dados de classes distintas.

O problema da localização de usuários é formulado como um problema de decodificação de Modelos Ocultos de Markov Fatoriais (FHMM), com parâmetros estimados *a priori* através de regras baseadas na planta do ambiente. O refinamento do modelo pode ser feito através da solução do problma do aprendizado do FHMM.

5 Experimentos e Resultados

5.1 Recursos Computacionais Utilizados

Os experimentos foram executados em um laptop com processador Intel Core i5, 4 GB de memória RAM, sem placa de vídeo dedicada.

5.2 Validação e Desempenho dos métodos de Extração de Características para Previsão de Eventos em *Smart Homes*

Os experimentos para validação da extração de características na previsão de eventos em *Smart Homes* consistem na avaliação e comparação dos diferentes métodos para para os dados no banco de testes utilizando o algoritmo classificador SVM com *kernel* RBF e ajuste de hiperparâmetros via busca em *grid*. A avaliação é feita via *5-fold Cross Validation* (KOHAVI, 1995), utilizando como métrica de desempenho a área sob a curva ROC (AUC).

Mostra-se os eventos considerados, juntamente com o número de ocorrências de cada um:

- COZINHA: Acionamento do interruptor da luz da Cozinha (913 ocorrências).
- MESA: Acionamento do interruptor da luz sobre a mesa de jantar (61 ocorrências).
- SOFA: Acionamento do interruptor da luz sobre o sofá (77 ocorrências).
- VARANDA: Acionamento do interruptor da luz da varanda (17 ocorrências).
- HALL: Acionamento dos interruptores das luzes do Hall (42 ocorrências).
- ABAJOUR: Acionamento do interruptor da luz do Abajour (9 ocorrências).

Conforme a abordagem *um-contra-todos*, classificadores foram construídos para separar cada evento dos demais, sendo também incluídos na classe negativa para cada evento os períodos que marcam a metade de grandes intervalos entre eventos (maiores de 10 minutos), estes últimos consistindo em 640 amostras.

Mostra-se na tabela 3 a quantidade de amostras positivas e negativas resultantes para o classificador especializado em cada evento.

Tabela 3 – Quantidade de amostras para os classificadores especializados em cada evento.

Evento	# Classe Pos.	# Classe Neg.	Total
COZINHA	913	966	1779
MESA	61	1718	1779
SOFA	77	1702	1779
VARANDA	17	1762	1779
HALL	42	1737	1779
ABAJOUR	9	1770	1779

5.2.1 Resultados

Os resultados dos experimentos utilizando os métodos propostos para extração de características, junto com o *rank* médio de cada método, são mostrados na Tabela 4. Para verificar a relevância estatística do *rank* médio, foi utilizado o teste de Bonferroni-Dunn, visualizado na figura 11.

Tabela 4 – AUC Média e Desvio Padrão para previsão de eventos para diversos métodos de extração de características.

Evento	Contagem (AUC)	SD	N-Gramas (AUC)	SD	Convolução (AUC)	SD
COZINHA	0.7185	0.0315	0.7208	0.0248	0.7512	0.0250
MESA	0.9869	0.01556	0.9866	0.0170	0.9811	0.0236
SOFA	0.9041	0.0513	0.8918	0.0619	0.9140	0.0621
VARANDA	0.8334	0.1132	0.6462	0.1154	0.7727	0.1575
HALL	0.9670	0.0382	0.9477	0.0307	0.9609	0.0430
ABAJOUR	0.9672	0.0453	0.8963	0.2096	0.9455	0.0747
Rank Médio	1.5		2.67		1.83	

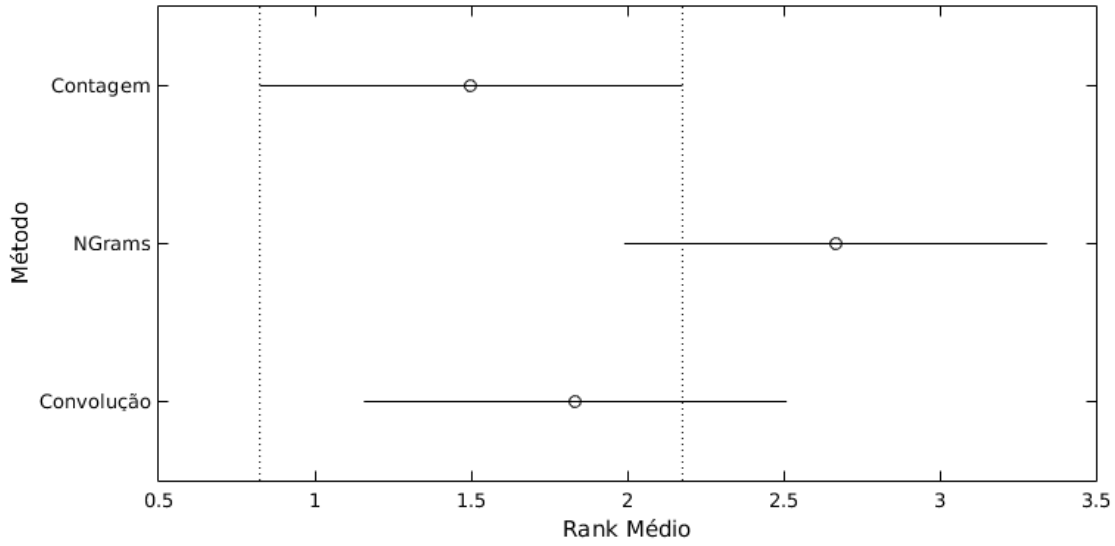

Figura 11 – Visualização para o Teste de Friedman com teste post-hoc de Bonferroni

5.2.2 Discussão

Os resultados apresentados validam os métodos utilizados neste trabalho para extração de características. Destaca-se o método da Contagem de Acionamentos de Sensores, descrito na seção 3.1.2, que apresentou o melhor *rank* médio para os eventos considerados mesmo sendo o mais simples dentre os três métodos testados.

O teste de Friedman não indicou, porém, que não se pode ter certeza da superioridade de nenhum dos métodos em relação aos demais.

5.3 Validação da abordagem AM-CHIP-CLAS para Classificação

A abordagem AM-CHIP-CLAS, com aprendizado de métrica, foi avaliada através de *10-fold Cross Validation* (KOHAVI, 1995). Mediu-se a porcentagem dos dados desconsiderados no treinamento e o desempenho da classificação através de AUC (área sob a curva ROC). Para validar os dados em diferentes condições e obter-se significância estatística nos experimentos, estes foram realizados com 13 bases de dados reais obtidas através do repositório UCI (BACHE; LICHMAN, 2013) e 2 problemas de expressão gênica: *Golub* (GOLUB, 1999) e *BcrHess* (HESS et al., 2006).

A porcentagem desconsiderada dos dados foi comparada com a obtida para o algoritmo CHIP-CLAS em sua abordagem original, sem aprendizado de métrica. O desempenho foi comparado com o algoritmo CHIP-CLAS sem aprendizado de métrica e com o classificador SVM com *Kernels* RBF e Polinomial. Os melhores parâmetros para o SVM foram encontrados através de *10-fold Cross Validation* e busca em *grid*.

Buscou-se também visualizar os efeitos do aprendizado de métrica na superfície de separação.

5.3.1 Visualização da Superfície de Separação

Para visualização do efeito do aprendizado de métrica, o método AM-CHIP-CLAS e o CHIP-CLAS original foram utilizados para separação de um conjunto de dados sintético de duas dimensões. Foi criado para tal um conjunto de dados consistindo em fileiras intercaladas de classes distintas alinhadas com o eixo X, adicionadas de ruído gaussiano em ambas dimensões. Desta forma, algumas amostras significativas para o treinamento ficam próximas de mais pontos da classe oposta que as demais. Assim, induz-se ao erro o método para eliminação de sobreposição do CHIP-CLAS original, destacando assim a diferença entre ambas as metodologias.

O método CHIP-CLAS original desconsiderou 41.67% dos dados no processo de classificação, gerando a superfície de separação da Figura 12. O método AM-CHIP-CLAS

não desconsiderou nenhuma amostra no processo de classificação, gerando a superfície da Figura 13.

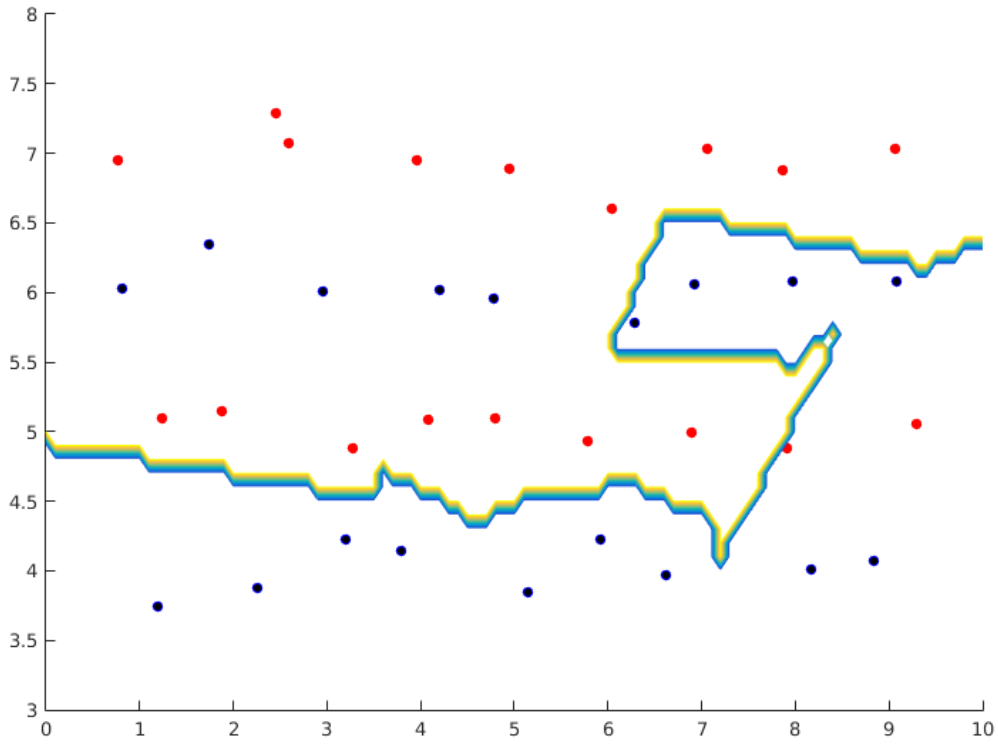

Figura 12 – Visualização da superfície de separação para o CHIP-CLAS original

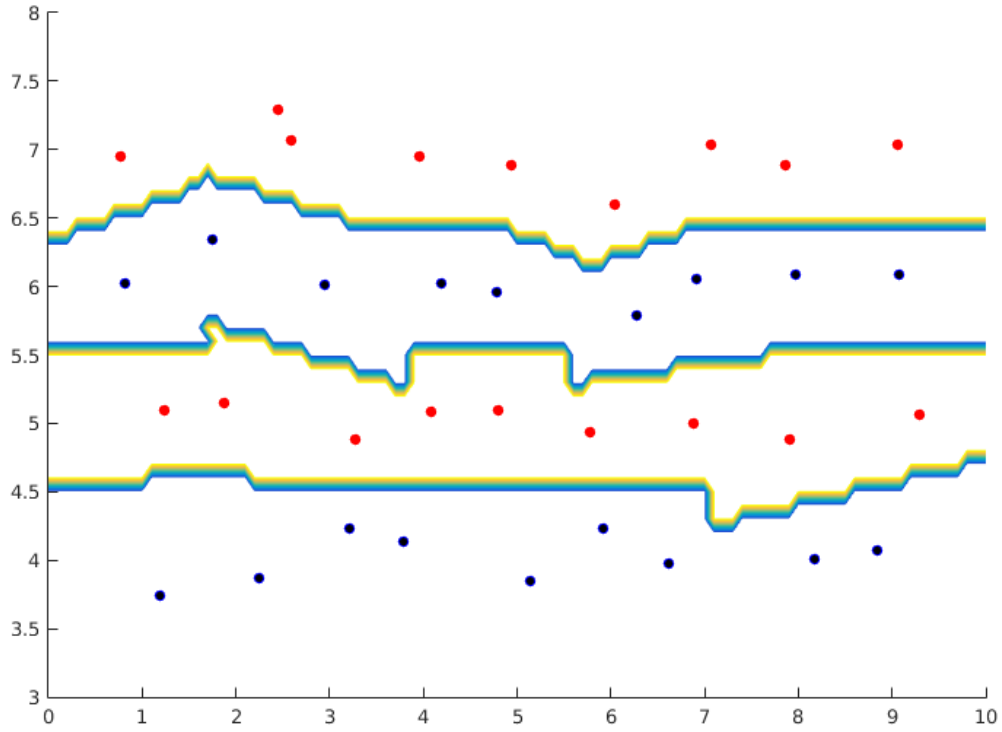

Figura 13 – Visualização da superfície de separação para o AM-CHIP-CLAS

5.3.2 Utilização dos Dados

A porcentagem dos dados desconsiderados no treinamento para cada execução da validação cruzada foi medida. Calculou-se a razão entre a quantidade de amostras descartadas no processo de filtragem e o total de dados da base, com e sem aprendizado de métrica. Os resultados médios obtidos para as execuções se encontram na Tabela 5.

Para 6 das 15 bases testadas, a porcentagem desconsiderada dos dados diminuiu consideravelmente, sofrendo variação de menos de 1% para cima ou para baixo nas bases restantes. Isto sugere uma maior utilização dos dados para o AM-CHIP-CLAS. A significância estatística desta superioridade pode ser estabelecida através de um teste estatístico de Wilcoxon pareado (DEMŠAR, 2006). O teste unilateral foi utilizado, com nível de confiança de 95% ($\alpha = 0.05$). O Valor-p obtido no teste foi $p = 0.040$, de forma que $p < \alpha$, confirmando estatisticamente a maior utilização dos dados para a nova abordagem com 95% de confiança.

Tabela 5 – Porcentagem média desconsiderada dos dados.

	dataset	CHIP-CLAS	AM-CHIP-CLAS
1	sonar	0.00	0.00
2	breastcancer	15.32	10.85
3	australian	37.97	38.89
4	diabetes	44.68	44.73
5	breastHess	38.94	26.16
6	bupa	48.28	48.76
7	haberman	45.97	45.24
8	banknote	0.00	0.00
9	fertility	43.11	24.78
10	parkinsons	10.36	3.24
11	climate	39.55	22.55
12	ILPD	47.55	47.27
13	german	46.86	47.30
14	heart	43.25	43.66
15	golub	37.05	0.00

5.3.3 Desempenho

A AUC para cada execução da validação cruzada foi medida e foi extraída a média para cada base de dados. Os resultados obtidos se encontram na Tabela 6, juntamente com a média da posição de cada classificador num *ranking* de desempenho para cada base de dados.

Tabela 6 – AUC Média das execuções e Rank Médio dos Classificadores.

dataset	AM-CHIP-CLAS	CHIP-CLAS	RBF-SVM	Poly-SVM
sonar	0.84	0.88	0.84	0.87
breastcancer	0.97	0.96	0.97	0.96
australian	0.86	0.85	0.86	0.87
diabetes	0.71	0.72	0.71	0.71
breastHess	0.83	0.81	0.76	0.77
bupa	0.58	0.61	0.67	0.72
haberman	0.54	0.56	0.52	0.50
banknote	1.00	0.99	1.00	1.00
fertility	0.50	0.59	0.50	0.50
parkinsons	0.89	0.90	0.77	0.81
climate	0.85	0.84	0.53	0.72
ILPD	0.57	0.57	0.49	0.50
german	0.70	0.67	0.66	0.68
heart	0.81	0.80	0.83	0.83
golub	0.55	0.77	0.80	0.78
Rank Mean	2.20	2.40	2.93	2.47

Para avaliação estatística dos resultados de múltiplos classificadores, é indicado o teste de Friedman (DEMŠAR, 2006). Para um nível de confiança de 95% ($\alpha = 0.05$),

foi obtido um Valor-p de $p = 0.445$. O resultado obtido não é suficiente para rejeitar a hipótese nula de que nenhum dos classificadores possui desempenho estatisticamente diferente dos demais. Para melhor visualizar o desempenho dos classificadores, foi feito o teste *post-hoc* de Bonferoni-Dunn (DEMŠAR, 2006), obtendo-se o gráfico da Figura 14, com o eixo horizontal indicando o *rank* (quanto menor, melhor o desempenho).

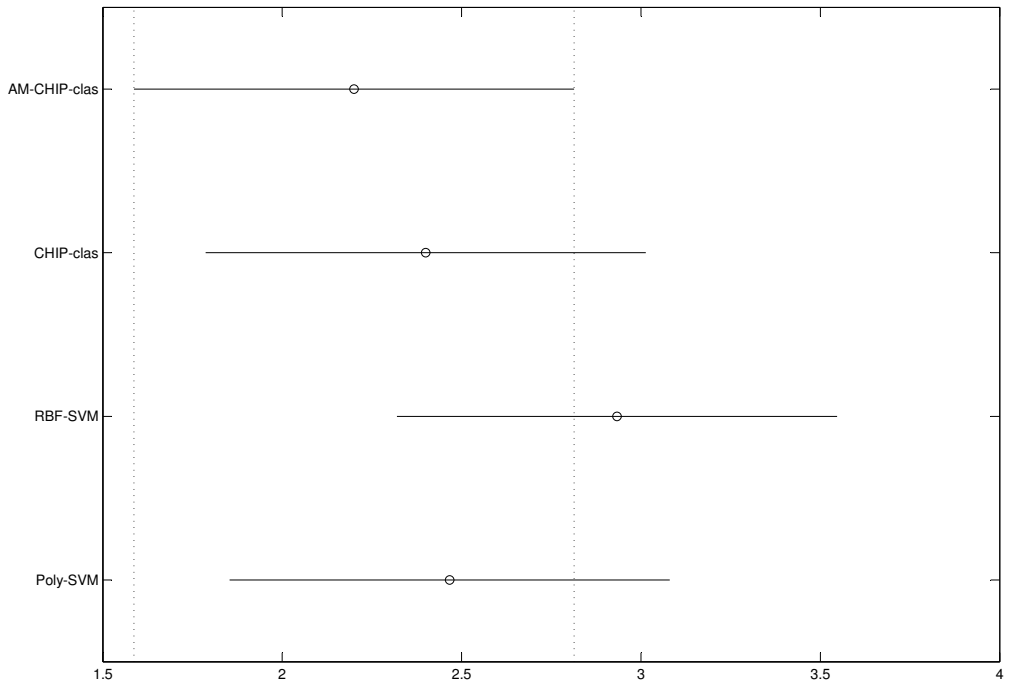

Figura 14 – Visualização para o teste *post-hoc* de Bonferroni-Dunn

5.3.4 Discussão

Verifica-se que o desempenho da abordagem AM-CHIP-CLAS não difere significativamente do classificador CHIP-CLAS sem aprendizado de métrica para os testes realizados, com *rank* médio pouco superior a este. Ambos CHIP-CLAS e AM-CHIP-CLAS se mostram superiores no *rank* médio aos classificadores SVM testados. O teste de Friedman, porém, mostrou que essa diferença no *rank* médio não foi suficiente para possuir relevância significativa, justificando um experimento futuro em maior escala.

5.4 Desempenho dos classificadores CHIP-CLAS e AM-CHIP-CLAS para o problema da Previsão de Eventos em *Smart Homes*

Tanto o algoritmo CHIP-CLAS sem aprendizado de métrica quanto o AM-CHIP-CLAS foram testados para o problema da previsão de eventos em *Smart Homes*, utilizando os dados do banco de testes, em conjunto com os três métodos validados para extração de características.

5.4.1 Resultado

O desempenho obtido para os classificadores CHIP-CLAS e AM-CHIP-CLAS no problema da previsão de eventos é mostrados na tabela 7. Para fácil comparação, também são incluídos os valores obtidos utilizando o algoritmo utilizado como *baseline* (SVM com *kernel* RBF), já mostrados na tabela 4.

Tabela 7 – AUC Média e Desvio Padrão para previsão de eventos para diversos métodos de extração de características utilizando classificadores CLAS.

	AM-CHIP-CLAS					
Evento	Contagem (AUC)	SD	N-Gramas (AUC)	SD	Convolução (AUC)	SD
COZINHA	0.4993	0.0110	0.5197	0.0115	0.5191	0.0146
MESA	0.5427	0.1105	0.5507	0.1433	0.8192	0.1071
SOFA	0.5907	0.1044	0.5682	0.0454	0.7646	0.0823
VARANDA	0.5954	0.1409	0.6002	0.1100	0.5875	0.0936
HALL	0.6818	0.2289	0.4999	0.2707	0.9084	0.0737
ABAJOUR	0.5489	0.1093	0.7045	0.2738	0.7407	0.1824
	CHIP-CLAS					
Evento	Contagem (AUC)	SD	N-Gramas (AUC)	SD	Convolução (AUC)	SD
COZINHA	0.5000	0.0000	0.5000	0.0000	0.5000	0.0000
MESA	0.4977	0.0052	0.5000	0.0000	0.5000	0.0000
SOFA	0.5049	0.0211	0.4996	0.0169	0.5000	0.0000
VARANDA	0.4997	0.0006	0.4994	0.0013	0.5000	0.0000
HALL	0.4983	0.0019	0.5273	0.0177	0.6488	0.0058
ABAJOUR	0.5492	0.1107	0.5989	0.2234	0.8266	0.2085
	SVM RBF					
Evento	Contagem (AUC)	SD	N-Gramas (AUC)	SD	Convolução (AUC)	SD
COZINHA	0.7185	0.0315	0.7208	0.0248	0.7512	0.0250
MESA	0.9869	0.01556	0.9866	0.0170	0.9811	0.0236
SOFA	0.9041	0.0513	0.8918	0.0619	0.9140	0.0621
VARANDA	0.8334	0.1132	0.6462	0.1154	0.7727	0.1575
HALL	0.9670	0.0382	0.9477	0.0307	0.9609	0.0430
ABAJOUR	0.9672	0.0453	0.8963	0.2096	0.9455	0.0747

5.4.2 Discussão

Apesar do benefício de se evitar o ajuste de parâmetros por busca em *grid*, o desempenho do método CHIP-CLAS verifica-se notoriamente inferior ao do SVM em todos os casos. Conforme visto na seção 4.5, este comportamento pode se explicar pela

superposição dos dados somada a uma baixa quantidade de amostras para as classes minoritárias.

O aprendizado de métrica, como esperado, melhora o desempenho do método em quase todos os testes efetuados. Porém, a abordagem AM-CHIP-CLAS também não alcança desempenho superior ou semelhante ao SVM para os dados utilizados neste experimento.

5.5 Localização Não-Supervisionada de Usuários

Experimentos foram realizados para validar o modelo desenvolvido na seção 4.6 para localização de usuários em *Smart Homes* de forma não-supervisionada.

Os dados coletados do banco de testes não foram utilizados, visto que a dimensão do espaço de estados resultante seria incompatível com os recursos computacionais disponíveis. O modelo para 5 residentes e 8 possíveis estados por residente (7 cômodos + ausência) resultaria em 32768 estados.

Para validar o modelo, buscou-se então na literatura conjuntos de dados semelhantes com menor número de residentes.

Foram escolhidos os dados utilizados em (COOK; SCHMITTER-EDGECOMBE, 2009), produzidos ao longo de um ano em uma residência de dois moradores. Os dados selecionados provém de um conjunto de 51 sensores de movimento espalhados pela casa e do sensor que detecta abertura da porta principal. Dados provenientes de outros sensores foram descartados devido à ausência de sensores correspondentes no banco de testes, buscando assim uma maior semelhança.

Dada a ausência dos dados verdadeiros de localização de cada residente para cada amostra, a coerência dos resultados obtidos foi verificada por inspeção manual. Foram verificadas 10 sequências de 50 amostras, 20 antes e 30 após os 10 primeiros acionamentos do sensor que detecta abertura da porta principal.

5.5.1 Resultados

Mesmo sem o treinamento dos parâmetros, utilizando apenas os valores estimados a priori, o modelo para múltiplos residentes apresentou resultados coerentes com as anotações e com os sensores acionados para 8 das 10 sequências inspecionadas. Erros ocorrem quando algum dos residentes passa um longo período de tempo sem acionar nenhum sensor de movimento (por exemplo, ao dormir). Na ausência de um método para identificação dos residentes, não é possível diferenciar entre eles.

6 Conclusões e trabalhos futuros

Neste trabalho, é mostrada a viabilidade de se realizar a previsão de eventos com base nos dados já centralizados no sistema de automação residencial *Minibox*, da empresa *Neocontrol*. Os dados disponíveis, de acionamentos de interruptores, luzes, sensores infravermelhos de movimento e sensores de porta apresentam informação significativa, ainda que os sensores sejam pouco numerosos e não estejam presentes em todos os cômodos. O sistema para acumulação e disponibilização dos dados mostrou-se robusto e suficiente, disponibilizando dados que podem ser utilizados em tempo real.

Para o problema de previsão supervisionada de eventos, arquiteturas com desempenho satisfatório foram obtidas utilizando a classificação através de Máquinas de Vetores de Suporte (SVM) e extração de características das sequências através de Contagem de Símbolos, N-Gramas ou convolução com função linear.

Os experimentos preliminares realizados com a previsão utilizando classificadores CHIP-CLAS indicam desempenho não-satisfatório para estes, consistentemente inferior ao da SVM, mostrando o mau comportamento destes em conjuntos de dados altamente desbalanceados e com poucas amostras da classe minoritária. Tais resultados são preliminares, visto que estes modelos são recentes e ainda são objeto de pesquisa. Foi desenvolvida e validada com sucesso uma abordagem para melhorar este comportamento para conjuntos de dados com tais características, AM-CHIP-CLAS, através de aprendizado de métrica. Esta teve sucesso em melhorar significativamente o desempenho do classificador nestas condições, ainda que não tenha sido suficiente para alcançar o desempenho da SVM, mas motiva mais estudos sobre a regularização de modelos CLAS.

Os testes realizados com um *benchmark* de 15 bases, sem desbalanceamento e com dados suficientes, mostraram desempenho equivalente entre AM-CHIP-CLAS, CHIP-CLAS e SVMs com kernel Radial e Polinomial.

Para o problema da Localização de Usuários, foi desenvolvido uma estimativa para parâmetros de um Modelo Oculto de Markov Fatorial capaz de realizar esta tarefa através, sendo o problema então formulado como um problema de decodificação. O número de estados no modelo HMM equivalente aumenta exponencialmente com o quantidade de moradores da residência, não tendo sido possível sua utilização para os dados do banco de testes. Foi necessário para validação do modelo a utilização de dados da literatura coletados em residências com um menor número de moradores, e os resultados se mostraram coerentes.

6.1 Trabalhos Futuros

Para o problema da localização de usuários, é ainda necessário uma solução para o problema de decodificação do FHMM escalável com o número de residentes. Possíveis soluções são a utilização de algoritmos aproximados para solução do problema de Máxima Verossimilhança em FHMMs, como *Gibbs Sampling*, *Conditional Random Fields* ou *Mean Fields*. Outra possível solução envolve o desenvolvimento de representação esparsa dos estados do modelo HMM equivalente.

Para os algoritmos da família CLAS os resultados evidenciam que, apesar do desempenho equivalente ao SVM em bases bem comportadas, o processo de filtragem dos dados utilizado para bases com superposição não leva a bons resultados na presença de grande desbalanceamento e pouca representatividade da classe minoritária. O aprendizado de métrica proposto no trabalho contribuiu para a melhora do desempenho nestes casos, mas novas abordagens e heurísticas para esta otimização devem ser propostas e testadas. Com isso, será viabilizada a larga utilização do CHIP-CLAS como um classificador robusto sem necessidade do ajuste de hiperparâmetros.

Por fim, os experimentos presentes na literatura sugerem que os resultados para a previsão de características poderiam ser ainda melhores com a utilização de métodos baseados em Redes Neurais Artificiais para a classificação de sequências.

Referências

AIPPERSPACH, R.; COHEN, E.; CANNY, J. Modeling human behavior from simple sensors in the home. In: SPRINGER. *International Conference on Pervasive Computing*. [S.l.], 2006. p. 337–348. Citado na página 23.

ALBUQUERQUE TEIXEIRA, R. de et al. Improving generalization of MLPs with multi-objective optimization. *Neurocomputing*, v. 35, p. 189–194, 2000. ISSN 09252312. Citado na página 26.

BACHE, K.; LICHMAN, M. *UCI Machine Learning Repository*. 2013. 0 p. Disponível em: <http://www.ics.uci.edu/{˜}mlearn/MLRepository.ht>. Citado na página 42.

BAUM, L. E. et al. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *The annals of mathematical statistics*, JSTOR, v. 41, n. 1, p. 164–171, 1970. Citado na página 32.

CARR, R. S. et al. *Home energy monitoring and control system*. Google Patents, 1987. Disponível em: <https://www.google.com/patents/US4644320>. Citado na página 14.

CHAWLA, N. V. Data mining for imbalanced datasets: An overview. In: *Data mining and knowledge discovery handbook*. [S.l.]: Springer, 2009. p. 875–886. Citado na página 28.

CLEARY, J.; WITTEN, I. Data compression using adaptive coding and partial string matching. *IEEE transactions on Communications*, IEEE, v. 32, n. 4, p. 396–402, 1984. Citado na página 19.

COOK, D. J. Learning setting-generalized activity models for smart spaces. *IEEE intelligent systems*, NIH Public Access, v. 2010, n. 99, p. 1, 2010. Citado na página 17.

COOK, D. J. et al. Casas: A smart home in a box. *Computer*, IEEE, v. 46, n. 7, p. 62–69, 2013. Citado na página 22.

COOK, D. J.; SCHMITTER-EDGECOMBE, M. Assessing the quality of activities in a smart environment. *Methods of information in medicine*, NIH Public Access, v. 48, n. 5, p. 480, 2009. Citado 3 vezes nas páginas 17, 18 e 48.

CORTES, C.; VAPNIK, V. Support-Vector Networks. *Machine Learning*, v. 20, n. 3, p. 273–297, 1995. ISSN 15730565. Citado na página 25.

CRANDALL, A. S.; COOK, D. J. Tracking systems for multiple smart home residents. In: *Human Behavior Recognition Technologies: Intelligent Applications for Monitoring and Security*. [S.l.]: IGI Global, 2013. p. 111–129. Citado na página 20.

DAS, S. K. et al. The role of prediction algorithms in the mavhome smart home architecture. *IEEE Wireless Communications*, IEEE, v. 9, n. 6, p. 77–84, 2002. Citado na página 19.

DEMŠAR, J. Statistical Comparisons of Classifiers over Multiple Data Sets. *Journal of Machine Learning Research*, v. 7, p. 1–30, 2006. ISSN 1532-4435. Citado 3 vezes nas páginas 44, 45 e 46.

DUDA, R. O.; HART, P. E.; STORK, D. G. *Pattern Classification*. [S.l.: s.n.], 2000. 654 p. ISSN 1740634X. ISBN 978-0-471-05669-0. Citado na página 29.

GABRIEL Graph. In: WIKIPEDIA: the Free Encyclopedia. Wikimedia, 2018. Disponível em: <https://en.wikipedia.org/wiki/Gabriel_graph>. Acesso em: 09 jan. 2019. Citado 2 vezes nas páginas 26 e 27.

GABRIEL, K. R.; SOKAL, R. R. A New Statistical Approach to Geographic Variation Analysis. *Systematic Zoology*, v. 18, n. 3, p. 259–278, 1969. ISSN 00397989. Disponível em: <http://sysbio.oxfordjournals.org/cgi/content/abstract/18/3/259>. Citado na página 25.

GHAHRAMANI, Z.; JORDAN, M. I. Factorial hidden markov models. In: *Advances in Neural Information Processing Systems*. [S.l.: s.n.], 1996. p. 472–478. Citado na página 32.

GOLUB, T. R. Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring. *Science*, v. 286, n. 5439, p. 531–537, 1999. ISSN 00368075. Disponível em: <http://www.sciencemag.org/cgi/doi/10.1126/science.286.5439.531>. Citado na página 42.

HESS, K. R. et al. Pharmacogenomic predictor of sensitivity to preoperative chemotherapy with paclitaxel and fluorouracil, doxorubicin, and cyclophosphamide in breast cancer. *Journal of Clinical Oncology*, v. 24, n. 26, p. 4236–4244, 2006. ISSN 0732183X. Citado na página 42.

HUNT, J.; HOLMES, J. Electrical Energy Monitoring and Control System for the Home. *IEEE Transactions on Consumer Electronics*, C, n. 3, p. 578–583, 1986. Disponível em: <http://ieeexplore.ieee.org/xpls/abs{_}all.jsp?arnumber=4071>. Citado na página 14.

JAKKULA, V. R.; COOK, D. J. Using temporal relations in smart environment data for activity prediction. In: *Proceedings of the 24th International conference on machine learning*. [S.l.: s.n.], 2007. p. 20–24. Citado na página 19.

KASTEREN, T. V. et al. Accurate activity recognition in a home setting. In: ACM. *Proceedings of the 10th international conference on Ubiquitous computing*. [S.l.], 2008. p. 1–9. Citado 2 vezes nas páginas 17 e 18.

KOHAVI, R. A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. *Proceedings of the 14th International Joint Conference on Articial Intelligence (IJCAI)*, p. 1137–1145, 1995. ISSN 10450823. Citado 3 vezes nas páginas 36, 40 e 42.

LUNDSTRÖM, J.; JÄRPE, E.; VERIKAS, A. Detecting and exploring deviating behaviour of smart home residents. *Expert Systems with Applications*, v. 55, p. 429–440, 2016. ISSN 09574174. Citado 3 vezes nas páginas 24, 25 e 36.

MOZER, M. C. The neural network house: An environment that adapts to its inhabitants. *American Association for Artificial Intelligence Spring Symposium on Intelligent Environments*, n. December, p. 110–114, 1998. Citado na página 14.

OASIS. MQTT Version 3.1.1. *OASIS Standard*, n. October, p. 81, 2014. Disponível em: <http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html>. Citado na página 34.

PADMANABHAN, V. N.; MOGUL, J. C. Using predictive prefetching to improve world wide web latency. *ACM SIGCOMM Computer Communication Review*, ACM, v. 26, n. 3, p. 22–36, 1996. Citado na página 19.

PITKOW, J.; PIROLLI, P. Mining longest repeating subsequences to predict world wide web surfing. In: *Proc. UsENIX symp. on Internet Technologies and systems*. [S.l.: s.n.], 1999. p. 1. Citado na página 19.

RABINER, L. R.; JUANG, B.-H. An introduction to hidden markov models. *ieee assp magazine*, Citeseer, v. 3, n. 1, p. 4–16, 1986. Citado na página 31.

SINGLA, G.; COOK, D. J.; SCHMITTER-EDGECOMBE, M. Recognizing independent and joint activities among multiple residents in smart environments. *Journal of ambient intelligence and humanized computing*, Springer, v. 1, n. 1, p. 57–63, 2010. Citado na página 18.

SZEWCYZK, S. et al. Annotating smart environment sensor data for activity learning. *Technology and Health Care*, IOS Press, v. 17, n. 3, p. 161–169, 2009. Citado na página 17.

TAKAHASHI, C. C. *Mapeamento Explícito com Kernel em Aprendizado de Máquinas de Vetores de Suporte*. Dissertação (Mestrado) — Universidade Federal de Minas Gerais, 2015. Citado na página 25.

TAX, N. Human activity prediction in smart home environments with lstm neural networks. In: IEEE. *2018 14th International Conference on Intelligent Environments (IE)*. [S.l.], 2018. p. 40–47. Citado 3 vezes nas páginas 18, 19 e 20.

TORRES, L.; CASTRO, C.; BRAGA, A. A Computational Geometry Approach for Pareto-Optimal Selection of Neural Networks. *International Conference on Artificial Neural Networks*, n. 22, 2012. Citado 2 vezes nas páginas 26 e 36.

TORRES, L.; CASTRO, C.; BRAGA, A. Gabriel Graph for Dataset Structure and Large Margin Classification: A Bayesian Approach. *Proceedings of the European Symposium on Neural Networks 2015*, p. 237–242, 2015. Citado na página 26.

TORRES, L. et al. Distance-based large margin classifier suitable for integrated circuit implementation. *Electronics Letters*, v. 51, n. 24, p. 1967–1969, 2015. ISSN 0013-5194. Citado 2 vezes nas páginas 26 e 27.

TORRES, L. et al. A geometrical approach for parameter selection of radial basis functions networks. In: *Lecture Notes in Computer Science*. [S.l.: s.n.], 2014. v. 8681 LNCS. ISBN 9783319111780. Citado na página 26.

TORRES, L. C. B. *Classificador por Arestas de Suporte (CLAS): Métodos de Aprendizado Baseados em Grafos de Gabriel*. Tese (Doutorado) — Universidade Federal de Minas Gerais, 2016. Citado na página 25.

VITERBI, A. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE transactions on Information Theory*, IEEE, v. 13, n. 2, p. 260–269, 1967. Citado na página 32.

WEINBERGER, K. Q.; SAUL, L. K. Distance Metric Learning for Large Margin Nearest Neighbor Classification. *The Journal of Machine Learning Research*, v. 10, p. 207–244, 2009. ISSN 1532-4435. Citado na página 30.

WOLD, S.; ESBENSEN, K.; GELADI, P. Principal component analysis. *Chemometrics and intelligent laboratory systems*, Elsevier, v. 2, n. 1-3, p. 37–52, 1987. Citado na página 36.

XING, Z.; PEI, J.; KEOGH, E. A brief survey on sequence classification. *ACM SIGKDD Explorations Newsletter*, v. 12, n. 1, p. 40, 2010. ISSN 19310145. Citado 2 vezes nas páginas 22 e 23.

ZIV, J.; LEMPEL, A. Compression of individual sequences via variable-rate coding. *IEEE transactions on Information Theory*, IEEE, v. 24, n. 5, p. 530–536, 1978. Citado na página 19.