CEFET/RJ
DIRETORIA DE PESQUISA E PÓS-GRADUAÇÃO

DETECÇÃO DE ANOMALIAS EM TURBINAS EÓLICAS UTILIZANDO MODELOS BASEADOS EM DADOS

Fernando Pereira Gonçalves de Sá

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação, do Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, CEFET/RJ, como parte dos requisitos necessários à obtenção do título de mestre.

Orientador(a): Diego Nunes Brandão, D.Sc.
Coorientador(a): Rodrigo Franco Toso, Ph.D.

Rio de Janeiro,

Dezembro 2020

DETECÇÃO DE ANOMALIAS EM TURBINAS EÓLICAS UTILIZANDO MODELOS BASEADOS EM DADOS

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação, do Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, CEFET/RJ, como parte dos requisitos necessários à obtenção do título de mestre.

Fernando Pereira Gonçalves de Sá

Banca Examinadora:

Presidente, Dr. Diego Nunes Brandão, D.Sc. (CEFET/RJ) (Orientador(a))

Dr. Rodrigo Franco Toso, Ph.D. (Microsoft AI & Research) (Coorientador(a))

Dr. Anderson de Rezende Rocha, D.Sc. (UNICAMP)

Dr. Diego Barreto Haddad, D.Sc. (CEFET/RJ)

Rio de Janeiro,

Dezembro 2020

Ficha catalográfica elaborada pela Biblioteca Central do CEFET/RJ

S111 Sá, Fernando Pereira Gonçalves de
Detecção de anomalias em turbinas eólicas utilizando modelos baseados em dados / Fernando Pereira Gonçalves de Sá — 2020.
117f. + apêndice : il. color. , enc.

Dissertação (Mestrado) Centro Federal de Educação Tecnológica Celso Suckow da Fonseca , 2020.
Bibliografia : f. 106-117
Orientador: Diego Nunes Brandao
Coorientador: Rodrigo Franco Toso

1. Energia eólica. 2. Turbina eólica. 3. Lógica difusa. 4. Markov, Processos de. 5. Energia – Fontes alternativas. I. Brandao, Diego Nunes(Orient.). II. Toso, Rodrigo Franco (Coorient.). III. Título.

CDD 621.45

Elaborada pela bibliotecária Tania Mello – CRB/7 nº 5507/04

DEDICATÓRIA

Dedico este trabalho à VIDA, que foi tão desafiada no ano de 2020.

AGRADECIMENTOS

Este trabalho é o resultado de um intenso processo de amadurecimento, aprendizado e transição. Escolhi cursar o mestrado em Ciência da Computação em um momento que tinha poucas esperanças em seguir com a carreira de Engenheiro. Fui acolhido pelo CEFET e pelo orientador Dr. Diego Brandão, a quem sou muito agradecido principalmente por me motivar a aprender tantos assuntos diferentes. Posteriormente, a coorientação do Dr. Rodrigo Toso acrescentou novos desafios e entusiasmo pela pesquisa. Aprendi muito e, com humildade, reconheço que tenho muito a aprender sobre os assuntos discutidos neste trabalho, o que desejo que seja a semente de trabalhos futuros. Agradeço imensamente aos professores do CEFET que colaboraram de alguma forma em suas aulas para o meu amadurecimento acadêmico, em especial aqueles com quem dividi coautoria de artigos: Dr. Eduardo Ogasawara, Dra. Laura Assis e Dra. Rafaelli Coutinho. Agradeço igualmente à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES), que financiou esta pesquisa. Estendo minha gratidão à minha família, em especial à minha mãe, sempre presente nas minhas conquistas, bem como em meus fracassos.

RESUMO

Nos últimos anos, a energia eólica tornou-se tendência na substituição da matriz energética baseada em recursos não-renováveis. A produção dessa energia limpa é realizada pela turbina eólica, cuja operação reúne diferentes componentes que atuam na conversão da energia cinética do vento em energia elétrica. Contudo, a turbina eólica é uma máquina complexa de custo elevado, constantemente submetida a diferentes pressões que podem lhe causar falhas em algum momento. Neste contexto, o monitoramento contínuo dos diferentes componentes de uma turbina eólica permite a aplicação de técnicas de prognóstico de falhas baseadas na detecção de anomalias no sistema. Detecção, diagnóstico e prognóstico de falhas compreendem um conjunto de técnicas que garantem a confiabilidade, a segurança e a viabilidade econômica de um sistema. A presença de anomalias é o indício de que a saúde do sistema que compõe a turbina eólica está se deteriorando em função do tempo de operação, cuja evolução pode resultar brevemente em uma falha, quando ocorre a paralisação da produção de energia elétrica e são registrados muitas vezes danos irreversíveis no sistema. Diferentes técnicas foram desenvolvidas com o propósito de identificar essas anomalias. Neste trabalho, discutimos duas abordagens com esse propósito. Abordamos a detecção, diagnóstico e prognóstico de falhas baseados na classificação semi-supervisionada em uma configuração na qual o algoritmo de otimização multiobjetivo Algoritmo Genético de Ordenação não-dominante II (NSGA II) realiza a seleção automática de características e parâmetros de processamento. Uma segunda abordagem processou a detecção e diagnóstico de falhas baseadas na classificação de componentes em processo de pré-falha realizada pelos modelos ocultos de Markov. Ambas as abordagens mostraram-se eficientes em seus objetivos, considerando o processamento de um conjunto de dados reais imperfeito e de elevada dimensionalidade, que demandou diferentes métodos de pré-processamento. O Fluxo de Trabalho 1 apresentou resultados 13% superiores em relação ao trabalho de referência. Já o Fluxo de Trabalho 2, obteve F-*score* de até 0,89 no processamento da classificação multiclasse.

Palavras-chave: Energia Eólica; Detecção de Anomalias; Máquinas de Vetores de Suporte; Lógica Difusa; Modelos Ocultos de Markov

ABSTRACT

Anomaly Detection in Wind Turbines Using Data Driven Models

In recent years, the wind energy has become a trend in change of energetic matrix based on non-renewable resources. The production of the clean energy is made by wind turbine, whose operation is composed by different components that act together to convert cinetic energy from wind into electricity. However, the wind turbine is a complex and expensive machine constantly submitted to different pressures which lead to failures in a certain moment. In this context, the continous monitoring of the different wind turbine components enable the applying of techniques of failures prognosis based on anomaly detection in the system. Fault detection and diagnosis, as well as failure prognosis, comprise a wide set of techniques that goal to ensure the reliability, safety, and economic viability of the system. The presence of anomalies is the indication that the health of the system that comprise the wind turbine is deteriorating as operation time increases, whose evolution may lead to failure soon, when the power generation is interrupted and many damages to system may to be registered. Too many techniques was developed to indetify these anomalies. In this work, we discusse two approaches with this purpose. We approach deteccion, diagnosis and prognosis of failures based on semi-supervised classification in a setting adopting the algorithm of multiobjective optimization Non-dominated Sorting Genetic Algorithm II (NSGA-II) to select features and parameters processing automatically. A second approach processed fault detecion and diagnosis based on classification of components in pre-fault state using hidden markov models (HMM). Both approaches proved to be efficient in their objectives, once they handled imperfect real data of high dimensionality, what demanded an efficent pre-processing step. The workflow 1 presented results 13% better than reference work. In its turn, the workflow 2 got F-score of 0,89 in the multiclass classification.

Keywords: Wind Energy; Anomaly Detection; Support Vector Machine; Fuzzy Logic; Hidden Markov Model

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ALGORITMOS

LISTA DE CÓDIGOS

LISTA DE ABREVIATURAS E SIGLAS

CMS	Sistema De Monitoramento De Condições Da Máquina
EDP	Energias De Portugal
GWEC	Conselho Global De Energia Eólica
IED	Dispositivo Eletrônico Inteligente
KNN	K Vizinhos Mais Próximos
LDA	Análise Do Discriminante Linear
MLE	Estimador De Máxima Verossimilhança
MOEA	Algoritmos Evolucionários Multiobjetivo
MOGA	Algoritmos Genéticos Multiobjetivo
MOP	Problemas De Otimização Multiobjetiva
NSGA II	Algoritmo Genético De Ordenação Não-dominante II
OAPEC	Organização Dos Países Árabes Exportadores De Petróleo
PCA	Análise De Componentes Principais
RTU	Unidade De Terminal Remoto
SCADA	Sistema De Supervisão E Aquisição De Dados
SQL	Linguagem De Consulta Estruturada
SVD	Decomposição Em Valores Singulares

SUMÁRIO

1- Introdução

O Brasil possui uma matriz energética cada vez mais diversificada e majoritariamente mais limpa quando comparada com outros países [SANTOS, 2017; Jacinto, 2016]. A energia hidrelétrica ainda corresponde à maior fatia da geração da energia elétrica, com 63,7 %, mas fontes alternativas têm ampliado a sua participação. A energia eólica já corresponde à terceira maior fonte de energia elétrica no país, com 15 GW em operação e mais 4,6 GW já contratados[1] [Lucena and Lucena, 2019]. Com isso, o Brasil assume importante protagonismo no uso da energia eólica, possuindo empresas nacionais competentes no desenvolvimento de tecnologias neste segmento [2].

A abundância de ventos no Brasil permite a instalação de fazendas eólicas em diferentes regiões do país. Em zonas costeiras dos Estados do Nordeste, onde estão concentradas 85% de toda produção nacional de energia eólica, é comum encontrar turbinas eólicas distribuídas por toda a paisagem [Lucena and Lucena, 2019]. A mudança de paradigma, com base no investimento em energias renováveis e limpas, tende a aprofundar este quadro, até porque estima-se que o potencial de geração de energia eólica no país seja da ordem de 300 GW, o equivalente a mais de duas vezes o total da matriz energética atualmente instalada [3].

Contudo, à medida que avança o seu uso, crescem também os desafios para a melhora do desempenho na geração de energia eólica. Turbinas eólicas são sistemas complexos, caros e que demandam manutenção. Durante a sua operação, a turbina eólica está submetida a diferentes pressões internas e externas. Pressões internas referem-se ao desgaste de mecanismos, vibrações, corrosão, etc. Pressões externas referem-se aos fatores humanos, ambientais e de infraestrutura, tais como: imperícia do operador, acidentes, condições climáticas, falha no fornecimento de energia elétrica, etc. [Kidam and Hurme, 2013].

Quando a capacidade do equipamento para suportar tais pressões cessa, uma

[1] Disponível em: < http://www2.aneel.gov.br/aplicacoes/atlas/pdf/06-energia_eolica(3).pdf >. Acessado em: 17 ago. 2019

[2] Disponível em: <https://epocanegocios.globo.com/Empresa/noticia/2019/05/epoca-negocios-weg-lanca-novo-modelo-de-turbina-de-energia-eolica-com-potencia-de-4-mw.html >. Acessado em: 17 ago. 2019

[3] Disponível em: <http://casadosventos.com.br/pt/energia-dos-ventos/energia-eolica>. Acessado em: 17 ago. 2019

falha ocorre, o que provoca a paralisação do processo de geração de energia, um evento extremo que resulta em perdas financeiras significativas. Contrapondo a estes riscos iminentes, a área de O&M(Operações & Manutenção) concentra-se no monitoramento e controle das múltiplas variáveis que influenciam na operação da turbina eólica. Para tanto, diferentes metodologias foram criadas para lidar com essa tarefa desafiadora [Kobbacy and Murthy, 2008]. Uma dessas metodologias consiste na manutenção preditiva, abordada neste trabalho através do monitoramento contínuo de variáveis de operação das turbinas eólicas realizado pelo Sistema de Supervisão e Aquisição de Dados (SCADA). O monitoramento envolve o registro de todos os eventos que abrangem ocorrências normais e não padronizadas. Tipicamente, estes registros são armazenados em conjunto com as leituras dessas variáveis em um histórico que se torna posteriormente disponível para a aplicação de métodos de detecção de anomalias baseados em dados [Echevarría et al., 2019].

Neste trabalho, o processamento da detecção de anomalias envolveu a aplicação de dois métodos baseados em dados: máquinas de vetores de suporte com rótulos difusos e modelos ocultos de Markov.

As máquinas de vetores de suporte com rótulos difusos integram o Fluxo de Trabalho 1, baseado na seleção automática de características e parâmetros de processamento. O algoritmo de otimização multiobjetivo NSGA II opera essa seleção automática através de uma população de cromossomos continuamente aprimorada ao longo das gerações. Ao fim do processamento, este fluxo de trabalho fornece um conjunto ótimo de Pareto contendo soluções não dominadas. A partir desse conjunto ótimo de Pareto, extraímos análises baseadas no *tradeoff* dos objetivos definidos no processamento da otimização, a fim de selecionar o melhor modelo para solucionar o problema de detecção de anomalias nas turbinas eólicas.

Já os modelos ocultos de Markov executam, no Fluxo de Trabalho 2, a classificação de amostras pertencentes à operação normal e aos processos de pré-falha dos componentes da turbina eólica. Este processamento consiste em associar cada estado do modelo a uma dessas classes. Um método de seleção de características baseado no cálculo do Estimador de Máxima Verossimilhança (MLE), cujo valor também é fornecido pelos modelos ocultos de Markov, foi proposto neste trabalho. Com isso, os resultados da detecção e diagnóstico de falhas em evolução são processados segundo uma classificação multiclasse, da qual extraímos métricas para avaliarmos a sua efetividade.

Dessa forma, este trabalho contribui para a discussão de métodos eficazes na identificação de anomalias durante a operação de turbinas eólicas utilizando os dados adquiridos pelo SCADA. Essa discussão é justificada pela necessidade de aprimorar os sistemas de detecção de anomalias que devem atender requisitos cada vez mais complexos dada a evolução dos sistemas supervisores e o avanço da tecnologia das turbinas eólicas para novas fronteiras de aplicação [Keivanpour et al., 2017; Hayes, 1977; Stetco et al., 2018].

1.1- Trabalhos Relacionados

A detecção de anomalias em turbinas eólicas é um problema sobre o qual foram desenvolvidas diferentes abordagens em trabalhos publicados recentemente. Este assunto é tão relevante que uma pesquisa na base Scopus com o termo de busca "(wind turbine) and (fault detection)" revela 1490 artigos sobre o tema, sendo 133 artigos de 2020 e 6 artigos já para 2021. Observando a quantidade excessiva de artigos relacionados, buscamos apresentar os artigos mais relevantes encontrados nos últimos cinco anos e indicamos duas revisões bibliográficas que podem auxiliar os leitores interessados sobre o tema.

Bangalore and Tjernberg [2015] utilizam redes neurais artificiais para a detecção de anomalias em turbinas eólicas. Os autores utilizam a caixa de velocidade dos rolamentos do gerador como objetos de análise para identificar as anomalias. Já Tautz-Weinert and Watson [2016] apresentam uma revisão sobre o tema, na qual discutem os diferentes modelos de detecção de anomalias adotados em trabalhos recentes. Em [Pandit and Infield, 2018], o modelo Gaussiano foi utilizado para estimar a curva de geração de energia elétrica no período. Neste trabalho, a detecção de anomalias é centrada em falhas no sistema de direção das pás, cuja influência na curva de potência gerada dispara o alerta quando se afasta substancialmente da predição realizada pelo modelo. No trabalho elaborado por [Li et al., 2019], os modelos ocultos de Markov são empregados na modelagem da deterioração da confiabilidade do sistema da turbina eólica por meio do monitoramento do desempenho de parâmetros relevantes para o processamento. Nesta modelagem, os estados transitórios e ocultos do modelo são associados à condição de

saúde do sistema, a qual evolui em direção à falha. Stetco et al. [2018] apresentam uma revisão minuciosa de abordagens recentes envolvendo aprendizado de máquina na construção de soluções de detecção de anomalias em diferentes componentes das turbinas eólicas.

Em [Chen et al., 2019], os autores empregam redes neurais de aprendizado profundo na construção de um modelo de detecção de processos de pré-falha nas pás do rotor causados pelo congelamento da estrutura. O trabalho soluciona o problema do conjunto de dados desbalanceado pela proposição de uma nova metodologia de extração de características baseada em redes neurais artificiais profundas, caracterizada como eficaz pelos autores na preservação das informações contidas originalmente nas classes. O trabalho desenvolvido por [Mao et al., 2020] aborda um novo método de detecção de anomalias em rolamentos da turbina eólica utilizando uma arquitetura de processamento em tempo real baseada em classificação semi-supervisionada. Este método de classificação semi-supervisionada discutido no trabalho é construído baseado no algoritmo convencional das máquinas de vetores de suporte, a exemplo do algoritmo que utilizamos neste trabalho. Em him mais uma revisão sobre o tema é apresentada, na qual os autores discutem também sobre novos desafios da área e perspectivas futuras.

1.2- Contribuições

A pesquisa bibliográfica forneceu importantes insumos para a proposição de soluções alinhadas com as tendências recentes na área de detecção de anomalias em turbinas eólicas. Com esse objetivo, propomos dois fluxos de trabalho que alcançaram ou resultados superiores em relação à referência ou incluíram abordagens não discutidas pela referência.

O Fluxo de Trabalho 1 propõe a seleção automática de características e parâmetros de processamento do aprendizado como um problema de otimização multiobjetivo aplicando o NSGA II, que tem sido amplamente adotado em problemas dessa natureza. Por meio do processamento da detecção e classificação semi-supervisionada, realizada pelas máquinas de vetores de suporte com rótulos difusos, anomalias em dados de turbinas eólicas são detectadas e posteriormente avaliadas em uma segunda classificação rea-

lizada pelo algoritmo convencional das máquinas de vetores de suporte. Este fluxo de trabalho envolve dois conceitos que têm recebido atenção em trabalhos recentes: (a) seleção automática de características baseada em algoritmo genético multiobjetivo; (b) classificação semi-supervisionada. A proposição desse fluxo de trabalho compreende a nossa primeira contribuição.

O Fluxo de Trabalho 2 baseia-se no problema de decodificação dos estados dos modelos ocultos de Markov para a realização da classificação multiclasse envolvendo as componentes da turbina eólica. As amostras processadas nessa classificação são obtidas pelo agrupamento de subsequências das séries temporais contendo observações da operação normal das turbinas eólicas e observações de processos de pré-falha de componentes deste sistema. Diferente da referência, o nosso fluxo de trabalho lidou com dados reais segundo uma série temporal multivariada e multidimensional. O nosso fluxo de trabalho também inclui a proposição de uma metodologia de seleção de características baseada no cálculo da máxima verossimilhança, cujo cálculo também foi realizado pelos modelos ocultos de Markov.

1.3- Organização

Este trabalho está dividido em mais 7 capítulos. O Capítulo 2 discute os fundamentos da detecção de anomalias, conceitos, métodos e abordagens desenvolvidas. O Capítulo 3 apresenta o algoritmo máquinas de vetores de suporte com rótulos difusos, adotado no processamento da classificação semi-supervisionada para detecção, diagnóstico e prognóstico de falhas em turbinas eólicas. O Capítulo 4 apresenta os modelos ocultos de Markov, adotado no processamento da classificação multiclasse de processos de pré-falha em componentes para detecção e diagnóstico de falhas em turbinas eólicas. O Capítulo 5 define o conceito de redução de dimensionalidade, operação essencial em um conjunto de dados que possui muitas variáveis, discutindo este conceito no contexto de um problema de otimização multiobjetivo. O Capítulo 6 define a metodologia dos dois fluxos de trabalho desenvolvidos para a detecção de anomalias em turbinas eólicas. O pré-processamento realizado, as ferramentas e métricas adotadas também são discutidas neste capítulo. O Capítulo 7 apresenta os resultados do trabalho, baseados na metodolo-

gia apresentada. As considerações finais são apresentadas no Capítulo 8; este capítulo também apresenta uma discussão do desenvolvimento realizado e aponta caminhos futuros deste trabalho.

2- Detecção, Diagnóstico e Prognóstico de Falhas em Turbinas Eólicas

Mudanças de paradigma direcionam o mundo para a adoção de novas fontes energéticas. A energia eólica surge como uma promissora alternativa, cuja maturidade tecnológica impulsiona sua aplicação em diferentes cenários. Contudo, a viabilidade econômica e a eficiência operacional estão diretamente relacionadas à disciplina de manutenção de máquinas, que assegura a operação do sistema dentro de condições normais. Tecnologias de monitoramento e supervisão, como o SCADA, são aliados na missão de detectar falhas em evolução e fornecer subsídios para técnicas de diagnósticos e prognósticos. Discutimos todos esses aspectos neste capítulo, que sistematiza o problema abordado neste trabalho.

2.1- Introdução

O mundo vive a transição da era do petróleo para o uso de energias limpas [Hayes, 1977]. Este processo tem se aprofundado nos últimos anos à medida que a crescente eficiência das chamadas fontes energéticas alternativas reduz seus custos [Li et al., 2015]. Dentre essas fontes de energias alternativas (ao petróleo), a utilização da energia eólica obteve relevante crescimento nos últimos anos. Mesmo assim, estamos distantes de alcançar a capacidade máxima do seu uso, conforme estimativa realizada por Lu et al. [2009], que determinaram a existência de um potencial útil de geração de energia eólica na ordem de 123 PetaWatt-hora (PWh) em todo o planeta, já considerando restrições discutidas no trabalho, o que representa 7 vezes o atual consumo de energia global.

Podemos identificar diferentes eventos que moldaram a história recente da energia eólica [Kaldellis and Zafirakis, 2011]. Os impactos gerados pelas duas grandes guerras mundiais no comércio do petróleo, elevando seu preço, ampliaram a popularidade da energia eólica em parte da Europa, fato que fez surgirem turbinas eólicas com diferentes

configurações em variados estágios tecnológicos [Jain, 2011].

Sempre após o fim das guerras, com a normalização do comércio e consequente redução do preço do petróleo, novamente a energia eólica perdia espaço para fontes convencionas de geração de energia. O mesmo declínio foi observado nos Estados Unidos, onde o Ato de Eletrificação Rural de 1936, ao expandir a rede elétrica em direção ao interior do país, frustrou as primeiras tentativas de consolidar a energia eólica como fonte de suprimento de energia elétrica dos povoados rurais. Uma mudança neste jogo só ocorreria em 1973, quando a deflagração da Crise do Petróleo orientou novamente a visão estratégica para as fontes de energia alternativas. O embargo, imposto pelos países da Organização dos Países Árabes Exportadores de Petróleo (OAPEC), provocou um choque de demanda que colocou em xeque a segurança energética dos países ocidentais [Kaldellis and Zafirakis, 2011].

A corrida tecnológica fez, desde então, surgirem novos protótipos de turbinas eólicas, com diferentes topologias de projeto, maiores, mais robustas e eficientes na conversão de energia mecânica em elétrica [Letcher, 2017]. Este movimento, que se intensificou principalmente a partir da década de 1980, gerou produtos comercialmente viáveis que determinaram o avanço tecnológico do sistema que compõe as turbinas eólicas, como mostra a Figura 1.

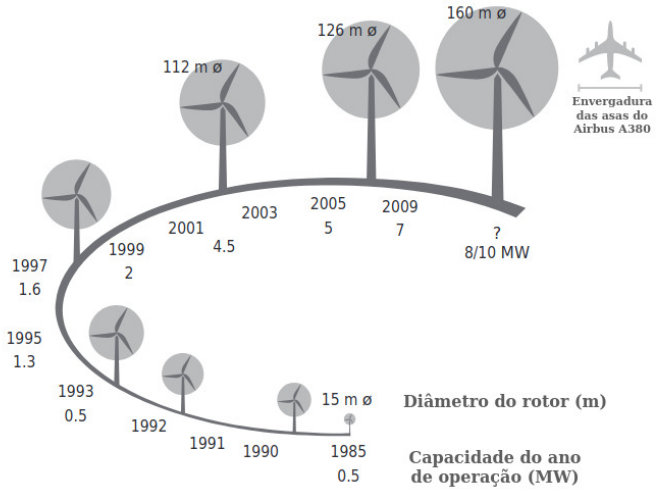

Figura 1 – Evolução tecnológica do sistema da turbina eólica. Adaptado de [Tavner, 2012].

Consolidando a tendência tecnológica, a opinião pública mostra-se cada vez mais crítica às fontes não-renováveis de energia e disposta a integrar os esforços para concluir essa transição energética. O resultado desse processo pode ser avaliado pela leitura

dos relatórios anuais produzidos pelo Conselho Global de Energia Eólica (GWEC), que apontam frequentemente um vigoroso crescimento da adoção da energia eólica nos próximos anos [Council, 2019]. O Brasil é um exemplo de país que vem apresentando tal crescimento, com destaque para os Estados da região nordeste do país [Rocha et al., 2012]. A Figura 2 mostra um parque eólico localizado no Estado do Ceará [1].

Figura 2 – Parque eólico localizado no Estado do Ceará, Brasil.

Aliada à ampliação da capacidade de geração instalada, novas tecnologias estão sendo aplicadas a fim de aumentar a eficiência, reduzir perdas e aumentar a confiabilidade das turbinas eólicas [Canizo et al., 2017]. O desenvolvimento de técnicas de detecção de anomalias em turbinas eólicas foi tema de trabalhos recentes [Márquez et al., 2012; Amirat et al., 2009; Almalawi et al., 2015; Habibi et al., 2019], entre outros. Os dados adquiridos por algum sistema supervisório, durante o monitoramento das condições da turbina eólica, alimentam um sistema especialista que se encarrega de detectar as anomalias. Em geral, as abordagens de diagnóstico e prognóstico de falhas em turbinas eólicas concentram-se em dados gerados pela observação de algum componente crítico, nos arquivos de registro ou na curva de potência [Tautz-Weinert and Watson, 2016; Kusiak and Verma, 2012; Schlechtingen et al., 2013]. Com isso, é possível construir sistemas analíticos e preditivos, que se antecedem à ocorrência de alguma falha grave, o que representa prejuízo em diferentes dimensões.

A Seção 2.2 discute aspectos técnicos relevantes da turbina eólica, a fim de compreendermos o significado das anomalias encontradas pelo sistema especialista.

[1] Disponível em:https://sengece.org.br/ceara-esta-entre-os-tres-estados-brasileiros-com-maior-capacidade-instalada-de-energia-eolica/. Acessado em: 17 ago. 2019

2.2- Turbinas Eólicas

A turbina eólica é composta por diferentes componentes e engrenagens, cuja operação tem a capacidade de converter a energia mecânica de rotação das pás em energia elétrica [Wang et al., 2014]. Aqui discutimos alguns aspectos técnicos fundamentais de funcionamento.

Em relação ao eixo de rotação, as turbinas eólicas podem ser classificadas em horizontais ou verticais. As turbinas eólicas horizontais possuem o eixo de rotação paralelo ao solo e ao fluxo de vento. Por sua vez, nas turbinas eólicas verticais o eixo de rotação é perpendicular ao solo. As turbinas eólicas horizontais oferecem elevadas eficiência e densidade de potência, baixo limiar de corte da velocidade do vento e satisfatória relação potência/custo. Já as turbinas eólicas verticais possuem a vantagem de aproveitar ventos de qualquer direção, o que dispensa a necessidade do sistema de guinada, além de possuir instalação mais simples por não necessariamente embarcar na torre outros componentes do sistema [Tong, 2010].

Neste trabalho, o conjunto de dados ao qual tivemos acesso nos forneceu leituras de diferentes parâmetros adquiridos durante a operação de turbinas eólicas horizontais. Independente da direção do eixo de rotação do rotor, a sua execução, após a ocorrência de um fluxo de vento, ativa uma complexa rede de mecanismos e dispositivos mecânicos/eletrônicos que atuam na transformação do seu movimento em energia elétrica. O diagrama da Figura 3 apresenta uma visão geral dos principais mecanismos do sistema que compõe a turbina eólica.

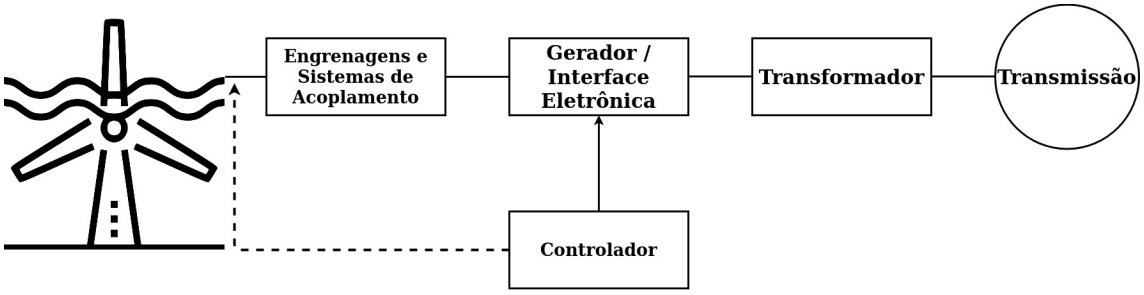

Figura 3 – Diagrama dos principais componentes em um sistema de turbina eólica. Adaptado de [Tong, 2010].

São descritos abaixo os componentes mostrados na Figura 3, de acordo com [Tong, 2010; Letcher, 2017]:

- Ativação do sistema: o giro do rotor, impulsionado por duas ou três pás, captura a energia mecânica do vento para transformá-la em energia elétrica;

- Engrenagens e sistemas de acoplamento: o eixo do rotor é acoplado a um sistema de transmissão composto por freios e pela caixa de velocidade. O freio é acionado por diferentes motivos: (a) durante tormentas climáticas que sobrecarregariam o sistema girando o rotor em alta velocidade; (b) durante o reinício do sistema; (c) e alguma paralisação necessária, para manutenção, por exemplo. A caixa de velocidade atua como um multiplicador da velocidade de rotação do eixo do rotor durante a transmissão do movimento para o gerador. Assim, através de engrenagens, a caixa de velocidade consegue transformar uma rotação em baixa velocidade do rotor em uma rotação com velocidade suficiente para a geração de energia pelo gerador;

- Gerador/Interface eletrônica: o gerador é um componente eletromecânico que desempenha a conversão da energia mecânica em energia elétrica. Dois componentes integram o gerador: estator e o rotor. O estator é um nicho fixo que abriga bobinas formadas por fios condutores enrolados. O giro do rotor excita essas bobinas do estator, gerando um campo eletromagnético que induz uma tensão elétrica. A corrente elétrica é o resultado deste processo. A interface eletrônica atua entre o gerador e a rede de transmissão. Seu objetivo é conciliar ambos os requerimentos do gerador e da rede de transmissão, respeitando o limite de custos e soluções de manutenção. De um lado, ela garante o ajuste da velocidade de rotação do gerador, extraindo o máximo de energia; por outro lado, ela controla as potências ativa e reativa, frequência e tensões de saída;

- Controlador: o controlador tem o objetivo de manter a turbina eólica sob operação normal utilizando meios ativos e passivos para tal. As variáveis continuamente monitoradas, em especial as velocidades do vento e do rotor, potências ativa e reativa, tensão, etc., possuem limiar de operação definidos que, quando extrapolados, agem como gatilhos para ações que podem até mesmo interromper o funcionamento da turbina eólica;

- Transformador: presente na extremidade do diagrama, o transformador reduz a tensão de saída do sistema da turbina eólica para conectá-la à rede de distribuição,

evitando, assim, uma sobrecarga na rede que poderia danificar equipamentos conectados a ela.

A Figura 4 exibe detalhadamente o posicionamento desses diferentes subsistemas abrigados no corpo da nacele em uma turbina eólica horizontal.

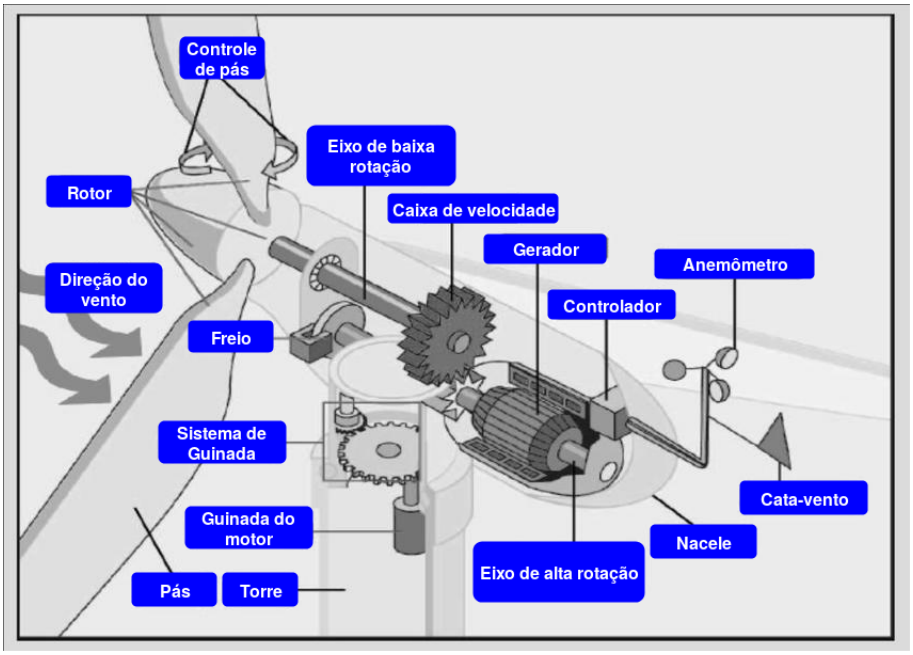

Figura 4 – Componentes de uma turbina eólica. Adaptado de [Ayad, 2009].

A expansão da energia eólica pressiona cada vez mais por inovação, a fim de alcançar maior eficiência durante a operação das turbinas eólicas. Dessa forma, cada componente crítico do sistema que compõe a turbina eólica é monitorado por diferentes sensores conectados a um sistema supervisório, a fim de alimentá-lo com leituras de parâmetros relacionados à velocidade de rotação das pás, velocidade do vento, temperatura de componentes, oscilação, geração de energia, etc. [Li et al., 2015].

Os dados históricos adquiridos durante o monitoramento são posteriormente analisados com a finalidade de identificar desvios da operação padrão. Esses desvios pertencem ao espectro das anomalias, caracterizadas como a extrapolação de valores esperados nos parâmetros monitorados de um sistema. Ainda que representem uma pequena fração das observações armazenadas em um conjunto de dados, seu conhecimento é vital para a compreensão do comportamento do sistema monitorado. A Seção 2.3 aprofunda-se na definição desse conceito, fundamental para a compreensão da necessidade de implementação de um sistema de detecção de anomalias em turbinas eólicas.

2.3- Anomalias como Fonte de Conhecimento do Sistema

Os dados obtidos pelos sensores formam uma série temporal multivariada e multidimensional $\boldsymbol{TS}$, formada por pontos normais, ruídos e *outliers*, como representado pela Figura 5. Também merece destaque o fato de que dificilmente o rótulo que identificaria a qual categoria pertence cada ponto $TS_i \in \boldsymbol{TS}$ é fornecido, seja por uma característica do sistema de aquisição de dados, pelo custo associado para realizar a rotulagem ou pela natureza do processo monitorado [Chandola et al., 2009].

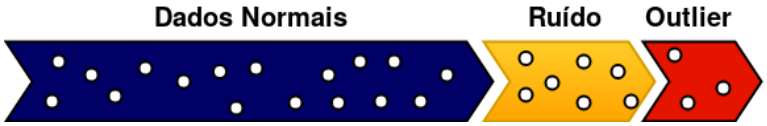

Figura 5 – Categorização dos pontos de um vetor de observações $\boldsymbol{TS}$ como ponto normal, ruído e *outlier*.

Os pontos identificados como ruído e *outlier* representam desvios do comportamento normal do sistema. Diferentes do ruído, os *outliers* indicam observações interessantes do sistema, cuja avaliação poderá gerar algum conhecimento. Os *outliers* representam eventos raros ou "observações que se desviam tanto de outras observações que despertam a suspeita de terem sido gerados por um mecanismo diferente daquele que gerou as observações normais do conjunto de dados", conforme descrito por Hawkins [1980].

O *outlier* pode ser classificado, de acordo com [Pathan, 2014; Singh and Upadhyaya, 2012], como:

- *Outlier* pontual: o ponto é considerado anômalo quando comparado ao restante dos dados. É a situação mais intuitiva durante a identificação de *outliers*, uma vez que estes pontos se destacam em relação ao agrupamento de pontos normais;
- *Outlier* contextual: o ponto é anômalo em um específico contexto em que ele se manifesta (mas não em outro). Como característica, este tipo de *outlier* possui seu valor inserido no intervalo de valores dos dados normais;
- *Outlier* coletivo: consiste na manifestação periódica de uma faixa ou grupo de pontos que, se forem analisados de forma desassociada, confundem-se com pontos normais pela similaridade entre ambos.

Sob o ponto de vista estatístico, o mecanismo que gerou o conjunto de observações TS é interpretado como uma distribuição de probabilidade de cauda longa, como *Student*, por exemplo. A possibilidade de surgimento de *outliers* neste conjunto de dados é determinada pela ação de distribuições de probabilidade propensas a *outlier* ou resistentes a *outlier* [Gather and Rauhut, 1990; Hawkins, 1980]. Conhecer o mecanismo que gerou o *outlier* facilitaria a criação de um modelo estatístico para descrever o processo que distinguiria o comportamento anômalo do normal [Hawkins, 1980]. Contudo, na maioria das vezes não se possui qualquer domínio de conhecimento sobre a distribuição de probabilidade responsável pela geração dos dados [Ramaswamy et al., 2000]. Assim, tornou-se necessário o desenvolvimento de abordagens alternativas para realizar a detecção de *outliers*.

A detecção de *outliers* é um problema fundamentalmente não-supervisionado [Hodge and Austin, 2004]. De acordo com Echevarría et al. [2019], os métodos de detecção de *outliers* podem ser divididos em dois grandes grupos: métodos baseados em modelos físicos ou matemáticos; e métodos que não utilizam quaisquer modelos. Os métodos baseados em dados correspondem a este segundo grupo e são caracterizados por realizarem o processamento utilizando dados históricos do problema. De acordo com [Hawkins, 1980; Wang et al., 2019; Kannan and Somasundaram, 2015], é possível estabelecer seis grandes abordagens para a solução do problema de detecção de *outliers*:

1. Detecção de *outlier* baseada na estatística: corresponde às primeiras técnicas desenvolvidas para detectar *outliers*. Em seu escopo há técnicas paramétricas e não-paramétricas [Niu et al., 2011]. Os métodos paramétricos demandam conhecimento *a priori* a respeito da distribuição de probabilidade dos dados para estimar os seus parâmetros utilizando MLE (*maximum likelihood estimation*). Em abordagens ingênuas, ferramentas básicas como *z-score* ou *box-plot* podem ser utilizadas para determinar quais pontos estariam além do intervalo de tolerância em um nível de significância dada a distribuição de probabilidade de cauda longa [Hawkins, 1980]. Técnicas não-paramétricas, por sua vez, não exigem nenhum conhecimento *a priori* sobre a distribuição de probabilidade dos dados. Posicionam-se nesta abordagem os modelos ocultos de Markov, propostos inicialmente por Baum and Petrie [1966] para o processamento de séries temporais. Outros métodos estatísticos paramétricos e não-paramétricos são discutidos em [Hawkins, 1980];

2. Detecção de *outlier* baseada em distância: nesta abordagem, os pontos correspondentes aos *outliers* são determinados a partir de uma medida de distância em relação aos pontos normais. Essa abordagem é adotada em problemas supervisionados e não-supervisionados. A técnica K Vizinhos Mais Próximos (KNN) é a mais utilizada. Ela cria partições no conjunto de dados, separando os *outliers* daqueles pontos normais [Hawkins, 1980];

3. Detecção de *outlier* baseada em clusterização: este conjunto de técnicas particiona o conjunto de dados em K agrupamentos a fim de identificar aqueles referentes aos *outliers*. O algoritmo K-Means é o mais empregado para realizar esta tarefa [He et al., 2003];

4. Detecção de *outlier* baseada em densidade espacial: as técnicas baseadas em densidade espacial particionam o conjunto de dados a partir da estimação da densidade de pontos na vizinhança de cada instância. As partições são formadas em torno de pontos centrais e pontos periféricos, que se conectam respeitando determinados parâmetros. O algoritmo DBSCAN é um algoritmo que realiza essa tarefa [Çelik et al., 2011];

5. Detecção de *outlier* baseada em redes neurais: redes neurais são um modelo não-paramétrico cujos parâmetros da rede são ajustados para realizar o processamento da classificação mesmo quando os limites entre as classes são complexos de se distinguir [Hawkins et al., 2002];

6. Detecção de *outlier* baseada em classificação semi-supervisionada: adota um algoritmo de classificação semi-supervisionado para identificar *outliers*. Neste trabalho, as máquinas de vetores de suporte para classificação binária utilizam a lógica difusa para determinar o grau de crença pelo qual uma amostra pode ser classificada como *outlier* [Zhao et al., 2016].

O resultado do processamento de detecção de *outliers* fomenta interpretações de acordo com a aplicação que originou o conjunto de dados. Em sistemas bancários, pode representar a ocorrência de fraude em cartão de crédito. Na medicina diagnóstica, o *outlier* é o indicativo de alguma célula tumoral ou alguma outra condição clínica. Em suma, os *outliers* recebem diferentes denominações de acordo com o campo de aplicação.

Neste trabalho, os *outliers* serão tratados como sinônimo de anomalias ocorridas durante a operação da turbina eólica.

A construção de métodos de detecção de anomalias baseados em dados demanda a existência de algum sistema de aquisição de dados. Essa tarefa usualmente é realizada por um sistema de controle e supervisão. Tal sistema é responsável por armazenar as leituras dos sensores e detectar anomalias no processo em operação. Nesta função de detecção, ele também fica responsável por realizar ações que evitam o acúmulo de degradação da saúde dos componentes, que muitas vezes culminam na paralisação das operações do sistema, como ilustra a Figura 6.

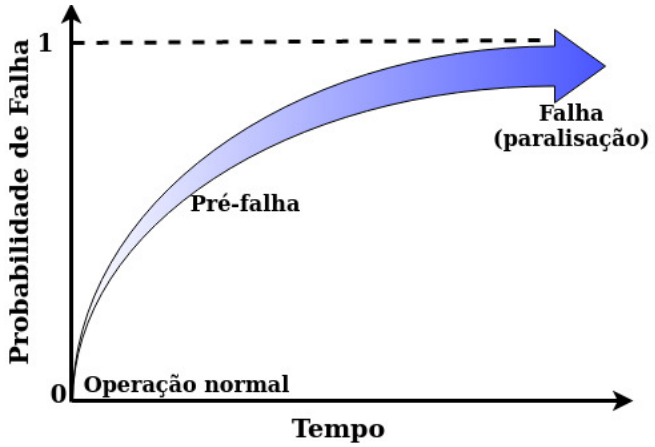

Figura 6 – Degradação do estado de saúde de um sistema em função do tempo. O sistema evolui de um estado de operação normal para o estado de pré-falha, que culminará em falha se nenhuma ação for realizada.

Como observa Maxion [1990], detecção, diagnóstico e reparo formam a tríade das ações de resposta a falhas. Por isso, além da identificação precoce de futuras causas de falhas, é preciso haver um consistente regime de manutenção dos componentes do sistema para realizar reparos de forma eficaz. Estes temas são abordados na Seção 2.4 durante a discussão sobre os fundamentos de sistemas de detecção de anomalias em componentes de turbinas eólicas.

2.4- Sistema de Detecção de Anomalias

Por séculos, a única forma de aprender sobre o mau funcionamento e localização dos problemas em máquinas era baseada em impressões de senso biológico, como mudança de formato e cor, sons incomuns, calor ou vibração, etc. [Gertler, 1998]. Já os métodos clássicos de monitoramento e proteção automática, baseiam-se na verificação se a leitura dos parâmetros monitorados ultrapassou algum limiar tolerado. Contudo, esta abordagem, embora simples e confiável, é sensível somente a bruscas variações, sendo incapaz de mensurar mudanças graduais no gradiente do risco de falha, conforme a Figura 6. Além disso, a identificação da causa da falha torna-se pouco prática em sistemas mais complexos monitorados com múltiplos sensores, o que significa o acionamento simultâneo de vários alarmes [Isermann, 2005]. Abordagens recentes que envolvem técnicas de reconhecimento de padrões, processamento de sinais, entre outras técnicas baseadas em dados, são capazes de identificar sinais incipientes de falhas com elevada precisão [Tautz-Weinert and Watson, 2016; Qiao and Lu, 2015; Zaher et al., 2009].

O sistema de detecção de anomalias é o resultado da evolução não só tecnológica mas também conceitual no campo da manutenção industrial. Diante da consolidação do uso de sistemas de informação, redução do quadro de funcionários, elevada exigência por qualidade, eficiência e redução de custos, é difícil imaginar que até há pouco tempo não havia consenso sobre a necessidade de antever graves problemas que causariam paralisação e perdas financeiras [Byrne et al., 1995]. A percepção do valor das operações associadas às manutenções e da economia que elas trazem causaram a ruptura das práticas comumente adotadas para a adoção de uma nova filosofia embasada no conhecimento empírico e na adoção de tecnologias de telemática [Kobbacy and Murthy, 2008]. Essa mudança representa importante avanço para a sociedade. Estamos rodeados por sistemas elétricos, pneumáticos, equipamentos, veículos, etc. A evolução da abordagem de O&M representa mais conforto, segurança, confiabilidade e mais valor para serviços básicos.

A Figura 7 apresenta os paradigmas e os ganhos proporcionados pela sua evolução [Kobbacy and Murthy, 2008]. Os paradigmas são discutidos a seguir:

(i) Nenhuma manutenção realizada: Esta perspectiva abrange casos ainda hoje válidos nos quais envolve alguma operação especial, para o qual não foi desenvolvida ainda

uma técnica efetiva de manutenção ou então máquinas que foram desenvolvidas para serem utilizadas uma única vez, cuja manutenção representaria um custo muito elevado;

(ii) Manutenção reativa: a manutenção reativa posiciona-se no senso comum, segundo o qual a manutenção deve ser realizada quando a "máquina quebra". É uma prática ineficiente e indica pouco conhecimento sobre o comportamento do equipamento, que opera até a ocorrência de uma falha. A manutenção reativa representa a ocorrência periódica de reparos emergenciais, o que gera irreparável perda de produtividade;

(iii) Manutenção preventiva: a manutenção preventiva representa uma transição importante em relação à manutenção reativa. A manutenção preventiva possui como estratégia a substituição, revisão ou remanufaturação de componentes a cada intervalo de tempo. Nesta perspectiva, são utilizadas ferramentas estatísticas para estabelecer o intervalo adequado para a ocorrência da manutenção, baseado no tempo de operação, meia-vida de componentes, entre outros parâmetros. Entretanto, o paradigma falha por não ter informações constantemente atualizadas sobre o estado atual do equipamento;

(iv) Manutenção preditiva: a manutenção preditiva reúne recursos humanos e tecnológicos para avaliar em processos continuamente monitorados se algum limiar de decisão foi alcançado, o que representa a necessidade de realização da manutenção;

(v) Manutenção proativa: a manutenção proativa é um conceito que envolve monitoramento remoto, distinção entre o diagnóstico e o prognóstico da falha e gerenciamento do ciclo de vida da máquina repercutindo em aspectos do ciclo de vida do produto. O aspecto principal da manutenção proativa é a busca pela causa raiz da falha que pode ocorrer, reservando especial atenção a este mecanismo;

(vi) Auto-manutenção: o conceito de auto-manutenção é emergente e envolve a aplicação de diferentes técnicas para dispor a máquina de inteligência, a fim de eliminar a imperícia humana do processo. Neste contexto, são bem vindos os conceitos trazidos pela indústria 4.0, que une o *big data* com os recursos ciberfísicos para realizar o gerenciamento do processo industrial, incluindo o planejamento eficaz de manutenção das máquinas interconectadas e constantemente monitoradas [Lee

et al., 2014].

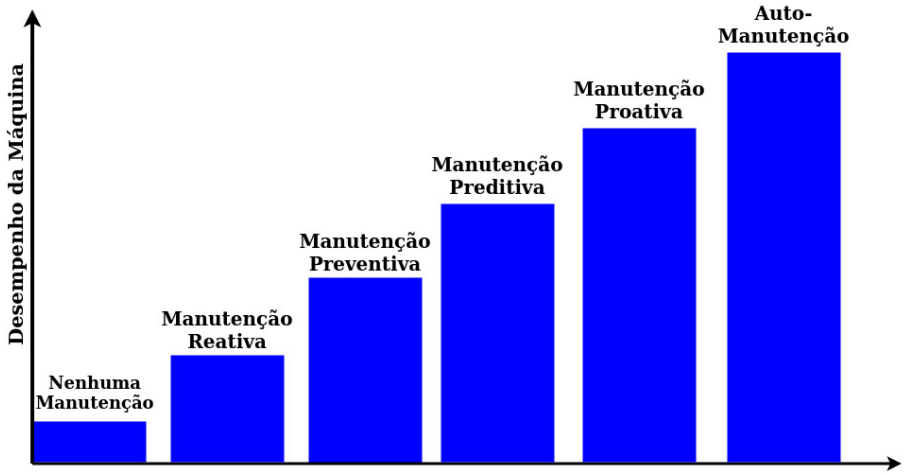

Figura 7 – Paradigmas de manutenção. Adaptado de [Kobbacy and Murthy, 2008]

Neste trabalho, o conjunto de dados $\mathcal{D}$ contém informações obtidas do sistema de monitoramento SCADA , definido como um *software* que centraliza o monitoramento e controle de processos críticos. Os dados que alimentam este sistema originam-se de um complexo conjunto de sensores, controladores, atuadores e outros dispositivos que constroem a rede de comunicação [Daneels and Salter, 1999]. A detecção de anomalias em máquinas monitorados pelo SCADA pode ser realizada por métodos baseados em dados, uma vez que não é possível obter o modelo matemático do processo a partir dos registros disponibilizados pelo sistema supervisório [Zhang, 2014]. Como iremos discutir mais adiante, lidaremos com métodos de aprendizado de máquina para detecção de anomalias que, no contexto dos paradigmas de manutenção, enquadram-se no limite entre a manutenção preditiva e a manutenção proativa, uma vez que abordaremos diagnóstico e prognóstico de falhas.

Gao et al. [2014] discute interessantes questões de segurança emergentes envolvendo o SCADA, além de apresentar características gerais de uso desse sistema de monitoramento e supervisão, extraídas de manuais operacionais, das quais extraímos algumas:

- Interface de usuário: o SCADA suporta múltiplas interfaces de usuário, com a finalidade de exibir detalhadamente diagramas e textos informacionais para os usuários;
- Tratamento de alarmes: o SCADA realiza periodicamente checagens do sistema

monitorado e determina o estado atual do sistema. Se alguma checagem identifica algum desvio importante, de acordo com o nível de prioridade configurada, alarmes são disparados para informar os usuários do sistema;

- Arquivo de registro: as diferentes checagens realizadas periodicamente são armazenadas em um arquivo de registro. Este arquivo contêm registros sobre o estado do sistema, eventuais disparo de alarmes e informações sobre processos críticos. Geralmente, o usuário têm acesso a esse arquivo de registro em conjunto com o arquivo de leituras dos sensores, que compõe o conjunto de dados $\mathcal{D}$;
- Geração de relatórios: relatórios podem ser gerados utilizando Linguagem de Consulta Estruturada (SQL);
- Automação: em uma situação anômala, ações corretivas podem ser disparadas se adotada uma configuração baseada em eventos.

Em comparação com o Sistema de Monitoramento de Condições da Máquina (CMS), o SCADA é mais barato porque é fornecido de forma embarcada em muitos sistemas industriais, sem demandar novas aquisições. Enquanto o CMS processa principalmente sinais de vibração adquiridos com elevadas taxas de frequência, o SCADA coleta as leituras dos sensores sobre temperatura, pressão, velocidades de rotação, etc. em intervalos de até 10 minutos [Yang et al., 2014].

Usualmente, a leitura dos diferentes parâmetros monitorados identifica dois tipos de processos geradores de anomalias [Basseville et al., 1993]:

- Processos aditivos: compreende entradas desconhecidas durante o monitoramento, geralmente relacionadas às falhas em sensores (apresentando discrepância entre a leitura realizada e o valor real), ao mau funcionamento de atuadores, a vazamentos, entre outros problemas relacionados;
- Processos multiplicativos: abrange as mudanças (graduais ou abruptas) que ocorrem em algum parâmetro da máquina. Estes processos causam importantes mudanças na saída do processo, pois estão relacionados à deterioração de equipamentos, entupimentos, perda de potência, entre outras causas.

O SCADA é formado basicamente por quatro componentes, dos quais a Unidade de Terminal Remoto (RTU) é o mais importante. A RTU coleta automaticamente os

dados a partir de sua conexão aos sensores, controladores, medidores, registradores e equipamentos. Os dados relevantes são transmitidos para a estação mestre, onde uma interface amigável do sistema pode ser analisada pelo operador. Ao mesmo tempo, as instruções recebidas da estação mestre são retransmitidas pela RTU para os controladores do sistema [Thomas and McDonald, 2017; Arghira et al., 2011]. Muitas vezes, o Dispositivo Eletrônico Inteligente (IED) está substituindo a RTU, ao integrar recursos de comunicação externa, para sincronizar o monitoramento remoto do processo, a capacidade de manipular volume elevado de dados e exibição de informações adicionais do processo [Hor and Crossley, 2005]. A Figura 8 apresenta os componentes que integram o SCADA.

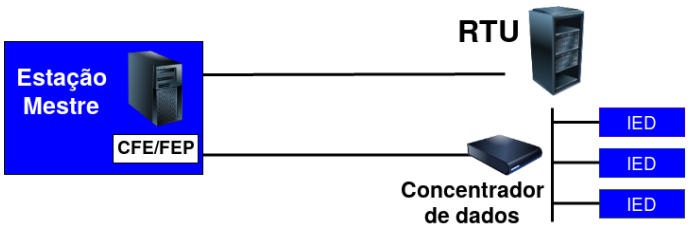

Figura 8 – Componentes básicos de um sistema SCADA.

Perspectivas futuras do SCADA indicam a sua transformação de um sistema monolítico e autossuficiente para um sistema *web* de acesso remoto que opera em tempo real com baixa latência, além de atender requisitos como robustez e segurança, uma demanda cada vez maior em aplicações de energia eólica *offshore* [Tavner, 2012; Gao et al., 2014]. Estes requerimentos são atendidos pela computação de borda (do inglês *edge computing*), um recém-estabelecido paradigma de computação distribuída que, em oposição à computação em nuvem, aproxima o processamento de aplicações das fontes de dados: sensores, dispositivos de IOT, etc. [Baker et al., 2020]. Assim, os dados gerados durante o monitoramento pelo SCADA são localmente processados por algum sistema especialista que poderá, ao término do processamento, enviar suas impressões para os recursos localizados na nuvem computacional [Sittón-Candanedo et al., 2019].

2.5- Sumário

Neste capítulo, apresentamos o histórico recente da energia eólica, cujo amadurecimento tecnológico alcançado a posicionou como uma promissora energia alternativa, com crescente participação na matriz energética em diferentes países. Abordamos, diante disso, os desafios que isso implica. Turbinas eólicas desempenham operações críticas que demandam uma consistente disciplina de O&M. Sistemas de detecção de anomalias integram esforços para a manutenção da operacionabilidade desses sistemas. Neste sentido, contextualizamos o SCADA na abordagem do problema discutido ao longo do trabalho.

3- Máquinas de Vetores de Suporte com rótulos difusos

Máquinas de vetores de suporte (SVM, do inglês *Support Vector Machine*) são uma técnica robusta fundamentada na teoria do aprendizado estatístico com uso extensivo em problemas de reconhecimento de padrões. Durante o processamento, os dados de treinamento são mapeados em um rótulo de classe $X \longmapsto Y$ enquanto um hiperplano é determinado pela maximização da distância das superfícies limítrofes das classes. Neste capítulo, discutimos uma abordagem das máquinas de vetores de suporte baseada na manipulação de conjunto de dados com rótulos difusos.

3.1- Introdução

O método máquinas de vetores de suporte com rótulos difusos estende os conceitos apresentados no Anexo A. Observamos que a classificação binária realizada pela abordagem convencional das máquinas de vetores de suporte lida com rótulos binários discretos $\{0, 1\}$ enquanto o método apresentado em [Thiel et al., 2007; Zhao et al., 2016] desenvolve as máquinas de vetores de suporte com rótulos contínuos no intervalo $[0, 1]$, que determinam um grau de pertinência às classes, conceito explorado pela lógica difusa.

No contexto da modelagem de problemas que lidam com conjunto de dados imperfeitos e informações imprecisas, o conceito de vagueza introduzido pela lógica difusa permite a construção de modelos de aprendizado de máquina baseados na semântica do problema [Williamson, 2002]. Ao admitir a ausência de rótulos nas amostras $\boldsymbol{x}_i$ para a realização do treinamento, o método máquinas de vetores de suporte com rótulos difusos situa-se no campo da classificação semi-supervisionada [Guo and Chen, 2009]. Neste cenário, dado que o processo de rotulagem de dados é dispendioso e até mesmo inviável, o domínio do conhecimento é crucial para o desenvolvimento de soluções.

Na abordagem proposta por Zhao et al. [2016], o conceito de vagueza é utilizado para determinar o grau de comprometimento da saúde da turbina eólica, que se presume piorar à medida que a observação se aproxima do registro da falha, de acordo com

o histórico fornecido pelo sistema SCADA. Dessa forma, o aprendizado consiste na identificação de anomalias que representam a evolução e o acúmulo de falhas. Em uma classificação binária, como proposto por Zhao et al. [2016], a identificação de amostras como positivas indica a ocorrência de variações importantes em relação ao padrão observado na série de dados. A Seção 3.2 apresenta o desenvolvimento desse algoritmo.

3.2- Classificação binária utilizando máquinas de vetores de suporte com rótulos difusos

O método máquinas de vetores de suporte com rótulos difusos foi baseado no método *fuzzy-input fuzzy-output* (ou simplesmente F^2-SVM), desenvolvido por Thiel et al. [2007]. Como o nome do método sugere, ele realiza o treinamento utilizando dados de entrada com rótulos difusos $y = [0, 1]$ e classifica estes dados também usando rótulos difusos $\hat{y} \in [0, 1]$. A abordagem apresentada por Zhao et al. [2016], contudo, discretiza os rótulos resultantes da classificação $\hat{y}$. Essa transformação foi realizada pela função sinal a fim de produzir $\hat{y} \in \{-1, 0, 1\}$. Nós trataremos do desenvolvimento do método F^2-SVM, mas seguiremos o desenvolvimento proposto por Zhao et al. [2016] para construir a decisão final da classificação.

Consideramos o conjunto de treinamento sob a forma:

$$(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_l, y_l),\ \boldsymbol{x} \in \Re^n,\ y \in \{-1, +1\}$$

onde n é a dimensão do conjunto de dados.

O desenvolvimento teórico do método F^2-SVM respeita o que foi discutido no Anexo A.3 em relação ao SVM com margens suaves, onde também é apresentada a formulação primal do problema de otimização.

A inclusão de novas variáveis permite estender aquela formulação para lidar com os rótulos difusos. Sejam u_i^+ e u_i^- os graus de pertinência de cada amostra $\boldsymbol{x}_i$ às classes positiva e negativa, respectivamente, onde $u_i^+ + u_i^- = 1$. Estes graus de pertinência são ponderados pelas variáveis de folga ξ_i^+ e ξ_i^-. A Equação (1) apresenta a formulação primal do problema de otimização do método F^2-SVM.

$$\text{Minimizar} \quad \frac{1}{2}||\boldsymbol{w}||^2 + C\left(\sum_{i=1}^{l}(\xi_i^+ u_i^+ + \xi_i^- u_i^-)\right)$$

sujeito a:

$$\begin{aligned} &\boldsymbol{w}^T\phi(\boldsymbol{x}_i) + b \geq 1 - \xi_i^+, \ i = 1, \ldots, l \\ &-\boldsymbol{w}^T\phi(\boldsymbol{x}_i) - b \geq 1 - \xi_i^-, \ i = 1, \ldots, l \\ &\xi_i^+, \xi_i^- \geq 0, \ i = 1, \ldots, l \end{aligned} \tag{1}$$

onde $\phi(\cdot)$ é a função kernel utilizada, $\boldsymbol{w}$ e b são parâmetros do modelo, e a constante C é o custo de uso das variáveis de folga ξ_i^- e ξ_i^+.

A Figura 9 apresenta a interação das variáveis de folga ξ_i^- e ξ_i^+ com as margens do hiperplano. Observamos que, para qualquer erro de classificação, $\xi_i^-, \xi_i^+ \neq 0$.

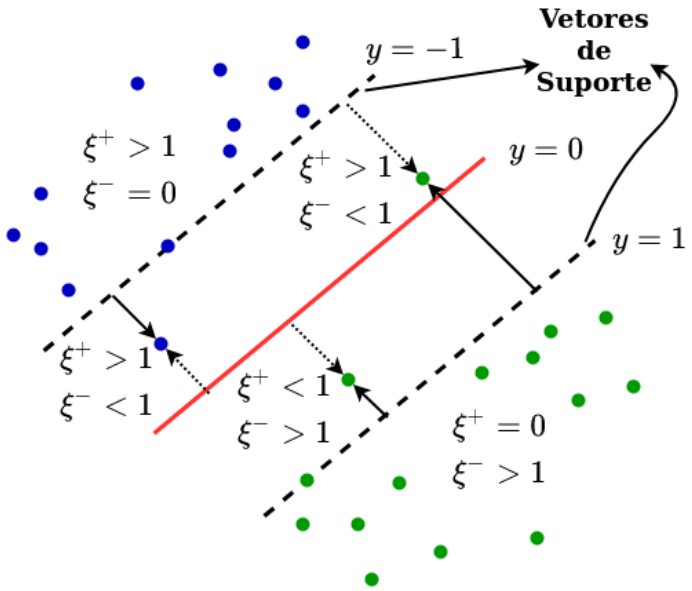

Figura 9 – Determinação do hiperplano ótimo no processamento das máquinas de vetores de suporte com rótulos difusos, construído baseado no relaxamento das variáveis de folga ξ_i^- e ξ_i^+.

Como apresentado para o caso do algoritmo convencional das máquinas de vetores de suporte no Anexo A.3, vamos derivar a formulação dual a fim de resolver a programação quadrática [Thiel et al., 2007]. O resultado da aplicação da função de Lagrange na formulação primal é apresentado na Equação (2).

$$\mathcal{L}(\boldsymbol{w}, b, \xi_i^+, \xi_i^-, \alpha_+, \alpha_-, \beta^+, \beta^-) = \frac{1}{2}||\boldsymbol{w}||^2 + C\left(\sum_{i=1}^{l}(\xi_i^+ u_i^+ + \xi_i^- u_i^-)\right)$$
$$- \sum_{i=1}^{l}\alpha_i^+((\boldsymbol{w}^T\boldsymbol{x}_i + b) - (1 - \xi_i^+)) + \sum_{i=1}^{l}\alpha_i^-((\boldsymbol{w}^T\boldsymbol{x}_i + b) - (1 - \xi_i^-)) \tag{2}$$
$$- \sum_{i=1}^{l}\beta_i^+\xi_i^+ - \sum_{i=1}^{l}\beta_i^-\xi_i^-$$

onde $\alpha-, \alpha_+, \beta^+$ e β^- são multiplicadores de Lagrange.

Diferenciando a função de Lagrange $\mathcal{L}(\boldsymbol{w}, b, \xi_i^+, \xi_i^-, \alpha_+, \alpha_-, \beta^+, \beta^-)$ em relação às variáveis que devem ser minimizadas $\boldsymbol{w}, b, \xi_i^+$, obtemos as condições necessárias para construir a solução:

$$\frac{\partial\mathcal{L}}{\partial\boldsymbol{w}} = \boldsymbol{w} - \sum_{i=1}^{l}\alpha_i^+\boldsymbol{x}_i + \sum_{i=1}^{l}\alpha_i^-\boldsymbol{x}_i = 0 \Rightarrow \sum_{i=1}^{l}(\alpha_i^+ - \alpha_i^-)\boldsymbol{x}_i$$
$$\frac{\partial\mathcal{L}}{\partial b} = \boldsymbol{w} - \sum_{i=1}^{l}\alpha_i^+ + \sum_{i=1}^{l}\alpha_i^- = 0 \Rightarrow \sum_{i=1}^{l}(\alpha_i^+ - \alpha_i^-) \tag{3}$$
$$\frac{\partial\mathcal{L}}{\partial\xi_i^+} = Cu_i^+ - \alpha_i^+ - \beta_i^+ = 0, \quad \frac{\partial\mathcal{L}}{\partial\xi_i^-} = Cu_i^- - \alpha_i^- - \beta_i^- = 0$$

Substituindo as parcelas da Equação (3) na Equação (2) e posteriormente reordenando, obtemos a formulação dual do problema de otimização, de acordo com a Equação (4).

$$\text{Maximizar} \quad -\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}(\alpha_i^+ - \alpha_i^-)(\alpha_j^+ - \alpha_j^-)K(\boldsymbol{x}_i, \boldsymbol{x}_j) + \sum_{i=1}^{l}\alpha_i^+ + \sum_{i=1}^{l}\alpha_i^-$$

sujeito a:

$$\sum_{i=1}^{l}(\alpha_i^+ + \alpha_i^-) = 0 \tag{4}$$
$$0 \leq \alpha_i^+ \leq Cu_i^+,\ 0 \leq \alpha_i^- \leq Cu_i^-,\ i = 1, \ldots, l$$

Das diferenciais na Equação (3), obtemos $\boldsymbol{w} = \sum_{i=1}^{l}(\alpha_i^+ - \alpha_i^-)x_i$, com $\alpha_i^+ = Cu_i^+$ e $\alpha_i^- = Cu_i^-$ determinando os vetores de suporte. De acordo com Zhao et al. [2016], a classificação das amostra é obtida pela Equação (5).

$$\hat{y}(\boldsymbol{x}) = sign\left(\sum_{i=1}^{l}(\alpha_i^+ - \alpha_i^-)K(\boldsymbol{x}_i, \boldsymbol{x}) + b\right) \tag{5}$$

onde $sign(\cdot)$ é a função sinal.

Seja $z(\boldsymbol{x}) = \left(\sum_{i=1}^{l}(\alpha_i^+ - \alpha_i^-)K(\boldsymbol{x}_i, \boldsymbol{x}) + b\right)$. A saída fornecida pela função sinal $sign(\cdot)$ respeita a condição apresentada na Equação (6).

$$sign(z(\boldsymbol{x})) := \begin{cases} -1, & \text{se } z(\boldsymbol{x}) < 0 \\ 0, & \text{se } z(\boldsymbol{x}) = 0 \\ +1, & \text{se } z(\boldsymbol{x}) > 1 \end{cases} \tag{6}$$

A Equação (6) atribui rótulos do tipo *crisp* aos dados de entrada originalmente difusos. Este resultado será fundamental para a solução do problema de detecção de anomalias, cujas amostras de entrada $\boldsymbol{x}_i$ associam-se aos rótulos $u_i^+ + u_i^-$ que determinam seu grau de pertencimento ao processo de pré-falha (crescente em função do tempo). O objetivo do processamento dessa classificação semi-supervisionada pelas máquinas de vetores de suporte com rótulos difusos será, portanto, identificar quais amostras, de fato, podem ser assim classificadas.

3.3- Sumário

Neste capítulo, abordamos o método central em um dos fluxos de trabalho presentes em nossas contribuições. As máquinas de vetores de suporte com rótulos difusos foram fundamentadas como uma extensão da abordagem convencional das máquinas de vetores de suporte. Dessa forma, pudemos compreender a sua participação no desenvolvimento da solução do problema deste trabalho.

4- Modelos Ocultos de Markov

Os modelos ocultos de Markov estendem os conceitos da Cadeia de Markov ao assumirem que cada observação é uma função de probabilidade do estado, o que resulta num modelo composto por um processo estocástico de dupla camada, na qual um processo estocástico não-observável (oculto) pode ser observado somente por outro processo estocástico, este último responsável por gerar a sequência de observações [Baum and Petrie, 1966]. Assim, a cadeia de Markov governa a transição entre estes estados, que em última instância determina qual a distribuição de probabilidade (segunda camada do processo estocástico) gera cada observação de uma série temporal. Neste capítulo, discutiremos os aspectos teóricos e práticos dos modelos ocultos de Markov.

4.1- Introdução

De acordo com Rabiner and Juang [1986], séries temporais são definidas como um conjunto de observações $\boldsymbol{O} = (O_1, O_2, \ldots, O_T)$, cada uma gravada no instante t . A trajetória da série temporal permite extrair diferentes propriedades do sistema monitorado. Para tanto, é necessária a aplicação de métodos clássicos de inferência, objetivando a estimação dos parâmetros do processo gerador da série temporal, para posteriormente construir o modelo estatístico que sistematiza o seu comportamento [Wasserman, 2013].

Entretanto, muitas vezes a série temporal demonstra possuir mecanismos mais complexos responsáveis pela sua furtividade ao padrão esperado. Nestes casos, podemos adotar os modelos ocultos de Markov (HMM, do inglês *Hidden Markov Models*) para modelar a série temporal, assumindo a ação de processos estocásticos que interagem sob a influência da cadeia de Markov [Gikhman and Skorokhod]. Este processamento é decomposto em dois estágios: (1) um processo estocástico estacionário não observável que abrange um conjunto finito de estados $\boldsymbol{S} = \{S_1, S_2, \ldots, S_N\}, i = 1, 2, \ldots, N$, cuja dinâmica de transição entre tais estados é explicada pela cadeia de Markov $\boldsymbol{A} = \{a_{ij}\}$, onde $a_{ij}(t) = P(q_{t+1} = S_j | q_t = S_i), 1 \leq i \leq N,\ 1 \leq j \leq N$ é a probabilidade de transição

do estado i para o estado j no instante $t+1$; (2) uma sequência de observações $\boldsymbol{O}$ gerada por distribuições multidimensionais determinadas somente pelo estado atual i.e. $\boldsymbol{B} = \{b_j(k)\}$ tal que $b_j(k) = P(O_t = v_k | q_t = S_j)$, onde b_j é a probabilidade da observação O_t no estado S_j, que emite um símbolo $v_k \in \boldsymbol{V}$ do vocabulário $\boldsymbol{V} = \{v_1, v_2, \ldots, v_M\}$ formado por observações distintas [Poritz, 1988].

Os modelos ocultos de Markov são expressos compactamente pela notação $\lambda = (\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\Pi})$, onde $\boldsymbol{A}$ é a matriz de transições da cadeia de Markov, $\boldsymbol{B}$ é a probabilidade de emissão dos elementos da sequência de observações $\boldsymbol{O}$ e Π é a probabilidade inicial do conjunto de estados $\boldsymbol{S}$ [Rabiner and Juang, 1986].

Sob a perspectiva do modelo probabilístico gráfico, os modelos ocultos de Markov são vistos como uma subclasse das redes Bayesianas dinâmicas, um tipo especial de rede Bayesiana que inclui a modelagem temporal nas relações de dependência condicional entre as variáveis latentes e observáveis organizadas na estrutura de um grafo. Conceitualmente, o modelo probabilístico gráfico representa diagramaticamente uma distribuição de probabilidade [Bishop, 2006; Koller and Friedman, 2009]. Neste grafo $\mathcal{G} = (\boldsymbol{\mathcal{V}}, \boldsymbol{\mathcal{E}})$, onde $\boldsymbol{\mathcal{V}}$ é o conjunto de todos os nós e $\boldsymbol{\mathcal{E}}$ é o conjunto de todas as arestas, temos as variáveis latentes e observáveis alocadas como nós, cujas arestas determinam relações de dependência condicional. Em contraste, a ausência de arestas conectando essas variáveis representa a independência condicional entre essas variáveis. Podemos interpretar o grafo resultante dessas conexões como um modelo que expressa os processos causais que geraram o conjunto de observações [Koller and Friedman, 2009]. Os modelos assim construídos são denominados geracionais, pois têm a capacidade de gerar observações sintéticas cuja distribuição de probabilidade seria a mesma de observações reais [Koller and Friedman, 2009]. A Figura 10 apresenta a arquitetura dos modelos ocultos de Markov como um grafo direcionado, no qual destacamos a atuação do processo estocástico instantâneo dos estados $q_t = S_i$ do modelo, que governa ocultamente a distribuição multidimensional $\boldsymbol{B}$, a fim de gerar a emissão no intervalo de tempo observado.

A decomposição das relações de dependência e independência condicional em um grafo é o conceito-chave na formulação das propriedades da fatoração de grafos [Koller and Friedman, 2009]. A Equação (7) apresenta a fatoração do grafo da Figura 10 [Ghahramani, 2001].

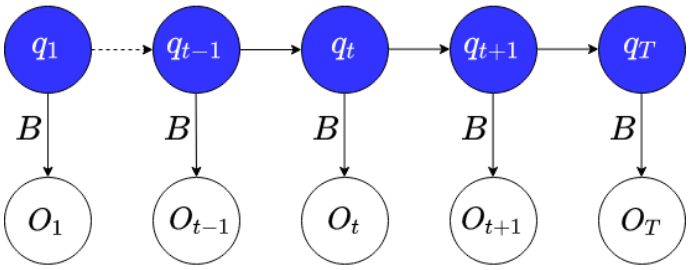

Figura 10 – As emissões O_t do modelos observadas no intervalo $1 \leq t \leq T$ são geradas por um distribuição $b_k = P(O_t = v_k | q_t = S_i)$ em respeito ao estado S_i atuante naquele instante. Adaptado de [Jordan, 2003].

$$P(\boldsymbol{S}, \boldsymbol{O}) = P(\Pi_1 | \boldsymbol{\Pi}) \left[\prod_{t=2}^{T} P(q_{t+1} = S_j | q_t = S_i) \right] \prod_{t=2}^{T} P(O_t = v_k | q_t = S_i) \tag{7}$$

onde $P(\Pi_1 | \boldsymbol{\Pi}) = \prod_{i=1}^{N} \Pi_i^{\Pi_{1i}}$ é a probabilidade marginal do estado inicial, com $\sum_i \Pi_i = 1$.

A Figura 11 apresenta o diagrama de transições da cadeia de Markov. Neste exemplo, a cadeia de Markov define a probabilidade de transição de estado $a_{ij} = P(q_{t+1} = S_j | q_t = S_i), 1 \leq i \leq N,\ 1 \leq j \leq N$ em um conjunto $\boldsymbol{S}$ que possui $N = 4$ estados (cada um sendo uma variável aleatória).

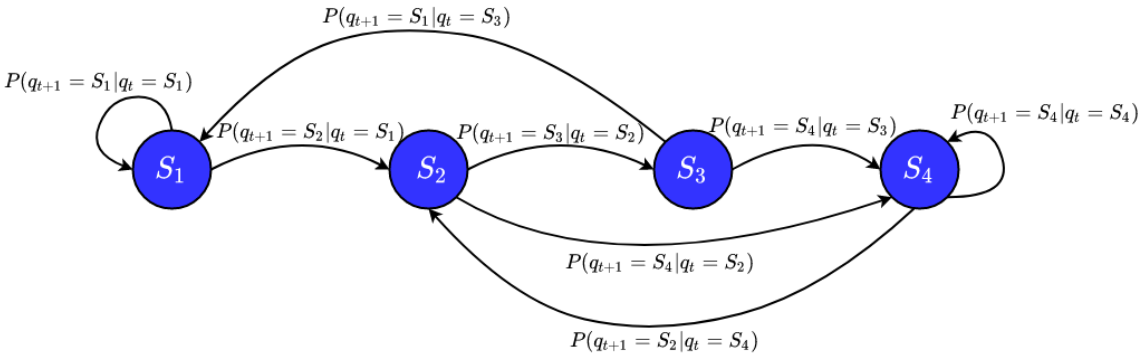

Figura 11 – Diagrama de transição entre os estados de uma cadeia de Markov, onde $a_{ij} = P(q_{t+1} = S_j | q_t = S_i), 1 \leq i \leq N,\ 1 \leq j \leq N$ determina o valor da probabilidade de transição do estado i para o estado j em um instante $t + 1$.

A probabilidade condicional da matriz de transição é definida explicitamente pela Equação (8) [Rabiner and Juang, 1986].

$$P(q_{t+1} = S_j | q_t = S_i) = \prod_{j=1}^{N} \prod_{i=1}^{N} \boldsymbol{A}_{ij}^{q_t = S_i, q_{t+1} = S_j} \tag{8}$$

Como observamos, a cadeia de Markov acoplada ao HMM desempenha um

papel crucial na dinâmica das emissões ao determinar as condições de transição entre os estados, que compõem o mecanismo oculto do método. Um aprofundamento teórico sobre as cadeias de Markov foi apresentado no Anexo B. Prosseguimos para os fundamentos dos modelos ocultos de Markov, desenvolvidos na Seção 4.2

4.2- Fundamentos dos Modelos Ocultos de Markov

Neste trabalho, assumimos que os modelos ocultos de Markov envolvem processos estocásticos de tempo discreto que operam também no espaço de estados discreto. A Figura 12 exemplifica o processamento dos modelos ocultos de Markov. Note que cada estado possui uma probabilidade de atuação cuja dinâmica de transição a_{ij} é determinada pelas probabilidades contidas na matriz de transição homogênea $\boldsymbol{A}$ da cadeia de Markov. Assumimos que cada estado está associado a uma distribuição gaussiana $\mathcal{N}(\mu, \sigma)$, onde μ é a média da distribuição e σ é o desvio padrão. No extremo da cadeia do processamento do método, temos as observações $\boldsymbol{O}$, geradas a partir da distribuição de probabilidade ativa.

Ao analisarmos a Figura 12, podemos extrair diferentes problemas que os modelos ocultos de Markov devem resolver [Rabiner and Juang, 1986]:

- Avaliação (*scoring*): este problema avalia a máxima verossimilhança de uma sequência de observações $P(\boldsymbol{O}|\lambda)$ em relação ao modelo $\lambda = (\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi})$, o que permite determinar se essa sequência foi gerada ou não por este modelo;
- Decodificação: este problema determina, a partir de uma sequência de observações $\boldsymbol{O}$, a sequência de estados ocultos $\boldsymbol{\chi} \subset \boldsymbol{S}$ mais provável que a gerou;
- Aprendizado: este problema está relacionado ao treinamento do modelo a partir de uma sequência de observações, a fim de ajustar os parâmetros do modelo $\lambda = (\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi})$ pela maximização da probabilidade *a posteriori* $P(\boldsymbol{O}|\lambda)$.

Os problemas canônicos dos modelos ocultos de Markov demandam a aplicação de técnicas de inferência em redes Bayesianas dinâmicas, que envolvem a execução de diferentes tarefas, como discutem Koller and Friedman [2009]. Mais adiante, exploraremos

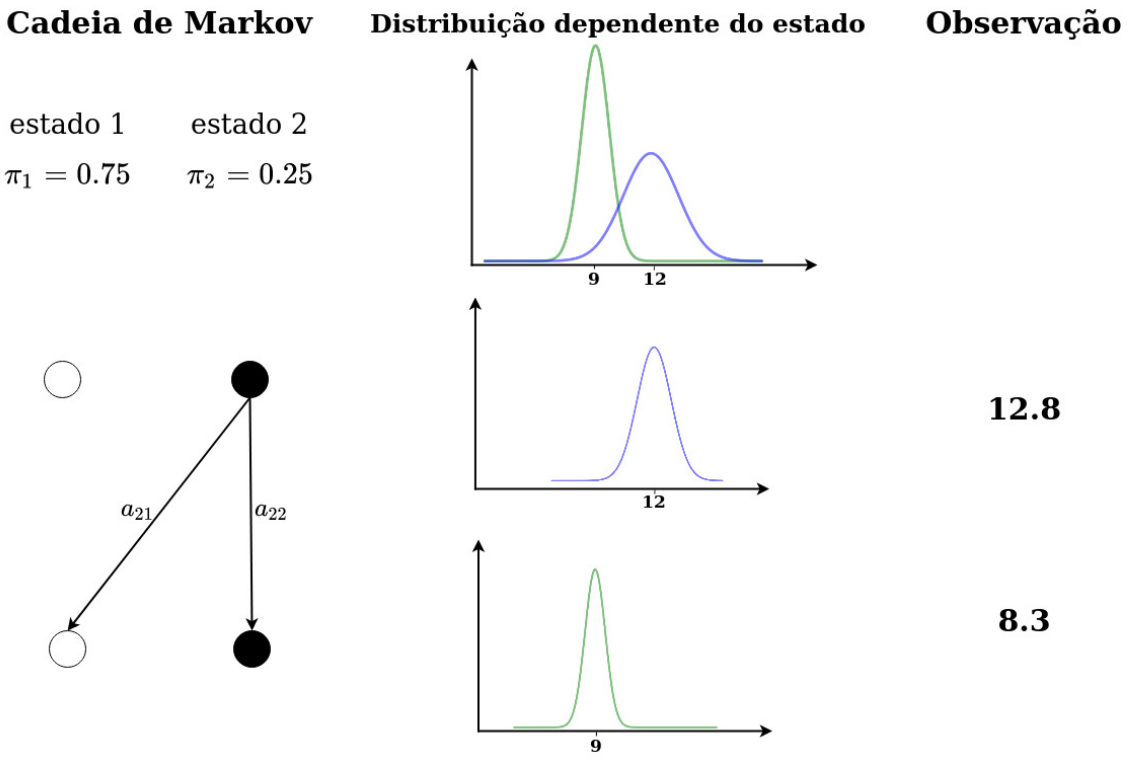

Figura 12 – Representação do processamento dos modelos ocultos de Markov. Adaptado de [Zucchini et al., 2017].

este conceito utilizando o protocolo de passagem de mensagens em um grafo de fatores derivado da representação gráfica original dos modelos ocultos de Markov. Essas tarefas de inferência baseiam-se na propagação de crenças com a finalidade de calcular a probabilidade marginal em relação a cada nó do grafo:

- Filtragem (ou rastreamento): A primeira tarefa da inferência em modelos gráficos é o rastreamento, que se encarrega, em qualquer instante t, de computar as crenças mais embasadas sobre o estado atual do sistema $q_t = S_i$ a partir de todas as evidências obtidas. Ou seja, o rastreamento deseja consolidar a crença da permanência do estado atual do sistema. Formalmente, temos que a crença neste estado no instante t é denotado por $\sigma^{(t)}(q_t = S_i) = P(q_t = S_i|\boldsymbol{O}^{(1:t)})$.

- Predição: Outra tarefa desempenhada durante a inferência é a predição, que, a partir do intervalo de observações $\boldsymbol{O}^{(1:t)}$, prediz a distribuição para instantes $t' > t$.

- Suavização: Uma terceira tarefa é a suavização, que computa a probabilidade *a posteriori* de $q_t = S_i$ a partir de todas as evidências fornecidas pelas observações $\boldsymbol{O}^{(1:t)}$, sob a forma $P(q_t = S_i|\boldsymbol{O}^{(0:u)})$, onde $t < u$. De acordo com Koller and

Friedman [2009], durante o rastreamento, as evidências acumulam-se gradualmente, o que assegura a mudança de estados do sistema de forma segura.

- Decodificação: Por fim, é executada a tarefa que se ocupa de identificar a mais provável trajetória do sistema a partir das evidências obtidas com as observações, definido por arg max $P(\boldsymbol{\chi}|\boldsymbol{O})$, onde $\boldsymbol{\chi} = \{q_1 = S_1, q_2 = S_i, \ldots, q_T\}$ é uma sequência de estados arbitrária de tamanho fixo igual a T. Esta última tarefa está diretamente relacionada ao problema canônico de decodificação da sequência de estados.

As tarefas de inferência exata discutidas acima nos guiarão na determinação das solução dos problemas canônicos dos modelos ocultos de Markov sob a perspectiva do modelo gráfico. A Seção 4.2.1 discute o problema de avaliação.

4.2.1 Problema de avaliação (ou *scoring*)

Para deduzirmos $P(\boldsymbol{O}|\lambda)$, o que soluciona o problema de avaliação, vamos considerar o cálculo da probabilidade condicional $P(\boldsymbol{\chi}|\boldsymbol{O}, \lambda)$ apresentada anteriormente [Jordan, 2003]. Pela definição do teorema de Bayes, temos:

$$P(\boldsymbol{\chi}|\boldsymbol{O}, \lambda) = \frac{P(\boldsymbol{\chi}, \boldsymbol{O}, \lambda)}{P(\boldsymbol{O}|\lambda)} \tag{9}$$

Definimos a probabilidade $P(\boldsymbol{O}|\lambda)$ a partir da Equação (7), mas notamos que ela não realiza a fatoração em relação a todos os N estados do conjunto $\boldsymbol{S}$ [Bishop, 2006]. Introduzindo o somatório em relação a todos os estados, obtemos a função de verossimilhança (10).

$$P(\boldsymbol{O}|\lambda) = \sum_{\boldsymbol{S}} P(\boldsymbol{O}, \boldsymbol{S}|\lambda) \tag{10}$$

Antes de prosseguirmos com o desenvolvimento da probabilidade *a posteriori* $P(\boldsymbol{\chi}|\boldsymbol{O}, \lambda)$, vamos ponderar sobre a complexidade computacional envolvida em seu cálculo. A fatoração apresentada na Equação (10) contempla o somatório de N fatorações envolvendo T variáveis, resultando em N^T termos. Note que a treliça apresentada na Figura 13 contempla o processamento da inferência sobre uma rede Bayesiana dinâmica

para a obtenção da probabilidade *a posteriori* $P(\boldsymbol{\chi}|\boldsymbol{O}, \lambda)$. Observe que as setas na cor vermelha indicam o percurso que fornece o máximo *a posteriori* (MAP, do inglês *maximum a posteriori*), denotado por arg max $P(\boldsymbol{\chi}^{(1:7)}|\boldsymbol{O}^{(1:7)}, \lambda)$.

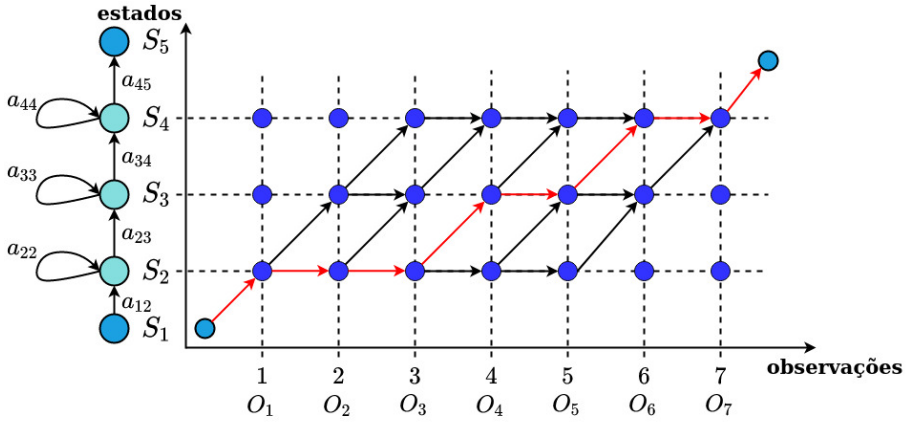

Figura 13 – Treliça de observações e estados dos modelos ocultos de Markov.

Concluímos que a inferência em redes Bayesianas dinâmicas enfrenta desafios relacionados ao tamanho da rede e ao processamento da dimensão temporal. Para lidar com isso, a decodificação da propagação da crença nos estados é obtida a partir de uma modificação do algoritmo soma-produto. Essa modificação originou o algoritmo *forward-backward*, caracterizado pela adoção da programação dinâmica para executar a passagem de mensagens pelos ramos do grafo de forma eficiente [Koller and Friedman, 2009].

A Figura 14 oferece a perspectiva do processamento da probabilidade *a posteriori* dos estados em cada instante t, ao invés de adotar toda a sequência [Jordan, 2003]. Utilizaremos essa perspectiva para desenvolvermos o algoritmo *forward-backward*.

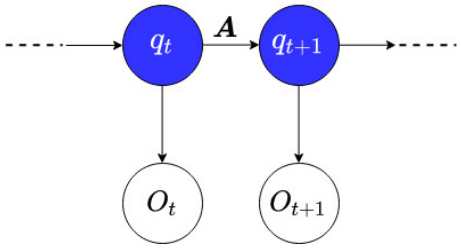

Figura 14 – Representação dos modelos de Markov como um modelo gráfico. Cada vértice representa um passo temporal. Os nós superiores representam a variável latente de distribuição multidimensional q_t enquanto os nós inferiores representam as variáveis de observação O_t. Adaptado de [Jordan, 2003].

A Equação (11) apresenta o cálculo da probabilidade *a posteriori* a exemplo da

Equação (9) utilizando o teorema de Bayes, mas sob a perspectiva apresentada pela Figura 14 [Jordan, 2003].

$$\begin{aligned} P(q_t = S_i|\boldsymbol{O}, \lambda) &= \frac{P(\boldsymbol{O}|q_t = S_i, \lambda)P(q_t = S_i|\lambda)}{P(\boldsymbol{O}|\lambda)} \\ &= \frac{P(O_1, \ldots, O_t|q_t = S_i, \lambda)P(O_{t+1}, \ldots, O_T|q_t = S_i, \lambda)P(q_t = S_i|\lambda)}{P(\boldsymbol{O}|\lambda)} \end{aligned} \tag{11}$$

Pelo reagrupamento dos termos, obtemos:

$$P(q_t = S_i|\boldsymbol{O}, \lambda) = \frac{P(O_1, \ldots, O_t, q_t = S_i, \lambda)P(O_{t+1}, \ldots, O_T|q_t = S_i, \lambda)}{P(\boldsymbol{O}|\lambda)} \tag{12}$$

Extraímos dois termos da Equação (12), $\mathfrak{F}(q_t = S_i)$ e $\mathfrak{B}(q_t = S_i)$. O termo $\mathfrak{F}(q_t = S_i) \triangleq P(O_1, \ldots, O_t, q_t = S_i)$ corresponde à etapa *forward* do algoritmo, que calcula a probabilidade de emissão da sequência parcial $O_1, \ldots, O_t$. Completando a sequência, o termo $\mathfrak{B}(q_t = S_i) = P(O_{t+1}, \ldots, O_T|q_t = S_i)$ corresponde à etapa *backward* do algoritmo, que calcula a probabilidade de emissão da sequência parcial $O_{t+1}, \ldots, O_T$.

Dado que o somatório de $P(q_t = S_i|\boldsymbol{O})$ sobre os possíveis valores de q_t é igual a 1, temos:

$$P(\boldsymbol{O}|\lambda) = \sum_{q_t} \mathfrak{F}(q_t)\mathfrak{B}(q_t) \tag{13}$$

Denotando a probabilidade *a posteriori* instantânea $P(q_t = S_i|\boldsymbol{O})$ como $\gamma(q_t)$, obtemos de acordo com [Rabiner and Juang, 1986; Ghojogh et al., 2019]:

$$\gamma(q_t) \triangleq \frac{\mathfrak{F}(q_t)\mathfrak{B}(q_t)}{P(\boldsymbol{O}|\lambda)} \tag{14}$$

onde $P(\boldsymbol{O}|\lambda)$, de acordo com a Equação (13), é o fator de normalização da probabilidade *a posteriori* em cada instante t.

A Equação (13) fornece a solução do problema canônico da avaliação de uma sequência (ou *scoring*), viabilizada pelo uso do algoritmo *forward-backward*, sobre o qual nos debruçaremos para apresentar o seu desenvolvimento [Rabiner and Juang, 1986; Jordan, 2003]. Tipicamente, o desenvolvimento é apresentado em duas partes, iniciando pelo algoritmo *forward* $\mathfrak{F}(q_t)$, de acordo com a Equação (15).

$$
\begin{aligned}
\mathfrak{F}(q_{t+1} = S_j) &= P(O_0, \ldots, O_t, q_t = S_j) \\
&\sum_{q_t} P(O_0, \ldots, O_t) P(q_{t+1} = S_j | q_t = S_i) P(O_{t+1} | q_{t+1}) \\
&\sum_{q_t} \mathfrak{F}(q_t = S_i) P(q_{t+1} = S_j | q_t = S_i) P(O_{t+1} | q_{t+1}) \\
&\sum_{q_t} \mathfrak{F}(q_t = S_i) a_{i,j} b_{ij}(O_{t+1}) \qquad (15)
\end{aligned}
$$

onde $\mathfrak{F}(q_t = S_1) = P(O_1 | q_1 = S_1)\Pi_1 = \prod_{i=1}^{N} \{\Pi_i p(O_1 | q_1 = S_1)$ é o valor inicial.

Podemos notar na Equação (15) que o algoritmo adota o mecanismo de recursão no escopo da programação dinâmica para ganhar desempenho. Com isso, a complexidade computacional do algoritmo *forward* é $O(N^2 T)$, considerando os N estados e T variáveis de $t = 1, \ldots, T$. O Algoritmo 1 generaliza o uso da Equação (15) [Ghojogh et al., 2019].

Algoritmo 1 – *Forward*

Entrada: $\lambda = (\mathbf{\Pi}, \boldsymbol{A}, \mathbf{B})$

// Inicialização

F($q_1 = S_i$) $= \Pi_i b_i(O_1), \quad \forall i \in \{1, \ldots, N\}$

// Indução

for *estado j de 1 até N* **do**

  for *tempo t de 0 até T* **do**

    F($q_{t+1} = S_j$) $= \left[\sum_{i=1}^{N} \mathfrak{F}(q_t = S_i) a_{i,j}\right] b_j(O_{t+1})$

P($\boldsymbol{O}|\lambda$) $= \sum_{i=1}^{N} \mathfrak{F}(q_T = S_i)$

// Finalização

Saída: $P(\boldsymbol{O}|\lambda), \forall i, t : \mathfrak{F}(q_t = S_i)$

A Equação (16) apresenta o desenvolvimento do algoritmo *backward* $\mathfrak{B}(q_t)$, de acordo com [Rabiner and Juang, 1986; Jordan, 2003].

$$
\mathfrak{B}(q_t = S_i) = P(O_T, \ldots, O_{t+1} | q_t = S_i)
$$

$$
\begin{aligned}
&= \sum_{q_{t+1}} P(O_{t+1}, \ldots, O_T | q_{t+1} = S_j | q_t = S_i) \\
&= \sum_{q_{t+1}} P(O_{t+1} | q_{t+1}) P(q_{t+1} = S_j | q_t = S_i) P(O_{t+2}, \ldots, O_t | q_{t+1}) \\
&= \sum_{q_{t+1}=S_j} \mathfrak{B}(q_{t+1} = S_j) a_{i,j} b_j(O_{t+1}) \qquad (16)
\end{aligned}
$$

O Algoritmo 2 apresenta a generalização da Equação (16) [Ghojogh et al., 2019].

Algoritmo 2 – *Backward*

Entrada: $\lambda = (\mathbf{\Pi}, \boldsymbol{A}, \mathbf{B})$

// Inicialização

$\beta(q_t = S_i) = 1, \ \ \forall i \in \{1, \ldots, N\}$

// Indução

for *estado j de 1 até N* **do**

  for *tempo t de (T-1) até 1* **do**

    B($q_t = S_i$) $= \sum_{j=1}^{N} a_{i,j} b_j(O_{t+1})$

Saída: $\forall i, \forall t : \mathfrak{B}(q_t = S_i)$

De acordo com Bishop [2006], durante a solução de problemas de inferência em modelos gráficos é conveniente converter a representação gráfica original do problema para a representação gráfica de fatores, que deve ser capaz de capturar a estrutura do grafo original. Um grafo fator é um grafo bipartido que expressa a fatoração de uma função global em muitas funções locais. Seja $f(S_1, \ldots, S_N)$ uma função que permite a sua decomposição em K fatores, de acordo com a Equação (17).

$$
f(S_1, \ldots, S_n) = \prod_{k=1}^{K} f_k(\mathbf{\Theta}_K), \qquad (17)
$$

onde $\mathbf{\Theta}_k \subset \{S_1, \ldots, S_n\}$ é o subconjunto de variáveis associadas com o fator f_k em seu espaço de configuração.

Como vemos na Figura 10, os modelos ocultos de Markov são representados originalmente como um grafo direcionado. Durante a conversão de um gráfico direcionado para um grafo fator, removemos algumas arestas direcionadas e acrescentamos os fatores denominados potencial $\psi_{t,t+1}(q_t, q_{t+1})$ e produto $\varphi(q_t)$ como substitutos, a fim de simplificar o processamento da inferência [Bishop, 2006; Koller and Friedman, 2009]. A

Figura 15 apresenta o processo de remoção das arestas direcionais e a substituição dos nós das variáveis observáveis pelos fatores.

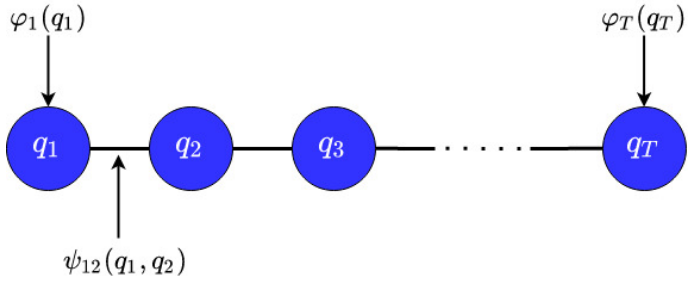

Figura 15 – Conversão do grafo direcionado dos modelos ocultos de Markov para a sua representação como um grafo de fatores. Adaptado de [Ghojogh et al., 2019]

Em extensão à representação diagramática dos modelos de Markov apresentada na Figura 15, vemos na Figura 16 o enfoque em um setor da treliça da Figura 13 que permite visualizar a atuação dos fatores no processamento da inferência em um grafo bipartido.

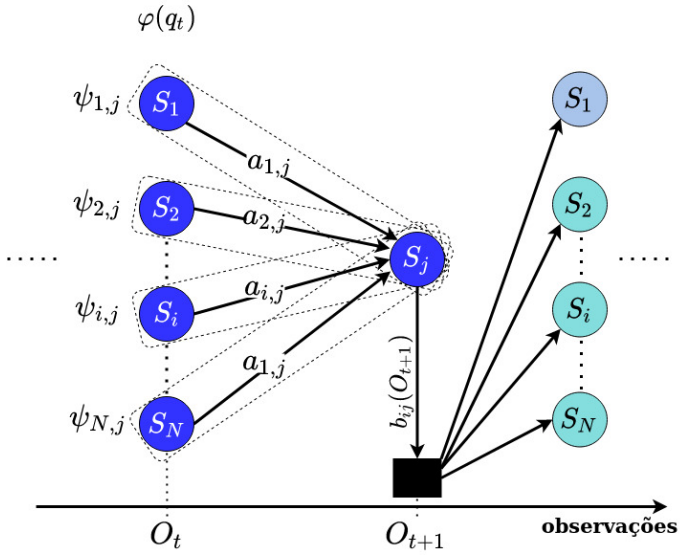

Figura 16 – Enfoque em um setor da treliça destacando a atuação dos fatores $\psi_{t,t+1}(q_t, q_{t+1})$ e $\varphi(q_t)$ para modelar a propagação da crença do algoritmo *forward*. Adaptado de [Ghojogh et al., 2019].

Definimos estes fatores como [Bishop, 2006; Koller and Friedman, 2009]:

$$
\begin{aligned}
&\varphi_1(q_1) = P(q_1, O_1) \\
&\varphi_t(q_t) = P(O_t|q_t),\ \forall_t \in \{2, \ldots, T\} \\
&\psi_{t,t+1}(q_t, q_{t+1}) = P(q_{t+1}|q_t),\ \forall_t \in \{1, \ldots, T-1\}
\end{aligned} \tag{18}
$$

O algoritmo *forward-backward* é derivado do algoritmo soma-produto, um algoritmo de inferência exata que estabelece o protocolo de passagem de mensagens (ou propagação de crenças probabilísticas) em grafos, cuja finalidade é o cálculo das probabilidades marginais em relação a cada nó. No contexto do algoritmo *forward-backward*, as mensagens $m_{t\to(t+1)}$ realizam o processamento da etapa *forward* enquanto as mensagens $m_{t\to(t-1)}$ realizam o processamento *backward*. Podemos visualizar este mecanismo de passagem de mensagens no diagrama da Figura 17.

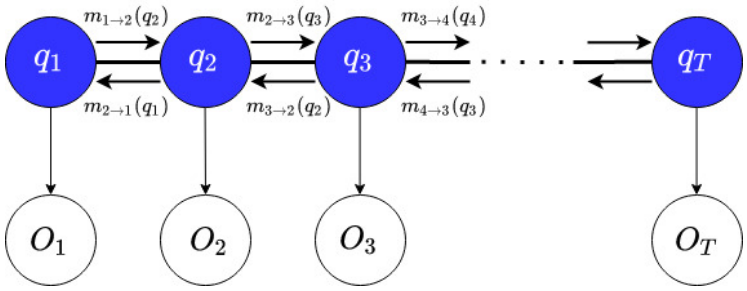

Figura 17 – Passagem de mensagens no algoritmo *forward-backward*. Adaptado de [Ghojogh et al., 2019].

A partir das definições dos fatores, calcularemos as mensagens *forward* $\mathfrak{F}(q_t)$ e *backward* $\mathfrak{B}(q_t)$ desenvolvendo o algoritmo de soma-produto, que propaga a mensagem $m_{t\to(t+1)}(q_{(t+1)})$ pela vizinhança de cada fator de acordo com a Figura 16. A Equação (19) apresenta o processamento *forward* [Bishop, 2006; Koller and Friedman, 2009].

$$
\begin{aligned}
m_{t,t+1}(q_{t+1}) &= \frac{1}{Z} \sum_{q} \prod_{t=1}^{T-2} \varphi_t(q_t) \prod_{t=2}^{T-2} \psi_{t-1,t}(q_{t-1}, q_t) m_{(t-1)\to(t-2)} \\
&= \frac{1}{Z} \sum_{q} \sum_{q_1} \varphi_1(q_1)\psi_{1,2}(q_1, q_2) \times \prod_{t=2}^{T-2} \varphi_t(q_t) \prod_{t=3}^{T-2} \psi_{t-1,t}(q_{t-1}, q_t) m_{(t-1)\to(t-2)}(q_{t-2}) \\
&= \frac{1}{Z} \sum_{q} m_{1\to2}(q_2) \prod_{t=2}^{T-2} \varphi(q_t) \prod_{t=3}^{T-2} \psi_{(t-1),(t)}(q_{t-1}, q_t) m_{(t-1)\to(t-2)}(q_{(t-2)}) \\
&= \sum_{q_t} m_{(t-1)\to t}(q_t)\varphi_t(q_t)\psi_{t,(t+1)}(q_t, q_{(t+1)})
\end{aligned} \tag{19}
$$

onde $Z = \sum_{q_t} m_{(t-1)\to t}(q_t)\varphi_t(q_t)m_{(t+1)\to t}(q_t)$.

Podemos realizar algumas deduções para verificar a validade desse desenvolvimento do algoritmo *forward* em comparação com o que vimos anteriormente.

$$\begin{aligned} m_{1\to 2}(q_2) &= \sum_{q_1} \varphi_1(q_1)\psi_{12}(_{1,2}) = \sum_{q_1} P(q_1, O_1)P(q_{t+1}|q_t) \\ &= \sum_{q_1} P(q_1, O_1, q_2) = P(O_1, q_2) \end{aligned} \tag{20}$$

$$\begin{aligned} m_{2\to 3}(q_3) &= \sum_{q_2} \varphi_2(q_2)\psi_{23}(_{2,3})m_{1\to 2}(q_2) = \sum_{q_2} P(O_2|q_2)P(q_3|q_2)P(O_1|q_2) \\ &= \sum_{q_2} P(q_3, q_2, O_2, O_1) \end{aligned} \tag{21}$$

A Equação (22) assegura que, se prosseguirmos o avanço temporal $1, 2, \ldots, t$, alcançaremos a mesma solução do algoritmo *forward*, desenvolvido na Equação (15).

$$m_{(t-1)\to t}(q_t) = P(O_1, O_2, \ldots, O_{(t-1)}, q_t) \equiv \mathfrak{F}(q_t) \tag{22}$$

De forma similar, podemos obter o algoritmo *backward*, de acordo com a Equação (23), utilizando o algoritmo soma-produto em seu desenvolvimento [Bishop, 2006; Koller and Friedman, 2009].

$$\begin{aligned} m_{t\to(t-1)}(q_{(t-1)}) &= \frac{1}{Z}\sum_{q}\prod_{t=1}^{T}\varphi_t(q_{ti})\prod_{t=2}^{T}(q_{t-1}, q_t) \\ &= \frac{1}{Z}\sum_{q}\prod_{t=1}^{T}\varphi_t(q_{ti})\prod_{t=2}^{T-1}\psi_{(t-1),t}(_{(t-1)}, q_t)\varphi_t(q_t)\psi_{(t-1),t}(q_{(t-1)}, q_t) \\ &= \frac{1}{Z}\sum_{q}\sum_{q_t}\prod_{t=1}^{T-1}\varphi_t(q_t)\prod_{t=2}^{t-1}\psi_{(t-1),t}(q_{(t-1)}, q_t)\varphi_t(q_t)\psi_{(t-1),t}(q_{(t-1)}, q_t) \\ &= \frac{1}{Z}\sum_{q}\prod_{t=1}^{t-1}\varphi_t(q_t)\prod_{t=2}^{t-1}\psi_{(t-1),t}(q_{(t-1)}, q_t)\sum_{q_t}\varphi_t(q_t)\psi_{(t-1),t}(q_{(t-1)}, q_t) \\ &= \frac{1}{Z}\sum_{q}\prod_{t=1}^{T-1}\varphi_t(q_t)\prod_{t=2}^{T-1}\psi_{(t-1),t}(q_{(t-1)}, q_t)\sum_{q_T}\varphi_T(q_T)\psi_{(T-1),T}(q_{(T-1)}, q_T) \end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{Z} \sum_{q} \prod_{t=1}^{T-2} \varphi_t(q_t) \prod_{t=2}^{T-2} \psi_{(t-1),t}(q_{(t-1)}, q_t) \times \\
&\times \sum_{q_{(T-1)}} \varphi_{(T-1)}(q_{(T-1)}) \psi_{(T-2),(T-1)}(q_{(T-2)}, q_{(T-1)}) m_{T \to (T-1)}(q_{(T-1)}) \\
&= \sum_{q_t} \psi_{(t-1),t}(q_{(t-1)}, q_t) \varphi_t(q_t) m_{(t+1) \to t}(q_t)
\end{aligned} \tag{23}
$$

onde $Z = \sum_{q_t} m_{(t-1)\to t}(q_t)\varphi_t(q_t)m_{(t+1)\to t}(q_t)$.

Também apresentaremos algumas deduções a fim de validar a solução do algoritmo *backward* utilizando fatores com o desenvolvimento previamente apresentado [Bishop, 2006; Koller and Friedman, 2009]:

$$
\begin{aligned}
m_{T \to (T-1)}(q_{(T-1)}) &= \sum_{q_T} \varphi_T(q_T) \psi_{(T-1),T}(q_{(T-1)}, q_T) = \sum_{q_T} P(O_T|q_T) P(q_T|q_{(T-1)}) \\
&= \sum_{q_T} P(q_T, O_T|q_{(T-1)}) = P(O_T|q_{(T-1)})
\end{aligned} \tag{24}
$$

$$
\begin{aligned}
m_{(T-1) \to (T-2)}(q_{(T-2)}) &= \sum_{q_{(T-1)}} \varphi_{(T-1)}(q_{(T-1)}) \psi_{(T-2),(T-1)}(q_{(T-2)}, q_{(T-1)}) \\
&= \sum_{q_{(T-1)}} P(O_{(T-1)}|q_{(T-1)}) P(q_{(T-1)}|q_{(T-2)}) P(O_T|q_{(T-1)}) \\
&= \sum_{q_{(T-1)}} P(O_{(T-1)}, O_T, q_{T-1}|q_{(T-2)}) = P(O_{(T-1)}, O_T|q_{(T-2)})
\end{aligned} \tag{25}
$$

Ao fim, a Equação (26) assegura que se prosseguirmos o retrocesso temporal $T, T-1, \ldots, t+1$, obteremos a mesma solução do algoritmo *backward* apresentada na Equação (16).

$$
m_{(t+1) \to t}(q_t) = P(O_T, \ldots, O_{(t+1)}|q_t) \equiv \mathfrak{B}(q_t) \tag{26}
$$

A solução do problema de avaliação é obtida pelo processamento individual de uma das etapas do algoritmo *forward-backward*, $\mathfrak{F}(q_t)$ ou $\mathfrak{B}(q_t)$.

4.2.2 Problema de decodificação de estados

Para solucionar o problema de decodificação, poderíamos nos concentrar na Equação (14) para identificar a sequência que gera a máxima probabilidade *a posteriori* instantânea, denotada por $\boldsymbol{\chi} = \underset{1 \leq t \leq T}{\textit{arg max}}[\gamma(q_t)]$, entretanto a adoção dessa solução exige que o sistema seja ergódico e $a_{ij},\ \forall 1 \leq i, j \leq N$ [Ghojogh et al., 2019]. Para contornar esses critérios, utilizamos o algoritmo de Viterbi, que encontra essa sequência maximizando a probabilidade conjunta $P(\boldsymbol{O}, \boldsymbol{\chi}|\lambda)$. Enquanto o algoritmo *forward-backward* é baseado no algoritmo soma-produto, Viterbi é baseado no algoritmo máximo-produto (ou máxima-soma, se transformarmos os produtos utilizando logaritmos) [Bishop, 2006; Koller and Friedman, 2009; Ghojogh et al., 2019]. Os algoritmos são similares, porém no algoritmo máximo-produto o operador de soma é substituído pelo operador de máximo. Dessa forma, o algoritmo de Viterbi também possui o mecanismo semelhante ao algoritmo *forward-backward*. Neste contexto, a etapa *forward* corresponde à atuação das variáveis $\nu_t(j) = \underset{q_1, q_2, \ldots, q_{t-1}}{\max} P(q_1, q_2, \ldots, q_t = S_j, O_1, O_1, O_2, \ldots, O_t|\lambda)$ e $\kappa_t(j)$, onde $\nu_t(j)$ armazena a probabilidade do caminho mais provável que leva ao estado S_j no instante t e $\kappa_t(j)$ armazena o índice dessa sequência de estados. A etapa *backward*, por sua vez, analisa os percursos realizados ao longo da treliça utilizando a técnica de recursão *backtracking* para encontrar a mais provável sequência de estados. O Algoritmo 3 apresenta sistematicamente todos os passos do algoritmo de Viterbi [Ghojogh et al., 2019].

Igualmente, podemos adotar a perspectiva de inferência em modelos gráficos para analisarmos o algoritmo de Viterbi [Bishop, 2006; Koller and Friedman, 2009; Ghojogh et al., 2019]. Utilizamos o mesmo conjunto de fatores definidos na Equação (18) para propagar as mensagens ao longo da treliça. Considere o grafo da Figura 18, que apresenta o conjunto de fatores $\psi_{i,j}$ que conecta os nós S_i aos nós S_j no tempo t. A Equação (27) apresenta a passagem de mensagem entre os fatores e os estados.

$$m_{i \to j}(q_t = S_j) = \max_q \varphi_i(q_{(t-1)} = S_i)\psi_{i,j}(q_{(t-1)} = S_i, q_t = S_j) \prod_{k \in N(S_i) \setminus S_j} m_{k \to i}(q_{(t-1)} = S_i) \tag{27}$$

A Equação (28) apresenta a passagem de mensagem entre fatores e variáveis.

Algoritmo 3 – *Viterbi*

Entrada: $\lambda = (\mathbf{\Pi}, \boldsymbol{A}, \mathbf{B})$
1 $\nu(q_1 = S_i) = \Pi_i b_i(O_1), \ \ \forall i \in \{1, \ldots, N\}$
2 $\kappa(q_1 = S_i) = 0, \ \ \forall i \in \{1, \ldots, N\}$
3 **for** *estado j de 1 até N* **do**
4 **for** *tempo t de 2 até T* **do**
5 $\nu(q_t = S_j) = \max_{1 \leq i \leq N} (\nu(q_{t-1} = S_i) a_{ij}) b_j(O_t)$
6 $\kappa(q_t = S_j) = \arg\max_{1 \leq i \leq N} (\nu(q_{t-1} = S_i) a_{ij})$
7 //Conclusão
8 $p^* = \max_{1 \leq i \leq N} \nu_T(i)$
9 $s^*(T) = \arg\max_{1 \leq i \leq N} \nu_T(i)$
10 // Uso da recursão backtracking
11 **for** *tempo t de (T-1) até 1* **do**
12 $s^*(t) = \kappa(q_{(t+1)})$
13 $P(\boldsymbol{O}, \boldsymbol{\chi}|\lambda) = p^*$
Saída: $P(\boldsymbol{O}, \boldsymbol{\chi}|\lambda), \boldsymbol{\chi} = \{s^*(1), \ldots, s^*(T)\}$

$$m_{j \to i}(q_{(t-1)} = S_i) = \varphi_i(q_{(t-1)} = S_i) \prod_{k \in N(i)} m_{k \to i}(q_{(t-1)} = S_i) \tag{28}$$

Como discutimos, o uso da recursão *backtracking* determina o caminho ótimo da treliça que determina a solução do problema de decodificação dos modelos ocultos de Markov. A Equação (29) apresenta essa aplicação em modelos gráficos.

$$\kappa_{i \to j}(q_t = S_j) = \max_q \varphi_i(q_{(t-1)} = S_i) \psi_{i,j}(q_{(t-1)} = S_i, q_t = S_j) \prod_{k \in N(S_i) \setminus S_j} m_{k \to i}(q_{(t-1)} = S_i) \tag{29}$$

4.2.3 Problema de aprendizado em modelos ocultos de Markov

Até o momento, assumimos que todos os parâmetros λ do modelo eram conhecidos, o que permitiu construir as soluções anteriores. Para concluir, discutiremos agora a solução do problema de aprendizado dos modelos ocultos de Markov. O aprendizado envolve ajustar os parâmetros λ do modelo a fim de maximizar a probabilidade das observações condicionadas a este modelo $P(\boldsymbol{O}|\lambda)$. Rabiner and Juang [1986] afirmam

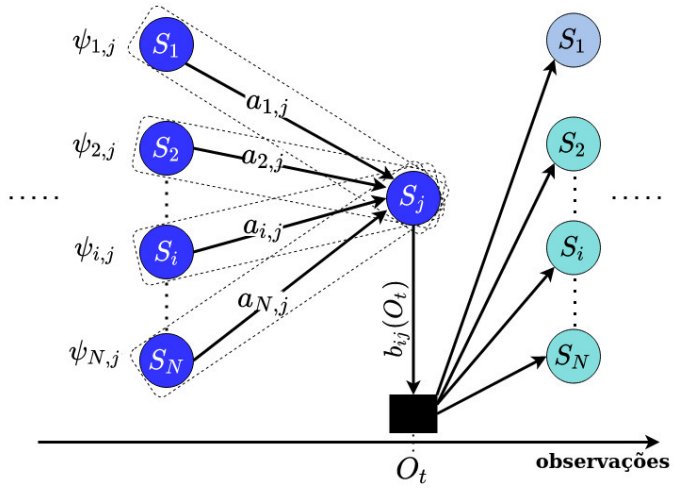

Figura 18 – Modelagem do algoritmo de Viterbi como um algoritmo máximo-produto em um grafo de fatores $\psi_{t,t+1}(q_t, q_{t+1})$ e $\varphi(q_t)$. Adaptado de [Ghojogh et al., 2019].

que este é o problema canônico mais difícil, pois não se conhece nenhuma forma analítica de obtenção da máxima verossimilhança sob a forma da Equação (30).

$$\overline{\lambda} = \underset{\lambda \in \boldsymbol{\Theta}}{arg\ max} P(\boldsymbol{O}|\lambda) \tag{30}$$

onde $\overline{\lambda}$ é o parâmetro que fornece o máximo global da probabilidade *a posteriori* dos dados durante a busca no espaço de possíveis valores desse parâmetro $\boldsymbol{\Theta}$.

Vemos essa solução global na Figura 19, para a qual $\overline{\lambda}_3$ fornece o valor máximo da probabilidade *a posteriori* $P(\boldsymbol{O}|\lambda)$.

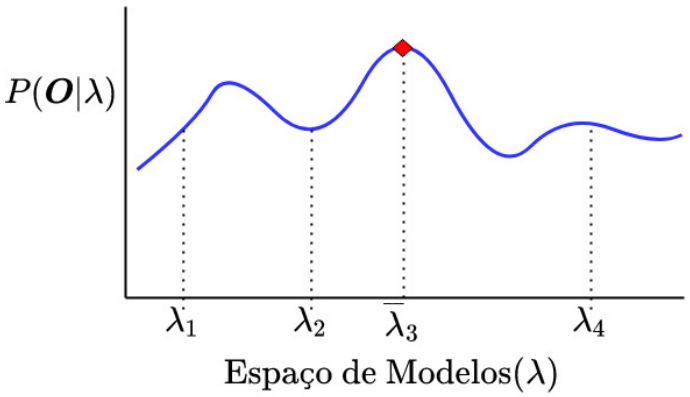

Figura 19 – Espaço de busca de modelos pelo algoritmo de Baum-Welch, que possivelmente decidirá sobre um máximo local.

Para solucionar este problema, utilizamos o algoritmo de Baum-Welch, um caso especial do algoritmo de maximização de expectativa (EM, do inglês *Expectation Maxi-*

mization). O algoritmo adota uma configuração inicial de parâmetros que serão iterativamente ajustados para maximizar localmente a verossimilhança dos dados [Seymore et al., 1999]. Portanto, não estamos lidando com um método que nos fornecerá o máximo global apresentado na Figura 19.

Uma vez que o processamento do método é iterativo, iniciamos um contador em $i = 1$ e configuramos um valor inicial para os parâmetros do modelo $\lambda^{(0)}$. É importante ressaltar que o algoritmo de Baum-Welch é bastante sensível a esta configuração inicial dos parâmetros. Note também que trataremos do desenvolvimento do algoritmo de maximização de expectativa considerando o conjunto de dados completo. Discriminamos abaixo, adotando a abordagem da função $Q(\lambda^{(i)}, \lambda^{(i-1)})$, as etapas de processamento que um algoritmo típico de maximização de expectativa (EM) realiza [Bilmes et al., 1998]:

- *Expectation*: baseado no valor inicial do parâmetro do modelo $\lambda^{(0)}$, são calculadas as probabilidades *a posteriori* das variáveis latentes $P(\boldsymbol{S}|\boldsymbol{O}, \lambda^{(i-1)})$ a fim de calcular a função $Q(\lambda^{(i)}, \lambda^{(i-1)})$:

$$Q(\lambda, \lambda^{(i-1)}) = \sum_{q_t \in \boldsymbol{\Upsilon}} \log P(\boldsymbol{O}, q_t|\lambda^{(i)}) P(\boldsymbol{O}, q_t|\lambda^{(i-1)}) \tag{31}$$

  onde $\boldsymbol{\Upsilon}$ é o espaço de todas as sequências de estados de tamanho T.

- *Maximization*: é computada a maximização de expectativa da função Q calculada no passo anterior, o que permite a atualização dos valores do parâmetro $\lambda^{(i)}$:

$$\lambda^{(i)} = \arg\max_{\lambda \in \boldsymbol{\Theta}} Q(\lambda^{(i)}, \lambda^{(i-1)}). \tag{32}$$

Vamos agora discutir os resultados que são obtidos em cada passo desse processamento, de acordo com Bilmes et al. [1998]. Considerando uma sequência de estado q qualquer representando a probabilidade *a posteriori* inicial $P(\boldsymbol{O}, q_t|\lambda^{(0)})$, encontramos:

$$P(\boldsymbol{O}, q_t|\lambda^{(i)}) = \Pi_0 \prod_{t=1}^{T} a_{(q_{(t-1)}=S_i, q_t=S_j)} b_j(O_t) \tag{33}$$

Assim, a função $Q(\lambda^{(i)}, \lambda^{(i-1)})$ torna-se:

$$Q(\lambda^{(i)}, \lambda^{(i-1)}) = \sum_{q_t \in \boldsymbol{\Upsilon}} \log \Pi_0 P(\boldsymbol{O}, q_t|\lambda^{(0)}) + \sum_{q_t \in \boldsymbol{\Upsilon}} \left(\sum_{t=1}^{T} \log a_{q_{(t-1)}=S_i, q_t=S_j} \right) P(\boldsymbol{O}, q_t|\lambda^{(0)}) + \\ + \sum_{q_t \in \boldsymbol{\Upsilon}} \left(\sum_{t=1}^{T} \log b_j(O_t) \right) P(\boldsymbol{O}, q_t|\lambda^{(0)}) \tag{34}$$

O primeiro termo da Equação (34) torna-se:

$$\sum_{q_t \in \boldsymbol{\Upsilon}} \log \Pi_0 P(\boldsymbol{O}, q_t|\lambda^{(0)}) = \sum_{i=1}^{T} \log \Pi_i P(\boldsymbol{O}, q_0 = S_i|\lambda^{(0)}) \tag{35}$$

De acordo com a Equação (35), estaríamos repetidamente selecionando os valores de q_0 para todo $q \in \boldsymbol{\Upsilon}$, o que torna esse termo somente uma expressão marginal para o tempo $t = 0$. Em extensão, adicionamos os multiplicadores de Lagrange ρ, utilizando a restrição $\sum_i \Pi_i = 1$, e definimos a derivada igual a 0:

$$\frac{\partial}{\partial \Pi_i} \left(\sum_{i=1}^{N} \log \Pi_i P(\boldsymbol{O}, q_0 = S_i|\lambda^{(0)}) + \rho(\sum_{i=1}^{N} \Pi_i - 1) \right) = 0 \tag{36}$$

Após a obtenção da solução da derivada, aplica-se o somatório sobre o índice i para obter ρ e, resolvendo para Π_i, obtemos:

$$\Pi_i = \frac{P(\boldsymbol{O}, q_0 = S_i|\lambda^{(0)})}{P(\boldsymbol{O}|\lambda^{(0)})} \tag{37}$$

O segundo termo da Equação (34) torna-se:

$$\sum_{q \in \boldsymbol{\Upsilon}} \left(\sum_{t=1}^{T} \log a_{q_{(t-1)}=S_i, q_t=S_j} \right) P(\boldsymbol{O}, q_t|\lambda^{(t-1)}) = \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{t=1}^{T} \log a_{ij} P(\boldsymbol{O}, q_{(t-1)} = S_i, q_t = S_j|\lambda^{0}) \tag{38}$$

Como fizemos antes para tratar o primeiro termo da Equação (34), adicionamos os multiplicadores de Lagrange à Equação (38) sob a restrição $\sum_{j=1}^{N} a_{ij} = 1$ para obtermos:

$$a_{ij} = \frac{\sum_{t=1}^{T} P(\boldsymbol{O}, q_{(t-1)} = S_i, q_t = S_j|\lambda^{(0)})}{\sum_{t=1}^{T} P(\boldsymbol{O}, q_{(t-1)} = S_i|\lambda^{(0)}} \tag{39}$$

O terceiro termo da Equação (34), torna-se:

$$\sum_{q_t \in \Upsilon} \left(\sum_{t=1}^{T} \log b_j(\boldsymbol{O}) \right) P(\boldsymbol{O}, q_t | \lambda^{(0)}) = \sum_{i=1}^{N} \sum_{t=1}^{T} \log b_i(O_t) P(\boldsymbol{O}, q_t = S_i | \lambda^{(0)}) \quad (40)$$

Por fim, também aplicamos os multiplicadores de Lagrange sob a restrição $\sum_{j=1}^{M} b(v_j) = 1$. Baseado no dicionário de amostras do modelo, somente as observações iguais a v_k contribuem com o k-ésimo valor de probabilidade:

$$b_i(O_t = v_k) = \frac{\sum_{t=1}^{T} P(\boldsymbol{O}, q_t = S_i | \lambda^{(0)}) \delta_{O_t, v_k}}{\sum_{t=1}^{T} P(\boldsymbol{O}, q_t = S_i | \lambda^{(0)})} \quad (41)$$

Os passos *Expectation* e *Maximization* do algoritmo são repetidos alternadamente até que um critério de parada seja atendido, por exemplo: $||\lambda^{(i)} - \lambda^{(i-1)}|| < \epsilon$, onde ϵ é um limiar do processamento.

4.3- Sumário

Abordamos ao longo deste capítulo os modelos ocultos de Markov, um método robusto de processamento de séries temporais que satisfaz a natureza do problema de detecção e diagnóstico de anomalias discutido neste trabalho. A Figura 12 forneceu os subsídios para compreendermos basicamente os mecanismos que atuam neste método. A seguir, discutimos os fundamentos teóricos que abrangem os três problemas canônicos do modelo, que serão extensivamente utilizados no desenvolvimento de um dos fluxos de trabalho. A apresentação da abordagem pelo modelo gráfico possibilitou a compreensão da propagação de crenças probabilísticas pelo grafo que compõe o processamento dos modelos ocultos de Markov, da qual derivamos as soluções dos problemas canônicos.

5- Redução de Dimensionalidade

O desenvolvimento de técnicas de pré-processamento é essencial nas abordagens que utilizam modelos baseados em dados, como será visto durante a descrição da etapa metodológica. Dentre tais técnicas destacam-se as de redução de dimensionalidade. A redução de dimensionalidade reúne métodos de seleção de características e extração de características. A importância da aplicação dessas técnicas é revelada quando se manipula um conjunto de dados de elevada dimensionalidade, cujo processamento demanda a retirada de características irrelevantes e redundantes para o modelo. A técnica de seleção de características como um problema de otimização de multiobjetivo é apresentada. O algoritmo NSGA II processa a minimização de duas funções objetivo, cujo resultado será a formação do subconjunto de características $\mathcal{F}'$ utilizado no processamento do modelo.

5.1- Considerações Gerais

A aplicação da redução de dimensionalidade do conjunto de dados é uma importante etapa durante o pré-processamento. O conceito de redução de dimensionalidade contempla um conjunto de técnicas aplicadas com o objetivo de remover ruídos e características redundantes [Alelyani et al., 2018]. Há duas principais categorias de técnicas de redução de dimensionalidade: extração de características e seleção de características.

A extração de características cria um novo espaço a partir de transformações ou combinações das características contidas no vetor de dados de entrada, geralmente resultando em um subespaço de características com dimensão inferior que preserva as informações relevantes contidas no espaço orignal [Khalid et al., 2014]. A extração de características é composta por técnicas que incluem Análise de Componentes Principais (PCA), Análise do Discriminante Linear (LDA) e Decomposição em Valores Singulares (SVD) [Alelyani et al., 2018].

De acordo com [Xue et al., 2015], a seleção de características constrói um subconjunto de características a partir do conjunto original, sem aplicar quaisquer transformações,

com o objetivo de criar um subconjunto ótimo de variáveis do problema [Guyon and Elisseeff, 2003]. Há diferentes motivações para a sua aplicação: remoção de dados irrelevantes, melhora do desempenho e acurácia do modelo, redução da complexidade e melhor compreensão do modelo [Guyon and Elisseeff, 2003].

5.2- Seleção de Características

Seja o domínio de um problema determinado por um conjunto de dados que contém uma série temporal multivariada com dimensão D, que fornece o conjunto de vetores de valor $\boldsymbol{f} = (f_1, f_2, \ldots, f_D)$ de um conjunto de características $\boldsymbol{\mathcal{F}} = (F_1, F_2, \ldots, F_D)$, onde D é a quantidade de características dessa série temporal. A seleção de características é formalizada por [Yu and Liu, 2004]. Seja $\boldsymbol{\mathcal{F}'} \subset \boldsymbol{\mathcal{F}}$ um subconjunto de características de $\boldsymbol{\mathcal{F}}$ e $\boldsymbol{f'}$ o conjunto de vetores de valor de $\boldsymbol{\mathcal{F}'}$. O processo de seleção de características fornecerá, ao seu fim, um conjunto mínimo de atributos $\boldsymbol{\mathcal{F}'}$ tal que $P(\boldsymbol{\mathcal{C}}|\boldsymbol{\mathcal{F}'} = \boldsymbol{f'}) \approx P(\boldsymbol{\mathcal{C}}|\boldsymbol{\mathcal{F}} = \boldsymbol{f})$, onde $P(\boldsymbol{\mathcal{C}}|\mathcal{F}' = \boldsymbol{f'})$ e $P(\boldsymbol{\mathcal{C}}|\boldsymbol{\mathcal{F}} = \boldsymbol{f})$ são as distribuições de probabilidade condicional das l possíveis classes $c_1, c_2, \ldots, c_l | c_l \in \boldsymbol{\mathcal{C}}$.

Conjuntos de dados com elevada dimensionalidade D são analisados a fim de verificar a relevância de cada característica para a construção do modelo. As características podem ser classificadas em três características, segundo a sua relevância: forte relevância, fraca relevância e irrelevante [John et al., 1994]. Seja $\mathbf{S}_i = \boldsymbol{\mathcal{F}} - \{F_i\}$, as seguintes definições formalizam o conceito de relevância das características.

Definição 1 (forte relevância) Uma característica F_i tem forte relevância se e somente se:

$$P(\boldsymbol{\mathcal{C}}|F_i, \mathbf{S}_i) \neq P(\boldsymbol{\mathcal{C}}|\mathbf{S}_i).$$

A distribuição de probabilidade condicional da classe $\boldsymbol{\mathcal{C}}$ dado F_i é diferente da distribuição de probabilidade das classe $\boldsymbol{\mathcal{C}}$ quando F_i é eliminado, revelando a sua importância.

Definição 2 (fraca relevância) Uma característica F_i tem fraca relevância se e somente se:

$$P(\boldsymbol{\mathcal{C}}|F_i, \mathbf{S}_i) = P(\boldsymbol{\mathcal{C}}|\mathbf{S}_i), \text{ e}$$

$$\exists\ \mathbf{S}'_i \subset \mathbf{S}_i, \text{ tal que } P(\boldsymbol{\mathcal{C}}|F_i, \mathbf{S}_i) \neq P(\boldsymbol{\mathcal{C}}|\mathbf{S}_i)$$

Em ao menos uma seleção $\mathbf{S}_i$, a remoção de F_i altera a distribuição de probabilidade condicional $P(\mathcal{C}|\mathbf{S}_i)$.

Corolário 1 (irrelevância) Uma característica F_i é irrelevante se e somente se:

$$P(\boldsymbol{\mathcal{C}}|F_i, \mathbf{S}_i) = P(\boldsymbol{\mathcal{C}}|\mathbf{S}_i)$$

De acordo com [García et al., 2015], a seleção de características é um problema de busca. Dado um conjunto de dados de dimensão D, o espaço de busca para seleção ótima de características é da ordem 2^D [Dash and Liu, 1997]. A Figura 20 apresenta um exemplo de conjunto contendo três características, no qual o subconjunto ótimo seria uma alternativa entre o conjunto de características completo $\boldsymbol{\mathcal{F}}$ e o vazio, localizados nos extremos.

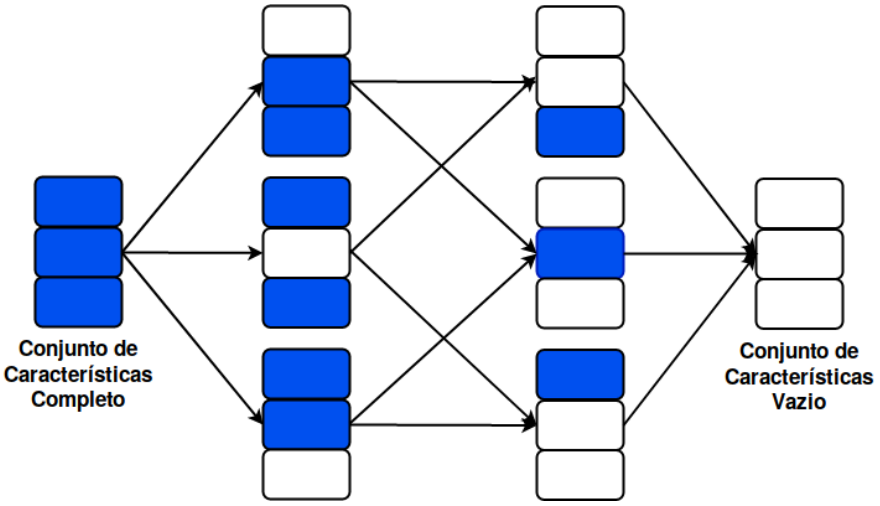

Figura 20 – Seleção de características como um problema de busca. Adaptado de [García et al., 2015]

De acordo com [Dash and Liu, 1997], os métodos de seleção de características devem cumprir as seguintes etapas durante a exploração do espaço de busca:

(a) gerar um subconjunto de características;

(b) aplicar no subconjunto uma função de avaliação;

(c) possuir um critério de parada;

(d) possuir mecanismo de validação do resultado.

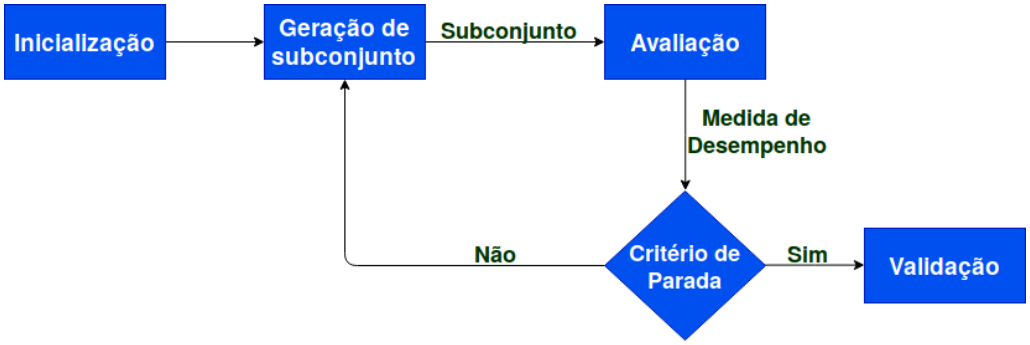

Figura 21 – Procedimentos realizados durante a busca do subconjunto ótimo durante a seleção de características.

A Figura 21 resume os procedimentos de busca durante a seleção de características.

Os métodos de seleção de características possuem diferentes estratégias de exploração do espaço de busca, como busca completa, busca heurística e busca aleatória, entre outras [García et al., 2015; Dash and Liu, 1997].

A busca completa avalia todas as combinações possíveis das características $F_i \in \mathcal{F}$. A complexidade da busca é $O(2^D)$, o que torna o seu uso impraticável em conjuntos de dados de elevada dimensionalidade D, embora somente a busca completa seja capaz de garantir a identificação do subconjunto de características ótimo.

A busca heurística é uma estratégia de seleção de características possivelmente sub-ótima que aplica heurísticas na execução da busca [Wang et al., 2016]. Diferente da busca completa, a busca heurística não explora todo o espaço de busca, prevalecendo a utilização de aleatorização e busca na vizinhança do subconjunto direcionada pela função de avaliação, que determina um percurso de comprimento D [Edelkamp and Schroedl, 2011; García et al., 2015]. Uma outra abordagem, a busca aleatória gera subconjuntos a partir da seleção aleatória de características [García et al., 2015].

O subconjunto F_i é selecionado atendendo um critério de avaliação. A interação entre a seleção desse subconjunto e o processamento do algoritmo de aprendizado é categorizada em três abordagens que se distinguem pelo desempenho do processamento [Khalid et al., 2014].

A abordagem Filtro realiza a seleção de características independente do algoritmo de aprendizado, que é empregado subsequentemente. Este pré-processamento no qual a abordagem atua é dividido em duas etapas: (a) aplicação de algum critério de avaliação no conjunto de dados (ganho de informação, medida de distância, medida de dependência e consistência) [García et al., 2015]. As características podem ser ranqueadas segundo

um critério de avaliação [Li et al., 2018]; (b) realização do treinamento do algoritmo de aprendizado utilizando o subconjunto F_i; realização do teste para a obtenção da acurácia de predição do algoritmo de aprendizado utilizando o conjunto de testes [García et al., 2015]. A Figura 22 apresenta o diagrama da abordagem filtro.

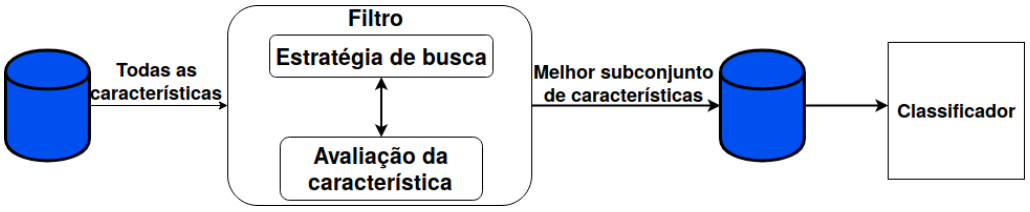

Figura 22 – Diagrama do método filtro.

A abordagem *Wrapper* conduz a seleção do subconjunto $\mathcal{F}'$ baseada do desempenho no algoritmo de aprendizado [Jović et al., 2015]. Quando o aprendizado é supervisionado, são adotados classificadores (e.g. *Naive* Bayes, SVM, *Decision Tree*); e, quando o aprendizado é não-supervisionado, são utilizados algoritmos de clusterização (e.g. k-means, KNN). Essa avaliação utilizando $\mathcal{F}'$ como conjunto de treinamento é realizada iterativamente até o atendimento do critério de avaliação [Khalid et al., 2014]. A acurácia é um dos critérios comumente utilizados nessa avaliação [Kohavi and John, 1997]. Em conjuntos de dados de elevada dimensionalidade D, a aplicação da abordagem *Wrapper* possui elevado custo computacional. A Figura 23 apresenta o diagrama da abordagem.

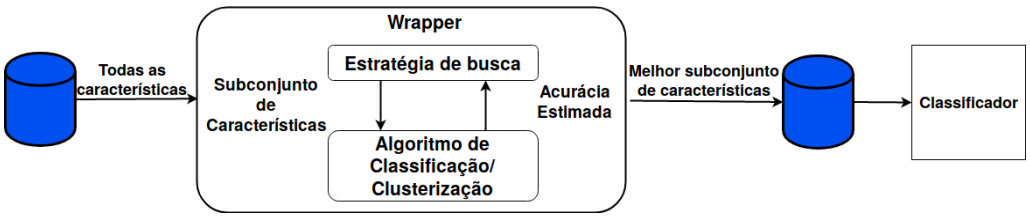

Figura 23 – Diagrama do método *Wrapper*.

O método embarcado é similar ao método *Wrapper*, entretanto a seleção de característica é realizada durante o aprendizado do algoritmo utilizado [Tang et al., 2014]. Assim, dois objetivos são alcançados simultaneamente: convergência do algoritmo de aprendizado e seleção das características [Quinlan, 1993]. O diagrama do método é apresentado na Figura 24.

A Seção 5.3 introduz o conceito da abordagem evolucionária para a seleção de características.

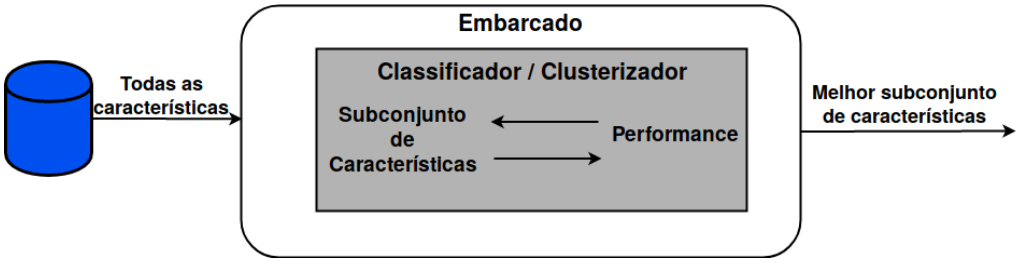

Figura 24 – Diagrama do método embarcado.

5.3- Abordagem Evolucionária na Seleção de Características

Os algoritmos evolucionários são um tópico interdisciplinar que envolve conceitos da biologia, inteligência artificial, otimização numérica e apoio à decisão [Back, 1996]. O comportamento desses algoritmos foi inspirado nas ideias propostas na teoria da evolução de Charles Darwin [Whitley, 1994; E. Goldberg and Henry Holland, 1988]. O modelo darwinista envolve uma população de indivíduos que interage com o objetivo de sobreviver ao meio e transmitir para as próximas gerações as suas características fenotípicas. A melhora da qualidade dessa população é potencializada pelo mecanismo de mutação, que provoca mudanças aleatórias nos genótipos desses indivíduos [Whitley, 1994]. Assim, o princípio da seleção natural age na condução do aprimoramento da população até a solução do problema [Back, 1996].

Neste trabalho, a seleção de características foi realizada utilizando os algoritmos genéticos, uma subclasse dos algoritmos evolucionários. Os algoritmos genéticos acrescentam ao escopo dos algoritmos evolucionários (discutido anteriormente) o mecanismo de elitismo, responsável pela seleção dos melhores indivíduos da população a fim de inseri-los intactos na população da geração subsequente. A Figura 25 apresenta o fluxograma detalhando o processamento do algoritmo genético.

É possível identificar duas grandes aplicações dos algoritmos genéticos: seleção de parâmetros para otimizar o desempenho de um sistema; teste e ajuste de parâmetros que reduzem a discrepância entre o modelo e os dados reais [Lawrence, 1991]. O trabalho [Tang et al., 1996] discute muitas dessas aplicações. O processo de aprendizado dos algoritmos genéticos adota a estratégia indutiva e adaptativa para a obtenção da solução [Vafaie and De Jong, 1992]. O aprendizado indutivo seleciona as amostras positivas e exclui as negativas. Para construir a regra decisória utilizada na seleção, adota-se a

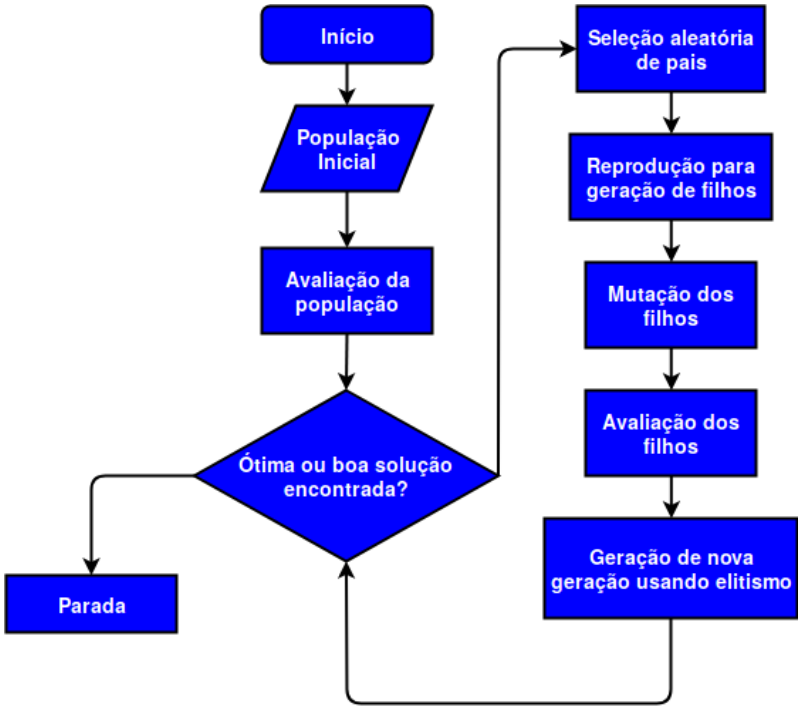

Figura 25 – Fluxograma do algoritmo genético.

inferência indutiva, que generaliza uma regra a partir de uma amostra reduzida de dados [Wang et al., 1999].

Os algoritmos genéticos envolvem técnicas que realizam a busca em um domínio independente do problema original, o que justifica a sua adoção em problemas sobre os quais não há domínio do conhecimento [Vafaie and De Jong, 1997]. A seleção de características é um problema cujo processamento deve lidar com limitado conhecimento sobre o domínio e presença de ruídos. O primeiro trabalho que adotou algoritmos genéticos como solução para a seleção de características foi proposto por [Siedlecki and Sklansky, 1993]. Por esta abordagem, o processamento da busca heurística ocorre em um domínio formado por uma população de indivíduos que determinam em seu genótipo quais características integrarão o subconjunto $\mathcal{F}'$. Assim, o espaço de busca do algoritmo é formado por uma população $\mathcal{P}$ de p indivíduos, cada um identificado pelo seu genótipo $\varrho_p = \{\varrho_1, \ldots, \varrho_D\}$, onde ϱ_i determina a inclusão da i-ésima característica F_i no subconjunto $\mathcal{F}'$ se $\varrho_i = 1$ ou a sua exclusão se $\varrho_i = 0$.

Considere um conjunto de dados com quatro características. A Figura 26 apresenta o espaço de busca desse conjunto de dados como um diagrama de treliça. Cada nó representa um indivíduo da população portando o seu genótipo. O nó localizado no topo identifica o indivíduo que seleciona todas as características $\mathcal{F}' = \mathcal{F}$; e o nó da base

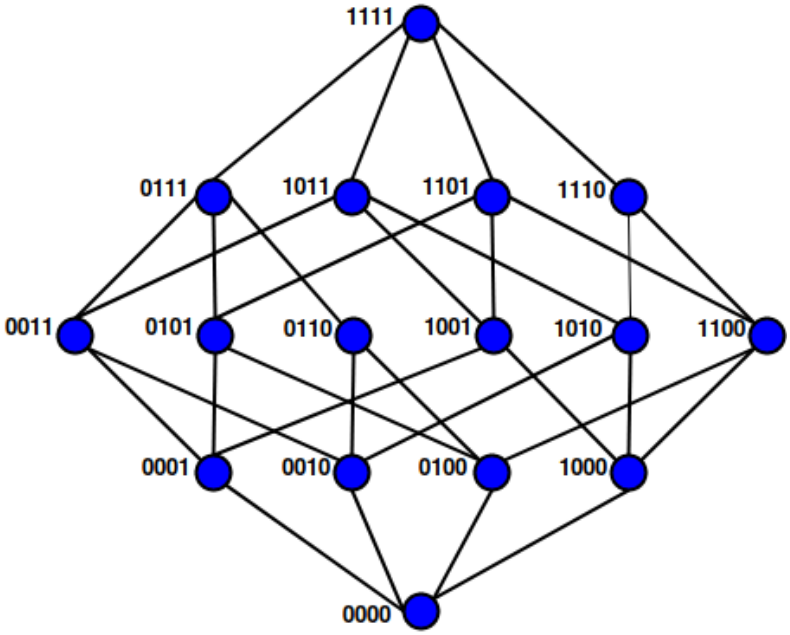

Figura 26 – Diagrama de treliça dos estados dos indivíduos que compõem o espaço de busca do algoritmo genético.

identifica o indivíduo que seleciona nenhuma característica $\boldsymbol{\mathcal{F}'} = \emptyset$. Os demais indivíduos estão localizados nos níveis segundo a quantidade de características selecionadas. A conexão entre os indivíduos ocorre somente entre nós que representam conjuntos de características e outros nós que representam subconjuntos destas características.

Diferentes métodos para a aplicação de algoritmos genéticos na exploração do espaço de soluções como apresentado na Figura 26 foram desenvolvidos recentemente [Lu et al., 2015; Welikala et al., 2015; Soufan et al., 2015; Emary et al., 2016; Ghareb et al., 2016], entre outros. Outros trabalhos abordam a busca heurística baseada em algoritmos genéticos para a seleção de características como um problema de otimização. De acordo com [Mukhopadhyay et al., 2013], a expressão da seleção de características $(\boldsymbol{\mathcal{F}}, J)$ como um problema de otimização é formalizada pela Equação (42).

$$J(\mathcal{F}^*) = min_{\boldsymbol{\mathcal{F}'} \in \boldsymbol{\mathcal{F}}} J(\boldsymbol{\mathcal{F}'}, \boldsymbol{X}) \tag{42}$$

onde $\boldsymbol{\mathcal{F}}^*$ é um subconjunto de características ótimo, $\boldsymbol{\mathcal{F}'}$ é um subconjunto de características, $\boldsymbol{\mathcal{F}}$ é o conjunto de características do conjunto de dados $\boldsymbol{\mathcal{D}}$ e $J : \boldsymbol{\mathcal{F}} \times \boldsymbol{\eta} \rightarrow (R)$ denota a aplicação de um critério de qualidade para avaliar o resultado da classificação/clusterização utilizando o subconjunto $\boldsymbol{\mathcal{F}'}$ em relação ao conjunto de pontos $X \in \boldsymbol{\eta}$ formados neste processamento.

Neste trabalho, a seleção de características é realizada pela aplicação de um método de otimização de Algoritmos Genéticos Multiobjetivo (MOGA), que identifica diferentes objetivos minimização/maximização para solucionar o problema de redução de dimensionalidade. A próxima Seção aborda os conceitos de uma otimização multiobjetivo.

5.4- Otimização Multiobjetivo

Quando a otimização envolve mais de um objetivo, muitas vezes conflituosos, estamos lidando com o problema de otimização multiobjetivo. Inúmeros problemas do mundo real possuem essa característica [Mukhopadhyay et al., 2013]. A seleção de características é um tipo de problema naturalmente multiobjetivo, dado que simultaneamente realiza a redução da dimensionalidade do problema e a minimização/maximização do valor de algum critério de qualidade do algoritmo de aprendizado, como apresentado na Equação (42). A forma geral de um problema de otimização multiobjetivo é apresentada na Equação (43) [Deb, 2001].

$$\begin{aligned} &\underset{x}{\text{Minimizar/Maximizar}} && f_m\boldsymbol{x} && m = 1, 2, \ldots, M \\ &\text{sujeito a} && g_j\boldsymbol{x} \geq 0 && j = 1, 2, \ldots, J \\ & && h_o\boldsymbol{x} = 0 && o = 1, 2, \ldots, O \\ & && x_i^{(L)} \leq x_i \leq x_i^{(U)} && i = 1, 2, \ldots, w \end{aligned} \tag{43}$$

onde a solução $\boldsymbol{x} = (x_1, x_2, \ldots, x_w)$ é um vetor de w variáveis limitadas pelos valores mínimo $x_i^{(L)}$ e máximo $x_i^{(U)}$; e os termos $g_j x$ e $h_o\boldsymbol{x}$ representam as restrições das M funções objetivos $f_m\boldsymbol{x}$, cujo objetivo pode ser de maximização ou minimização da variável $\boldsymbol{x}$.

É importante notar que a solução de problemas multiobjetivos não é única. Cada função objetivo $f_m\boldsymbol{x}$ possui a sua solução ótima. Assim, teremos M soluções ótimas $\boldsymbol{z}^* = \boldsymbol{f}^* = (f_1^*, f_2^*, \ldots, f_M^*)^T$. Portanto, é inexistente o vetor $\boldsymbol{x}$ que seja capaz de fornecer a solução ótima simultaneamente para todas as funções objetivo. Assim, é estabelecida uma relação de dominância entre as diferentes soluções dos problemas pertencentes ao espaço de busca. A utilização do operador $\lhd$ denota que entre duas soluções $i \lhd j$,

i é melhor do que j a respeito de um objetivo particular. Da mesma forma, $i \rhd j$, i é pior do que j neste objetivo. De acordo com [Deb, 2001], o conceito de dominância é determinado por:

Definição 1 (Dominância) Uma solução $x^{(1)}$ domina $x^{(2)}$ quando:

1. A solução $x^{(1)}$ não é pior do que $x^{(2)}$ para todo $j = 1, 2, \ldots, M$ objetivos, ou matematicamente $f_j(x^{(1)}) \not\rhd f_j(x^{(2)})$;
2. A solução $x^{(1)}$ é estritamente melhor do que $x^{(2)}$ em ao menos um objetivo $j = 1, 2, \ldots, M$, ou matematicamente $f_j(x^{(1)}) \lhd f_j(x^{(2)})$.

Dessa definição deriva o conceito de não dominância. Considerando o espaço de busca viável, aquelas soluções que não são dominadas por qualquer outra são incluídas no conjunto ótimo de Pareto. Em uma otimização multiobjetivo, o objetivo é a identificação deste conjunto, que fornece importantes *tradeoffs* para definir ganhos ou perdas de acordo com cada função objetivo f_m. Considere o espaço de soluções da função de *benchmarking* ZDT1 [Zhang and Li, 2007] mostrado na Figura 27. Os pontos na cor azul correspondem àqueles pertencentes ao conjunto de Pareto. Estes pontos atendem ao critério de não dominância. Durante o processamento são identificados diferentes conjuntos de Pareto, até que o ótimo seja alcançado. É importante notar que o conjunto ótimo de Pareto localiza-se na borda de um conjunto de soluções [Deb, 2001]. Caminhando ao longo dos pontos demarcados em azul, é possível realizar os *tradeoffs* de soluções em relação às funções objetivo f_1 e f_2.

A otimização multiobjetivo foi aplicada com diferentes abordagens para a solução de problemas evolucionários [Mukhopadhyay et al., 2013]. Os Algoritmos Evolucionários Multiobjetivo (MOEA) obtiveram importantes vantagens sobre os tradicionais Problemas de Otimização Multiobjetiva (MOP), por exemplo: o conjunto de técnicas fornecidas pelos MOEA é capaz de otimizar o direcionamento para as melhores soluções do problema no espaço de busca, mesmo em espaços grandes e complexos, sendo capaz de fornecer ricos *tradeoffs* devido à sua capacidade de trilhar diferentes soluções simultaneamente [Van Veldhuizen, 1999].

O NSGA II é uma implementação dos MOEA baseada em algoritmos genéticos. O NSGA II têm atuação sobre o operador de seleção no escopo do algoritmo tradicional do algoritmo genético, com a criação de subpopulações alocadas em conjuntos de Pareto [Mukhopadhyay et al., 2013; Deb et al., 2002]. Trabalhos recentes apresentaram soluções

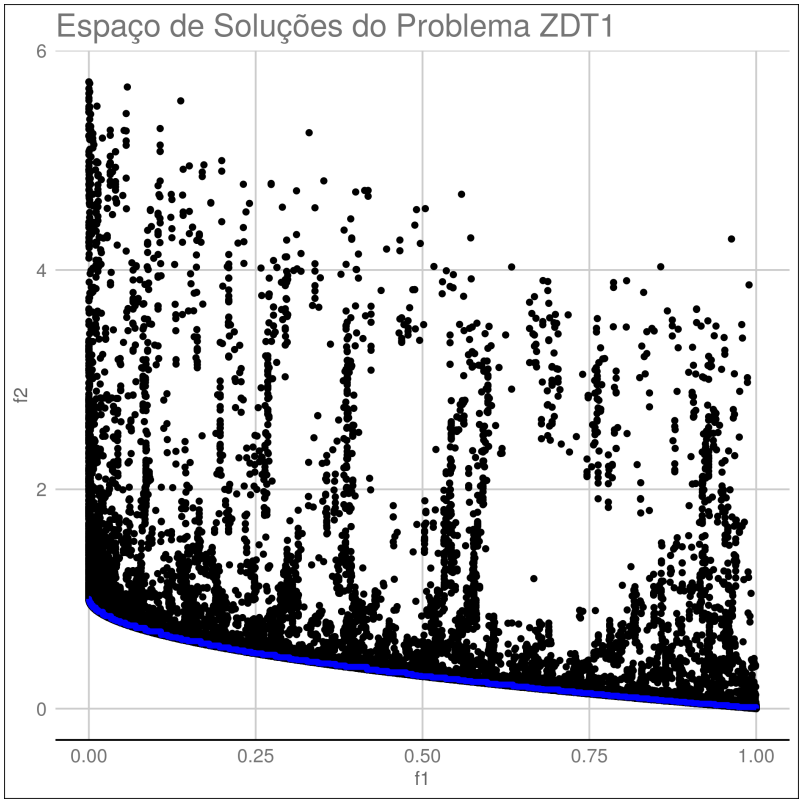

Figura 27 – Solução de Pareto da função ZDT1.

para a seleção de características adotando NSGA II [Hamdani et al., 2007; Huang et al., 2010; Soyel et al., 2011; Tekguc et al., 2009; Singh and Singh, 2017; Li et al., 2016]. A próxima Seção apresenta os fundamentos do algoritmo NSGA II e sua utilização na seleção de características.

5.5- Algoritmo NSGA II

A abordagem do algoritmo NSGA II apresentada em [Deb et al., 2002] aperfeiçoa a sua primeira versão apresentada em [Srinivas and Deb, 1994]. É possível estabelecer três características principais do algoritmo:

1. Adota elitismo no processo de seleção, o que auxilia na convergência;
2. Possui um mecanismo explícito de manutenção da diversidade de soluções na população;

3. Subdivide a população em diferentes conjuntos de soluções não-dominantes.

O algoritmo subdivide essa população $\mathcal{P}$ de soluções em diferentes frentes de Pareto $\mathscr{F}_i$. Essas frentes são criadas exaustivamente respeitando o princípio de não dominância até que todas as soluções pertençam a uma frente. Destaca-se que a primeira frente de Pareto formada contém as melhores soluções. A qualidade das soluções incluídas nas frentes decresce à medida que novas frentes são geradas [Deb et al., 2002].

Cada solução possui duas entidades relacionadas: (a) o número de dominação n_p conta o número de soluções que dominam a solução p (b) conjunto de soluções $\boldsymbol{\mathcal{S}}_p$ que são dominadas por p. Todas as soluções incluídas na primeira frente de Pareto $\mathscr{F}_1$ possuem $n_p = 0$, uma vez que elas elas não são dominadas.

A seguir descrevemos os passos da ordenação não-dominante, de acordo com o Algoritmo 4. Inicialmente, em relação a cada solução solução p (linha 1), o conjunto de soluções $\boldsymbol{\mathcal{S}}_p$ está vazio (linha 2) e o número de dominação n_p é igual a 0 (linha 3). Posteriormente, avaliamos as soluções q que dominam a solução p (linhas 4 –9). Aquelas soluções q dominadas por p são incluídas no conjunto $\boldsymbol{\mathcal{S}}_p$ (linha 6). Caso contrário, acrescentamos o valor do número de dominação n_p em relação a p (linha 8). Se o valor n_p for igual a 0, a solução p é inserida na primeira frente de Pareto (linhas 10 – 12). Após identificar as soluções pertencentes à primeira frente de Pareto $\mathscr{F}_1$, analisa-se cada solução $q \in \boldsymbol{\mathcal{S}}_p$, referente a todas as soluções p contidas em todas as frentes $\mathscr{F}_i$ (linhas 17 – 18). Selecionada cada solução q, decrementa-se o seu número de dominação n_q (linha 19). Se o número de dominação dessa solução q tornar-se 0, acrescentamos o seu valor de ranque q_{rank} e a incluímos em uma lista separada $\boldsymbol{Q}$ (linhas 20 – 22). Após a incursão em todas os conjuntos $\boldsymbol{\mathcal{S}}_p$ das soluções da primeira frente de Pareto e formação da lista $\boldsymbol{Q}$, incluímos os elementos dessa lista na formação da próxima frente de Pareto $\mathscr{F}_2$. Este procedimento deve ser repetido até a criação de todas as frentes de Pareto da população [Deb et al., 2002].

A preservação da diversidade dentro da população é um importante requisito durante o processamento de MOEA. Quanto mais diverso for o conjunto de soluções, mais informação do espectro de soluções poderá ser extraída pelo algoritmo de aprendizado, sem limitar-se a uma única parte do espaço de busca [Adra and Fleming, 2010]. A manutenção da diversidade do NSGA II é realizada pela aplicação do operador de comparação de *crowding distance* [Deb et al., 2002].

Algoritmo 4 – Ordenação-não-dominante($\mathcal{P}$)

```
1  foreach p ∈ P do
2  |  S_p = ∅
3  |  n_p = 0
4  |  foreach q ∈ P do
5  |  |  if p ≺ q then
6  |  |  |  S_p = S_p ∪ {q}
7  |  |  else if q ≺ p then
8  |  |  |  n_p = n_p + 1
9  |  end
10 |  if n_p = 0 then
11 |  |  p_rank = 1
12 |  |  F_1 = F_1 ∪ {p}
13 |  i = 1
14 end
15 while F_i ≠ ∅ do
16 |  Q = ∅
17 |  foreach p ∈ F_i do
18 |  |  foreach q ∈ S_p do
19 |  |  |  n_q = n_q − 1
20 |  |  |  if n_q = 0 then
21 |  |  |  |  q_rank = i + 1
22 |  |  |  |  Q = Q ∪ {q}
23 |  |  end
24 |  end
25 |  i = i + 1
26 |  F_i = Q
27 end
```

A manutenção da diversidade no algoritmo NSGA II é resumida em duas etapas: (a) cálculo da densidade de estimação da vizinhança das soluções; (b) aplicação do operador *crowding distance*. A densidade de estimação de uma solução consiste no cálculo da distância média desta solução e suas vizinhas, considerando cada função objetivo f_m. O conceito de um *N*-cubo, que envolveria a solução e suas vizinhas imediatas é sugerido por [Deb et al., 2002]. Por exemplo, na Figura 28, a solução $x^{(i)}$ tem em sua vizinhança as soluções $x^{(i-1)}$ e $x^{(i+1)}$, todas pertencendo à mesma frente de Pareto $\mathscr{F}$. O valor de i corresponde à do *N*-cubo.

Para calcular a *crowding distance* do conjunto, a partir da densidade de estimação de cada solução na frente de Pareto, primeiro é realizada a ordenação crescente das soluções em relação a cada função objetivo f_m. Para cada solução que está nos limites dessa ordenação por função objetivo $(l, u) \rightarrow \{i \in \mathbb{R} : l \leq i \leq u\}$ é atribuído o valor infinito

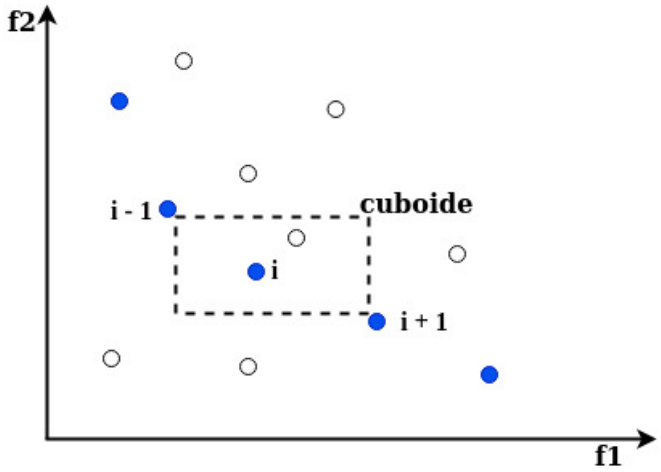

Figura 28 – Demonstração do cálculo da distância entre soluções.

em sua distância $\mathcal{I}[l]_{\text{distância}}, \mathcal{I}[u]_{\text{distância}} = \infty$. Às demais soluções é atribuído o valor absoluto resultante do cálculo da diferença entre as soluções vizinhas $\mathcal{I}[i+1].m - \mathcal{I}[i-1].m$ dividida pela amplitude de valores $f_m^{\text{max}} - f_m^{\text{min}}$ da função objetivo f_m. Nota-se que $\mathcal{I}[i].m$ se refere ao valor da função objetivo m para a solução $x^{(i)}$. O Algoritmo 5 detalha esses procedimentos realizados na etapa (a) para a manutenção da diversidade.

Algoritmo 5 – Atribuição-crowding-distância($\mathcal{I}$)

```
1  l = |I|                    % número de soluções em |I|
2  foreach i do
3  |  I[i] = 0
4  end
5  foreach objetivo m do
6  |  I = sort(I, m)
7  |  I[1]_distância = I[l]_distância = ∞
8  |  for i ← 2 to (l − 1) do
9  |  |  I[i]_distância = I[i]_distância + (I[i + 1]_m − I[i − 1]_m)/(f_m^max − f_m^min)
10 |  end
11 end
```

Após todos os membros do conjunto $\mathcal{I}$ terem a sua distância atribuída, uma solução $x^{(i)}$ poderá ser comparada com as demais pelo critério de proximidade [Deb et al., 2002].

A etapa (b) corresponde à utilização do operador *crowding distance*. O operador *crowding distance* ($\prec_n$) é utilizado no processo de seleção em diferentes etapas do processamento do algoritmo para formar as frentes de Pareto de maneira uniforme. Considere que cada solução i da população possui dois atributos:

1. ranqueamento baseado na não dominância i (solução da frente $\mathscr{F}_i$);

2. *crowding distance*.

Entre duas soluções com diferentes *rankings* de não dominância i, seleciona-se aquela com o menor valor i, ou melhor posicionada no *ranking*. Entre soluções com o mesmo ranque, seleciona-se aquela que possui a região menos densa, informação fornecida pelo índice *crowding distance*. Este processo decisório é formalizado a seguir, utilizando o operador $\prec_n$:

$$
\begin{gathered}
i \prec_n j \\
\text{se } (i_{\text{rank}} < j_{\text{rank}}) \\
\text{ou } [(i_{\text{rank}} = j_{\text{rank}}) \\
\text{e } (i_{\text{distância}} > j_{\text{distância}})]
\end{gathered}
$$

A execução do algoritmo tem início com a geração aleatória da população inicial $\boldsymbol{\mathcal{P}}_0$. O Algoritmo 4 é utilizado para formar as frentes de Pareto da população, a partir da qual as soluções são ranqueadas. Os demais operadores comuns dos algoritmos genéticos processam, sobre esta população inicial, a seleção, reprodução e mutação para gerar uma população de filhos Q_0 de tamanho N.

A combinação da população com seus filhos produz o conjunto $\boldsymbol{R}_t = \boldsymbol{\mathcal{P}}_t \cup \boldsymbol{Q}_t$. A população $\boldsymbol{R}$ possui $2N$ soluções. Nesta população $\boldsymbol{R}$, aplica-se o Algoritmo 4. O objetivo é gerar uma nova população de tamanho N. As N soluções são continuamente inseridas nas frentes de Pareto $\mathscr{F}_i$. Para realizar qualquer desempate, a população da última frente de Pareto $\mathscr{F}_i$ é ordenada de forma descendente utilizando o operador $\prec_n$. Assim, a população $\boldsymbol{\mathcal{P}}_{t+1}$ de tamanho N é submetida aos operadores de seleção, reprodução e mutação para gerar os filhos em $\boldsymbol{Q}_{t+1}$, de tamanho também N. De acordo com [Deb et al., 2002], a escolha dos pais para reprodução é realizada pela seleção por torneio adotando como critério o operador *crowding distance*, que utiliza a regra decisória já discutida. A Figura 29 sintetiza os procedimentos do algoritmo NSGA II.

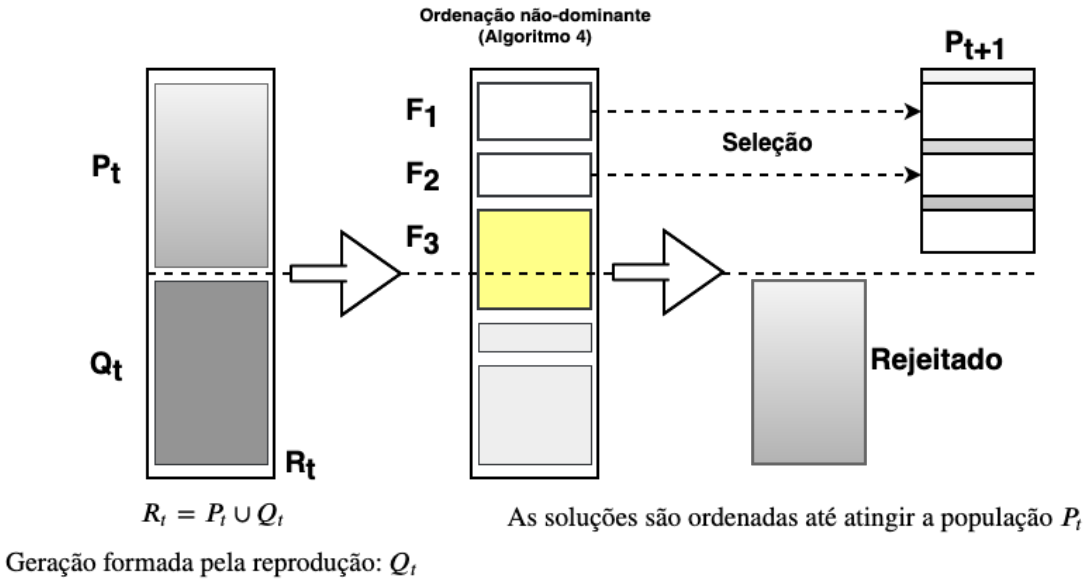

Figura 29 – Algoritmo NSGA II.

5.6- Sumário

O Capítulo 5 discutiu as técnicas de redução de dimensionalidade e apresentou a abordagem baseada em otimização natural multiobjetivo NSGA II. Como veremos mais adiante, o método *wrapper*, também discutido neste capítulo, determinará a configuração de uma arquitetura de seleção automática de características e parâmetros, na qual o NSGA II será fundamental para a otimização de objetivos que integram o processamento da detecção de anomalias.

6- Metodologia

Este Capítulo apresenta a metodologia desenvolvida para a identificação das anomalias nas turbinas eólicas. Apresentamos a abordagem de pré-processamento utilizada, e duas possibilidades para a etapa de processamento. Na primeira, utilizamos os métodos SVM, conforme apresentado no Capítulo 3. Já na segunda, utilizamos o método HMM, apresentado no Capítulo 4.

6.1- Introdução

A série temporal multivariada e multidimensional $\boldsymbol{TS}$ de dimensão $\mathscr{L} \times D$ possui registro de leituras a partir dos sensores instalados em diferentes componentes da turbina eólica. Devido ao fato de o SCADA operar de maneira intermitente sob a ocorrência de falhas, que são registradas no histórico do sistema, esta série teve sua sequência temporal interrompida diversas vezes, de acordo com o número de ocorrências. Assim, lidaremos com $\mathscr{V}$ séries temporais extraídas da série original de tamanho T, cujo valor será determinado pelo tempo que a operação seguiu normal no intervalo até a próxima falha, $\boldsymbol{TS} = \{\boldsymbol{TS}^{(1)}, \boldsymbol{TS}^{(2)}, \ldots, \boldsymbol{TS}^{(\mathscr{V})}\}$. Estas séries temporais servirão aos métodos de classificação abordados neste trabalho, cuja tarefa é a identificação das anomalias que poderão desenvolver futuras falhas nas turbinas eólicas.

Wei and Keogh [2006] discutem diferentes técnicas usualmente adotadas na manipulação dessas séries temporais durante a fase de pré-processamento. As subsequências representam fatias da série temporal cujo tamanho é determinado pelo valor k da janela deslizante. Seja a série temporal $\boldsymbol{TS}^{(j)} = \{TS_0^{(j)}, \ldots, TS_{T-1}^{(j)}\},\ TS_i^{(j)} \in \Re^D, \forall i = 0, \ldots, T-1,\ j = 1, \ldots, \mathscr{V}$, onde D é a dimensão de cada amostra e T é o número total de amostras [Zhao et al., 2016]. Os segmentos consecutivos da série temporal $\boldsymbol{x}_n^k = \{x_n, \ldots, x_{n+k-1}\}\ (0 \leq n \leq T-k,\ 1 \leq k \leq T)$ são denominados subsequências. Quando nos referirmos às subsequências, omitiremos o sobrescrito k. Considere, dessa forma, o conjunto de treinamento sob a forma $\{\boldsymbol{x}_n, y_n\}_{n=1}^{\mathcal{U}}$, onde $\mathcal{U}$ é o total de sub-

sequências formadas a partir das T amostras da série temporal $\boldsymbol{TS}^{(j)}$, adotando uma janela $k < T$. De acordo com Zhao et al. [2016], essas subsequências estão imersas em um gradiente que abrange três estados de saúde do sistema da turbina eólica:

- Normal: a operação da turbina eólica ocorre como esperado, havendo um risco mínimo de falhas, uma vez que as especificações de projeto do sistema são atendidos;
- Pré-falha: representa um estado de alerta, pois há falhas em evolução, mas a operação da turbina ainda atende as especificações do projeto;
- Paralisação forçada: indica o desligamento do sistema, quando ocorre paralisação na geração de energia pela turbina eólica.

A Figura 30 apresenta a evolução do estado de saúde do sistema da turbina eólica, posicionando as subsequências $\boldsymbol{x}_n^k$ ao longo da linha temporal, limitada entre a operação normal e a iminência da falha.

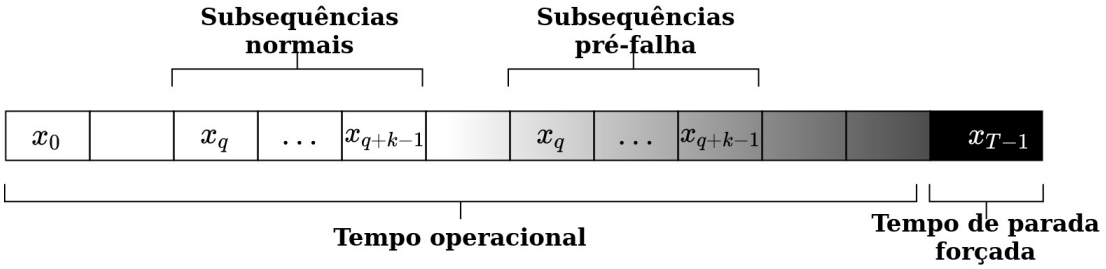

Figura 30 – Gradiente temporal do estado de saúde do sistema da turbina eólica. As subsequências estão associadas à evolução do estado de saúde desde a operação normal (subsequências normais) até o estado de pré-falha (subsequências pré-falha), na vizinhança da parada forçada, quando nenhum registro é mais obtido pelo sistema. Adaptado de [Zhao et al., 2016].

Nós assumimos que a série temporal $\boldsymbol{TS}$ não possui amostras contidas no espectro do estado de falha, no interior do lapso temporal da parada forçada, de acordo com a Figura 30. A construção das subsequências será essencial para o desenvolvimento dos dois métodos de aprendizado de máquina abordados neste trabalho. Contudo, a série temporal $\boldsymbol{TS}$ possui incongruências que demandaram a aplicação de tarefas de pré-processamento, que são discutidas na Seção 6.2.

6.2- Pré-Processamento

As tarefas de pré-processamento envolvem a manipulação do conjunto de dados, com a realização da limpeza de dados, remoção de ruídos e inconsistências, integração de dados de diferentes fontes, transformação de dados, redução de dados (técnicas de redução de dimensionalidade), entre outras. [García et al., 2015]. A Figura 31 resume as técnicas utilizadas.

Figura 31 – Etapas realizadas durante o pré-processamento.

Os métodos utilizados neste trabalho demandam diferentes conjuntos de técnicas de pré-processamento:

- Limpeza de dados: entre as 81 características relacionadas aos parâmetros do sistema da turbina eólica, a série temporal $\boldsymbol{TS}$ contabiliza 2 características com dados faltantes em uma quantidade que se aproxima do total de amostras $\mathscr{L}$. Com isso, optamos por remover estas duas características do processamento de ambos os métodos de aprendizado de máquina [García et al., 2015].

- Imputação de dados: outras duas características da série temporal $\boldsymbol{TS}$ apresentaram uma pequena quantidade de dados faltantes. Corrigimos este problema que interferiria no processamento de ambos os métodos aplicando uma técnica simples de imputação de dados, utilizando o valor médio da série nas posições onde há dados faltantes [García et al., 2015];

- Normalização de dados: valores brutos das características não são sempre boas opções para utilizarmos em modelos de aprendizado. A normalização equaliza a contribuição das diferentes características na construção do modelo, ao passo que características que possuem intervalo de valores maiores não dominam aquelas com intervalos menores [Singh and Singh, 2019]. Isso reduz o viés do modelo e otimiza o processamento. A normalização foi adotada previamente ao processamento das

máquinas de vetores de suporte;

- Balanceamento de classes: o excesso de amostras pertencentes a alguma classe provoca problemas de desbalanceamento. Durante o processamento do Fluxo de Trabalho 1, na classificação binária realizada pelo algoritmo convencional das máquinas de vetores de suporte, observamos a ocorrência deste problema, o que motivou a utilização da técnica SMOTE (*Synthetic Minority Over-sampling TEchnique*) para artificialmente produzir amostras daquela classe sub-representada a fim de balancear o conjunto de dados [Chawla et al., 2004];
- Seleção de características: a seleção de características ocorre em diferentes etapas nos métodos de aprendizado de máquina abordados. Enquanto a seleção de características é realizada sob o método *wrapper* no Fluxo de Trabalho 1, utilizamos uma metodologia baseada na maximização da verossimilhança para selecionar um subconjunto de características adequado para classificar os tipos de falha utilizando as cadeias ocultas de Markov no Fluxo de Trabalho 2.

A Seção 6.3 discute o método de detecção, diagnóstico e prognóstico de falhas em turbinas eólicas adotando as máquinas de vetores de suporte. O fluxo de trabalho proposto aborda os conceitos discutidos no Capítulo 3.

6.3- Fluxo de Trabalho 1

O método que vamos tratar nesta seção aborda dois tópicos que têm chamado atenção em trabalhos recentes na área de aprendizado de máquina: seleção automática de características e classificação semi-supervisionada. A seleção automática de características, nesta abordagem, é realizada pelo algoritmo genético multiobjetivo NSGA II, que, em respeito às funções objetivos, determina um conjunto de características e parâmetros de processamento. A classificação semi-supervisionada foi uma consequência da ausência de rótulos em nosso conjunto de dados, caracterizando um conjunto de dados imperfeito. Assim, nos baseamos no contexto do problema (a degradação do sistema como função temporal) para construir uma solução adotando as máquinas de vetores de suporte com a finalidade de detectar e classificar falhas em evolução nos

componentes.

Após executar as tarefas de pré-processamento (limpeza, imputação e normalização dos dados), as subsequências x_n^k são construídas. Adotamos uma janela deslizante fixa com $k = 36$ leituras, o que equivale ao período de 6 horas de observação, uma vez que as leituras pelo SCADA ocorrem a cada 10 minutos. Seja $\mathbf{A_0}$ a matriz de dimensão $36 \times \mathcal{F}'$ denotando o espaço de características da subsequência x_0 de acordo com a Figura 30, onde $\mathcal{F}'$ é o subconjunto de características selecionadas. Este espaço de características bruto é aumentado para um vetor de características que concatena: (1) a matriz $\mathbf{A_1}$; (2) entradas do triângulo superior da matriz de covariância da matriz $\mathbf{A_0}$; e (3) a matriz $\mathbf{A_0}$ normalizada. Como mostramos na Figura 30, a posição da subsequência na linha temporal expressa a sua suscetibilidade em agrupar amostras com anomalias. Dividimos essa linha temporal para a construção dos conjuntos de treinamento e testes. O intervalo de até 2 dias antes do evento de falha abrange as subsequências do conjunto de teste correspondente ao estado de pré-falha. O intervalo de 4 dias anteriores a este período abrange as subsequências que formam o conjunto de treinamento. Já as subsequências inseridas no intervalo de 5 dias anteriores ao período de treinamento formam o conjunto de teste correspondente ao estado normal do sistema da turbina eólica. A Figura 32 expressa este desenvolvimento.

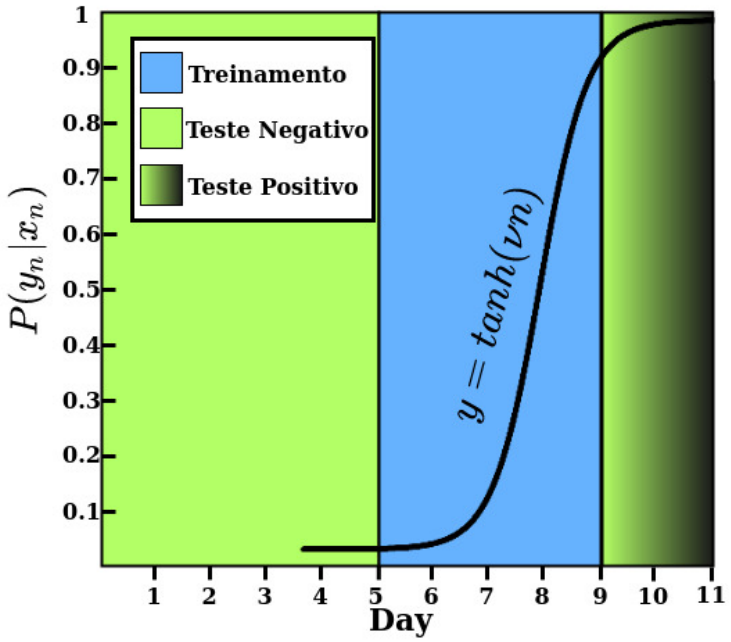

Figura 32 – Fatiamento temporal do conjunto de dados para determinar os conjuntos de treinamento e teste. Adaptado de [Zhao et al., 2016].

Como discutimos no Capítulo 3, a abordagem das máquinas de vetores de suporte com rótulos difusos processa um treinamento semi-supervisionado do modelo. Este processamento baseia-se na crença de que uma subsequência x_n pertence a uma classe. Dessa forma, definimos $y_n \in \Delta^L$, onde $\Delta^L = \{\boldsymbol{y} \in [0,1]^L | \sum_{j=1}^{L} y_j = 1\}$ e L é o número de classes. Como estamos lidando com classificação binária, $L = 2$ classes. Uma escala probabilística denotada por $\Delta^2 = \{u^+, u^-\}$ é adotada para expressar o grau de crença que uma subsequência está no estado de pré-falha ou no estado normal, respectivamente:

$$\begin{aligned} P(y_n = 1|\boldsymbol{x}_n) &= u_n^+, \text{ e} \\ P(y_n = -1|\boldsymbol{x}_n) &= u_n^-, \end{aligned} \tag{44}$$

onde $u_n^+ + u_n^- = 1$.

Utilizamos a função tangente hiperbólica $p_n = tanh(\nu n)$, onde $\nu > 0$ é o parâmetro de declividade da curva e n é o índice da subsequência, para determinar os valores entre 0 e 1 que configuram o grau de crença de pertencimento dessas subsequências a cada classe. Reescrevemos a Equação primal (1), na qual aplicamos os valores de u_n^+ e u_n^- sob este contexto [Zhao et al., 2016].

$$\begin{aligned} \text{Minimizar} \quad & \frac{1}{2}||\boldsymbol{w}||^2 + C\left(\sum_{i=1}^{\mathcal{U}}(\xi_i^+ u_i^+ + \xi_i^- u_i^-)\right) \\ \text{sujeito a:} \quad & \\ & \boldsymbol{w}^T\phi(\boldsymbol{x}_i) + b \geq 1 - \xi_i^+, \ i = 1, \ldots, \mathcal{U} \\ & -\boldsymbol{w}^T\phi(\boldsymbol{x}_i) - b \geq 1 - \xi_i^-, \ i = 1, \ldots, \mathcal{U} \\ & \xi_i^+, \xi_i^- \geq 0, \ i = 1, \ldots, \mathcal{U} \end{aligned} \tag{45}$$

onde $\phi(\cdot)$ é a função kernel utilizada, $\boldsymbol{w}$ e b são parâmetros do modelo, e a constante C é o custo de uso das variáveis de folga ξ_i^- e ξ_i^+.

Contudo, processaremos a classificação utilizando a formulação generalizada da Equação (45), desenvolvida na Equação (46) [Zhao et al., 2016].

$$\underset{\boldsymbol{w}}{\text{Minimizar}} \ \frac{1}{2}||\boldsymbol{w}||^p + C\sum_{c=1}^{2}\sum_{n=1}^{\mathcal{U}} u_{nc} E[f(\boldsymbol{x}_n), y_c] \tag{46}$$

onde $p > 0$ é a ordem de regularização e $P(y_n = y_c|\boldsymbol{x}_n)$, $y_c = (-1)^c$ é o rótulo difuso de cada amostra. A função perda E pode ser apresentada sob duas formas:

- Perda de articulação (do inglês *hinge loss*): $E[f(\boldsymbol{x}_n), y_c] = \max(0, 1 - y_c f(\boldsymbol{x}_n))$;
- Perda quadrada de articulação (do inglês *Squared hinge loss*): $E[f(\boldsymbol{x}_n), y_c] = \max(0, 1 - y_c f(\boldsymbol{x}_n))^2$.

No fluxo de trabalho proposto, o algoritmo NSGA II (assunto da Seção 5.5) opera na seleção de parâmetros de processamento e de características que serão processadas pelos algoritmos de aprendizado. Com base na notação utilizada em algoritmos genéticos, propusemos um cromossomo tripartite que expressa de forma binária essa seleção, como vemos na Figura 33.

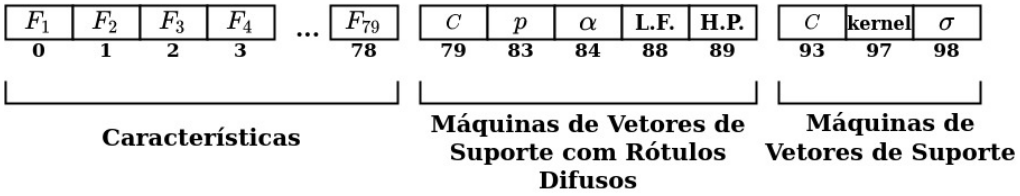

Figura 33 – Construção do cromossomo tripartite proposto neste trabalho para embarcar a seleção de características e parâmetros de processamento das classificações.

Em um configuração binária (0 ou 1), o cromossomo subdivide seus *bits* para formar os genes que determinam quais as características são selecionadas, os parâmetros da classificação usando máquinas de vetores de suporte com rótulos difusos e os parâmetros da classificação feita com o algoritmo convencional das máquinas de vetores de suporte. É importante ressaltar que a conversão de binário para decimal considerando os genes ocorre da seguinte forma: (1) foi construído um vetor $\boldsymbol{v}$ de valores decimais para cada parâmetro, respeitando os limites mínimo, máximo e o número q de elementos contidos neste vetor; (2) obtemos um índice a partir da conversão da representação binária do gene de q *bits* para a sua representação decimal: índice $= i_q \cdot 2^q + \ldots + i_1 \cdot 2^1 + i_0 \cdot 2^0$, onde i é o valor do *bit* no campo de índice q; (3) a partir do índice calculado no passo anterior, obtemos o valor do parâmetro usando $\boldsymbol{v}[\text{índice}]$.

Sob esta organização, o cromossomo fornece as informações necessárias para as duas etapas nas quais este método se divide. Primeiro, definimos o subconjunto de características $\mathcal{F}'$ selecionando aquelas características cujo respectivo índice i do gene ϱ_i possui valor do *bit* igual a 1. Em seguida, avaliamos os genes associados ao processamento da classificação utilizando as máquinas de vetores de suporte com rótulos difusos. Estes genes determinam parâmetros como a constante custo C, a ordem de regularização p, o parâmetro ν da declividade da curva da função tangente hiperbólica,

a função perda E (L.F.) e o horizonte de predição (H.P.). O horizonte de predição é o parâmetro relacionado ao prognóstico da falha, cujo intervalo abrange valores entre 12–48 horas, respeitando o conceito expresso na Figura 32. O objetivo da classificação realizada pelas máquinas de vetores de suporte com rótulos difusos é fornecer os rótulos de acordo com a Equação (6), considerando que os rótulos fornecidos no início do processamento eram difusos, pertencendo ao intervalo $[0, 1]$. A terceira parte do cromossomo fornece os genes que determinam os parâmetros da classificação binária das máquinas de vetores de suporte com margens suaves, como visto na Equação (67). Obtemos desses genes a constante custo C, a função kernel e o valor do desvio-padrão σ, aplicado quando a função kernel escolhida for a RBF. Este modelo treinado é avaliado utilizando as amostras de testes separadas durante o fatiamento das amostras, como mostrado na Figura 32.

O fluxo de trabalho descrito acima é inserido no método *wrapper*, no qual o algoritmo NSGA II atua em um ciclo fechado como a força motriz do processamento, selecionando iterativamente características e parâmetros enquanto novas gerações de indivíduos são criadas e aprimoradas. Este processamento se esgota quando um número máximo de gerações é alcançado. Ao fim, obtemos um conjunto de Pareto do qual extraímos o modelo de diagnóstico e prognóstico de falhas.

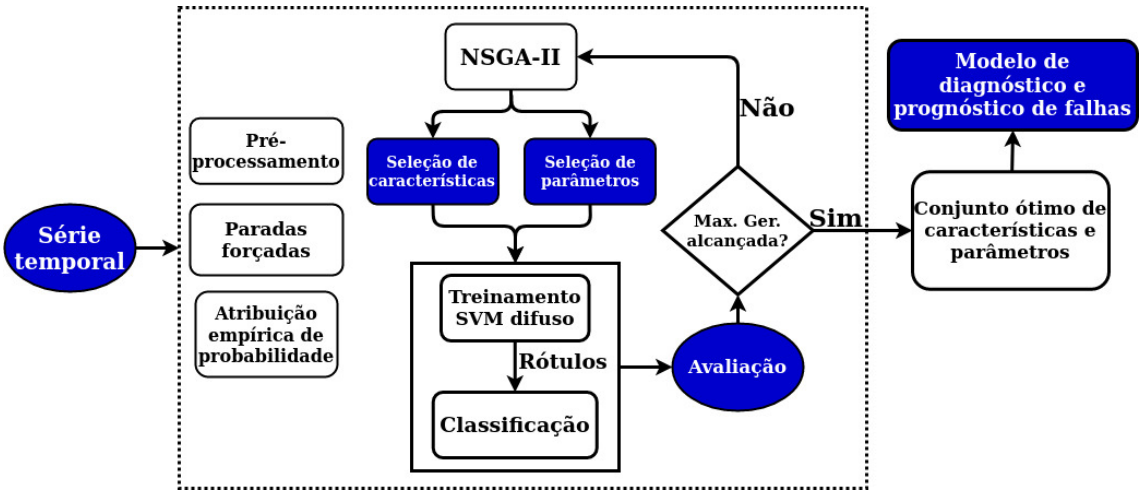

Figura 34 – Fluxo de trabalho do processamento proposto de seleção automática de características e parâmetros realizado pelo NSGA II operando de acordo com o método *wrapper*. Este conjunto de características e parâmetros é transmitido para os algoritmos de classificação que deverão identificar as anomalias do sistema. Este processamento fornece um conjunto de soluções que respeita os princípios do conjunto ótimo de Pareto. A partir dessas soluções, podemos extrair o modelo procurado.

A Seção 6.4 discute uma segunda abordagem de classificação de séries temporais, baseada nos modelos ocultos de Markov.

6.4- Fluxo de Trabalho 2

Kouadri et al. [2020] propõem um método de detecção e diagnóstico de causas potenciais de falhas em sistemas de turbinas eólicas baseado na classificação realizada pelos modelos ocultos de Markov. O método consiste em identificar os processos de pré-falha de diferentes componentes como os estados nos modelos ocultos de Markov, reservando um estado aos registros saudáveis do sistema. Assim, temos os seguintes estados e seu correspondente significado: (S_1 = operação normal, S_2 = pré-falha do componente 1, S_3 = pré-falha do componente 2, ..., S_N = pré-falha do componente N).

Utilizamos esse agrupamento de subsequências para construir o espaço de características para treinar e testar o modelo. Assim como fizeram os autores, também adotamos janelas contendo $k = 2.000$ amostras em cada subsequência. Essas 2000 amostras para construir as subsequências foram escolhidas da seguinte forma: (a) selecionamos sempre as 2000 últimas amostras das séries temporais $\boldsymbol{TS}^{(j)}$, onde $j = 2, \ldots, \mathscr{V}$; (b) as subsequências da operação normal do sistema foram construídas utilizando as 2000 primeiras amostras da série temporal $\boldsymbol{TS}^{(1)}$, correspondente ao início do histórico do registro, com início em 01/01/2016.

Formalizamos a construção desse conjunto de subsequências definindo a amostra $\boldsymbol{\tau}_i = \{\boldsymbol{x}^{\text{(op. normal)}}_{0:2000}, \boldsymbol{x}^{\text{(caixa de veloc.)}}_{(T-2000):T}, \boldsymbol{x}^{\text{(gerador)}}_{(T-2000):T}, \boldsymbol{x}^{\text{(rol. do gerador)}}_{(T-2000):T}, \boldsymbol{x}^{\text{(grupo hidráulico)}}_{(T-2000):T}, \boldsymbol{x}^{\text{(transformador)}}_{(T-2000):T}\}$, construída a partir do agrupamento de sucessivas subsequências relacionadas ao estado de operação normal e pré-falha de componentes do sistema.

Antes da seleção de características, cada subsequência compreende uma matriz $\boldsymbol{A}$ de dimensão 2.000×79. Contudo, estamos interessados na identificação da característica que melhor representa a evolução do estado de falha de cada componente. Assim, a matriz $\boldsymbol{A}$ de cada subsequência do conjunto $\boldsymbol{\tau}$ deverá, ao fim do processo de seleção de características, possuir dimensão 2000×5. É importante mencionar que a operação normal compreenderá o comportamento esperado dessas características. A Figura 35 ilustra uma amostra $\boldsymbol{\tau}_i$ destacando as séries formadas pelas características relevantes para cada tipo de falha.

Uma preocupação durante a adaptação do método ao conjunto de dados deste trabalho foi avaliar se a operação normal das 5 turbinas desenvolve um mesmo padrão de série temporal. Por isso, avaliamos, baseado nas 5 características relevantes para o

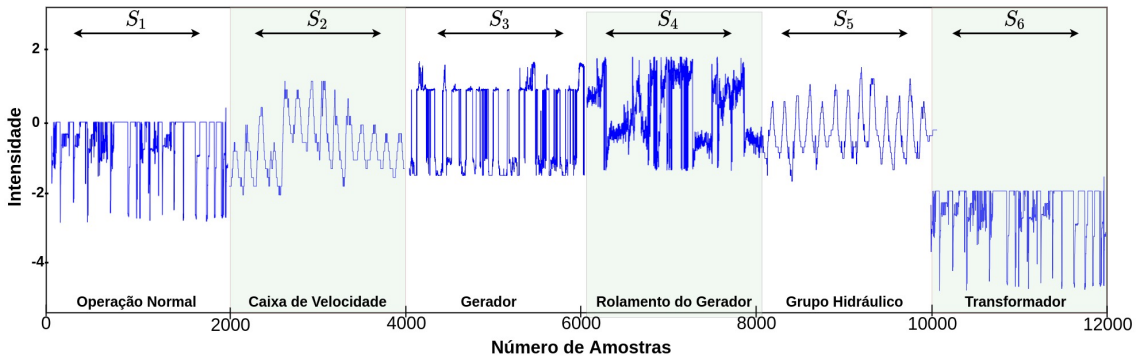

Figura 35 – Cada processo nesta representação contém destacada a característica que determina o estado de pré-falha do componente (exceto para a operação normal, que exige todas nesta situação). Por exemplo, o processo de falha do gerador, nesta figura, exibe somente a característica que marca o seu estado de deterioração, a despeito de a representação da subsequência ser multidimensional.

problema, se a operação normal das 5 turbinas possui alguma correlação. A Figura 36 é o resultado dessa avaliação, considerando uma das cinco características, utilizando o coeficiente de correlação de Pearson [Derrick et al., 1994]. Observamos, a partir desse resultado, que há uma elevada correlação entre todas as séries correspondentes à operação normal, não sendo relevante a turbina de origem.

Diferente do método da Seção 6.3, baseado na seleção automática de características utilizando o algoritmo genético multiobjetivo NSGA II, processamos a seleção de características numa etapa anterior e independente. Descrevemos abaixo os procedimentos realizados para formar o subconjunto $\mathcal{F}'$ a partir das 79 características:

- Iterativamente, todas as características são avaliadas;
- Um processo aleatório constrói amostras contendo somente a subsequência da operação normal e a subsequência que compreende o estado de pré-falha do componente avaliado. Este processo constrói amostras de treinamento e teste, sempre observando que subsequências contidas na amostra de treinamento não estão contidas na amostra de teste;
- O processamento armazena o cálculo do logaritmo da máxima verossimilhança (problema de avaliação dos modelos ocultos de Markov);
- As características que fornecerem o maior valor do logaritmo da máxima verossimilhança para cada componente em estado de pré-falha são selecionadas para compor o subconjunto $\mathcal{F}'$.

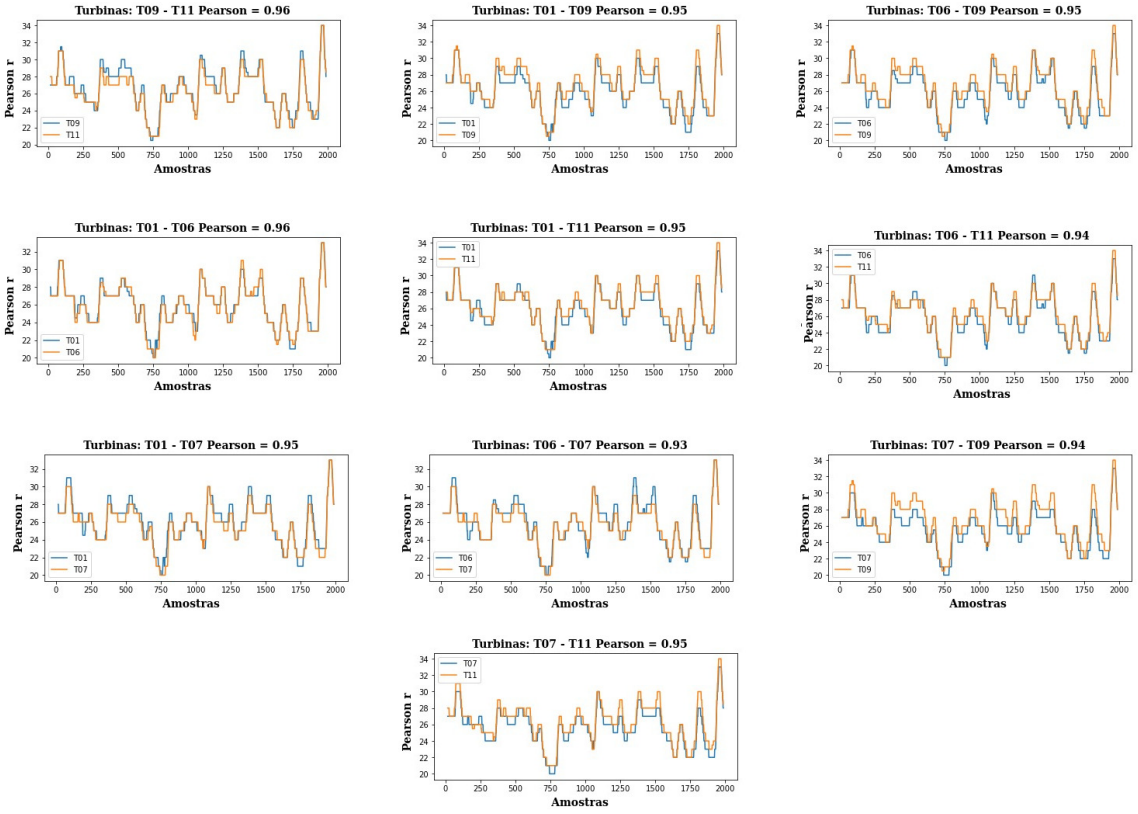

Figura 36 – Avaliação do coeficiente de correlação de uma única característica no processo de operação normal em diferentes turbinas eólicas. Observamos que as turbinas desenvolvem padrão de série temporal altamente correlacionado quando as amostras estão distantes do processo de pré-falha de algum componente.

As amostras τ_i são construídas a partir de todas as combinações possíveis das subsequências extraídas do conjunto de dados adquiridos pelo SCADA, considerando o histórico de falhas no período. É importante mencionar que subsequências contidas em amostras do conjunto de treinamento não estarão contidas em amostras do conjunto de teste.

Diferente do desenvolvimento apresentado por Kouadri et al. [2020], utilizamos dados reais adquiridos em baixa frequência pelo SCADA. Outra distinção entre a nossa abordagem e a apresentada pelos autores refere-se à metodologia de seleção de características adotada. Enquanto neste trabalho nos baseamos no resultado dos modelos ocultos de Markov, os autores aplicaram a extração de características utilizando análise de componentes principais (PCA). Em nosso resultados, avaliaremos os resultados a partir da análise da matriz de confusão, da qual extrairemos diferentes métricas do processamento da classificação multiclasse.

A Seção 6.5 apresenta as ferramentas computacionais utilizadas para o processa-

mento dos nossos resultados.

6.5- Ferramentas Utilizadas

Durante o desenvolvimento do trabalho foram utilizadas as seguintes ferramentas computacionais:

1. Linguagem de programação: Java v.1.8, R v.3.6.0, Python v.3.6.9 e MATLAB v.2017A;
2. Processamento do NSGA II: JMetal v.4.5.2 [Durillo and Nebro, 2011];
3. *Framework* de otimização convexa: CVX v.2.0 [Grant et al., 2009];
4. *Framework* de processamento dos modelos ocultos de Markov: hmmlearn v.0.2.4 [1].

6.6- Métricas de Desempenho

Diferentes índices foram utilizados para avaliar o problema de classificação da detecção de anomalias [Hossin and Sulaiman, 2015]. Os índices mais adequados para avaliar uma classificação binária não-balanceada foram utilizados [Gu et al., 2009]. Durante a análise do resultado da clusterização, são determinados os valores dos parâmetros verdadeiro positivo (TP), verdadeiro negativo (TN), falso positivo (FN) e falso negativo (FN). A partir destes valores, são obtidas métricas para avaliar o desempenho do processamento [Douzas et al., 2018].

A precisão acusa se o modelo é capaz de identificar corretamente as anomalias no conjunto de dados. Uma precisão elevada indica que durante a detecção de anomalias o modelo não provocou muitos falsos alarmes, ou identificou como anomalias observações normais. A Equação (47) apresenta o cálculo da precisão:

$$\text{Precisão} = \frac{TP}{TP + FP} \tag{47}$$

[1] Disponível em: https://github.com/hmmlearn/hmmlearn. Acessado em: 08/11/2020

A sensitividade avalia a relação entre a identificação correta da anomalia TP e os casos que seriam anomalias mas foram ignorados, portanto o parâmetro falso negativo FN. Um valor de sensitividade elevado indica que poucas observações anômalas foram classificadas como normais. A Equação (48) apresenta o cálculo da sensitividade:

$$\text{Sensitividade} = \frac{TP}{TP + FN} \tag{48}$$

O F-*score* calcula a média harmônica entre a precisão e a sensitividade. O seu cálculo permite avaliar o *tradeoff* entre as duas métricas. A Equação (49) apresenta o cálculo do F-*score*:

$$\text{F-}\textit{score} = \frac{2 * \text{Precisão} * \text{Sensitividade}}{\text{Precisão} + \text{Sensitividade}} \tag{49}$$

O caso de classificação multiclasse é abordado por Kautz et al. [2017]. Seja a matriz de confusão $k \times k$ apresentada na Tabela 1, onde cada elemento $c_{i,j}$ descreve o número de instâncias que foram preditas como classe i mas que pertence à classe j.

Tabela 1 – Matriz de confusão resultante de uma classificação multiclasse.

		Real		
Predição	$c_{1,1}$	$c_{1,2}$	$\cdots$	$c_{1,k}$
	$c_{2,1}$	$c_{2,2}$		$\vdots$
	$\vdots$		$\ddots$	
	$c_{k,1}$	$\cdots$		$c_{k,k}$

A partir da matriz de confusão, extraímos diferentes análises. O número de predições verdadeiras para cada classe m é:

$$TP_m = c_{m,m} \tag{50}$$

O número de falsos negativos para a classe m é:

$$FN_m = \sum_{i=1, i \neq m}^{k} c_{i,m} \tag{51}$$

O número de predições negativas verdadeiras de uma classe m é calculada como:

$$TN_m = \sum_{i=1, i \neq m}^{k} \sum_{j=1, j \neq m}^{k} c_{i,j} \tag{52}$$

Por fim, o número de falsos-positivos de uma classe m é dado por:

$$FP_m = \sum_{i=1, i \neq m}^{k} c_{m,i} \tag{53}$$

A partir dos valores TP_m, FP_m e FN_m é possível obter as métricas precisão e sensitividade para cada umas das k classes. Adotamos o cálculo do F-*score* macro para avaliar o *tradeoff* entre a precisão e a sensitividade. Seu cálculo consiste em primeiro calcular o F-*score* de acordo com a Equação (49) para cada uma das k classes e, então, dividir a soma desses valores por k.

6.7- Sumário

Neste capítulo, descrevemos inicialmente as etapas que consistem na construção de subsequências a partir da série temporal multivariada e o pré-processamento desse conjunto de dados. Em seguida, foram apresentadas as metodologias utilizadas para a construção dos dois fluxos de trabalho sobre o qual processamos os resultados adotando diferentes ferramentas computacionais. Por fim, discutimos as métricas de desempenho utilizadas na avaliação da eficiência dos fluxos de trabalho propostos.

7- Resultados

Apresentamos neste capítulo o resultado do processamento do nosso conjunto de dados utilizando a metodologia discutida.

7.1- Introdução

A empresa de serviços energéticos portuguesa EDP promove periodicamente desafios em sua plataforma de dados abertos [edp, b]. Quem apresentar a melhor solução, segundo alguns critérios fornecidos, é, por fim, remunerado pela EDP. Recentemente, a empresa disponibilizou os dados de 5 turbinas eólicas localizadas em uma de suas fazendas de energia eólica [edp, a]. Os dados correspondem a registros adquiridos nos anos de 2016 e 2017, capturados em intervalos de 10 minutos, formando a série temporal multivariada $\boldsymbol{TS}$, com a dimensão $\mathscr{L} \times D$ igual a 521.838×83, onde $\mathscr{L}$ é o número de amostras e D é o número de características. Os parâmetros monitorados estão organizados em 83 características $F_i \in \boldsymbol{\mathcal{F}}$, incluindo a identificação da turbina e o *timestamp*, que registra com precisão de segundos a data da observação $TS_i \in \boldsymbol{TS}$, com registros realizados no período 01/01/2016 – 29/11/2017. O conjunto de dados $\boldsymbol{\mathcal{D}}$ possui ainda o histórico de falhas, informações meteorológicas e arquivo de registros de eventos, como podemos ver na Figura 37.

Figura 37 – Descrição do conjunto de dados $\boldsymbol{\mathcal{D}}$ adquirido durante o monitoramento de turbinas eólicas.

Uma análise exploratória neste conjunto de dados permite identificar o gerador como o componente que mais falhou, de acordo com a Figura 38.

Também podemos ver que, entre as 5 turbinas, a turbina 06 (T06) foi a que

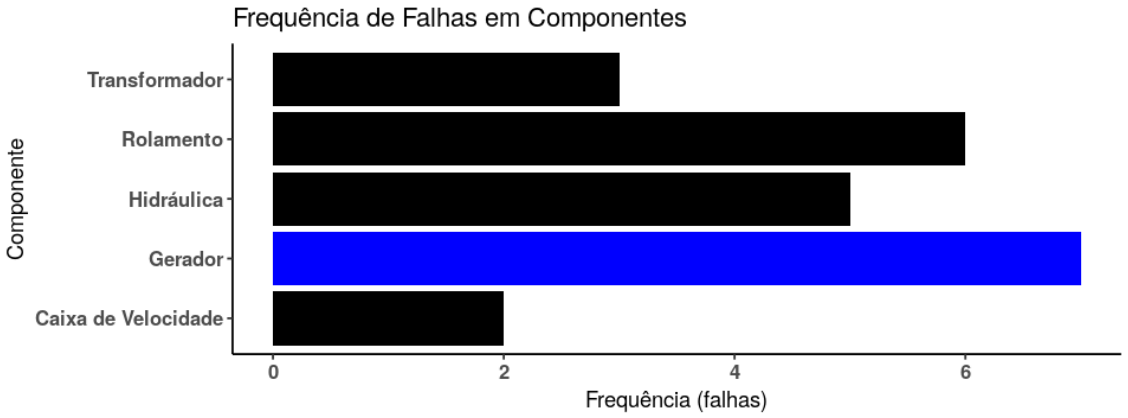

Figura 38 – Frequência de falhas em componentes considerando as 5 turbinas eólicas pertencentes à EDP.

apresentou maior frequência de falhas, como mostra a Figura 39.

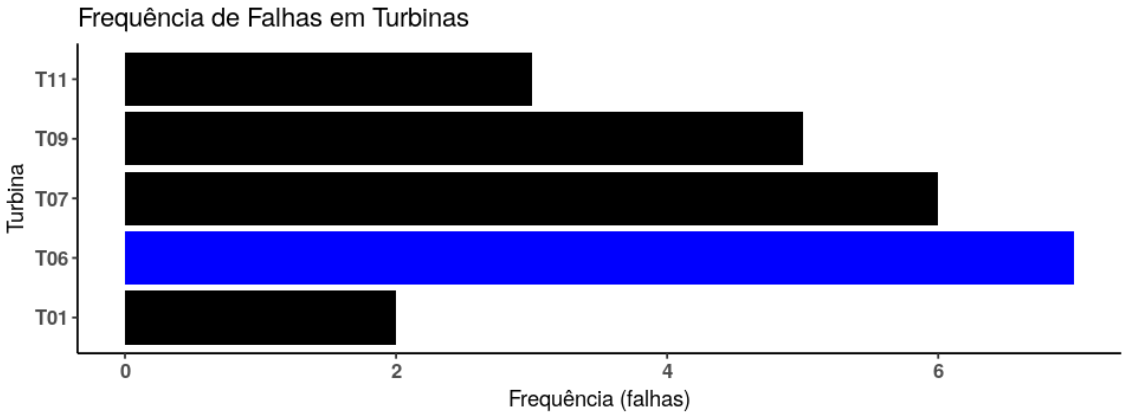

Figura 39 – Frequência de falhas em turbinas eólicas da EDP.

Os parâmetros monitorados abrangem os seguintes grupos de componentes:

- Gerador: compreende variáveis relacionadas à velocidade de rotação (mínima, média, máxima e desvio-padrão), temperatura média do estator e temperatura média do rolamento do gerador;
- Sistemas hidráulicos: variável com valores da temperatura média do óleo de lubrificação;
- Caixa de velocidade: compreende variáveis com valores da temperatura média do óleo de lubrificação e da temperatura média do rolamento;
- Rotor: variáveis relacionadas à velocidade de rotação do rotor (mínima, média, máxima e desvio-padrão);

- Informações meteorológicas: variáveis relacionadas ao monitoramento de condições da velocidade do vento (mínima, média, máxima, desvio-padrão), direções médias relativa e absoluta do vento e temperatura ambiente média;
- Nacele: variáveis da temperatura média e da direção da nacele;
- Geração de energia elétrica: compreende variáveis que monitoram a potência ativa (mínima, média, máxima e desvio-padrão), potência reativa (mínima, média, máxima e desvio-padrão), potência reativa indutiva (mínima, média, máxima e desvio-padrão), potência reativa capacitiva (mínima, média, máxima e desvio-padrão), potência gerada (mínima, média e máxima), temperatura em diferentes componentes, deslocamento médio de fase, frequências médias e tensões médias produzidas;
- Transformador: variáveis relacionadas à temperatura média de diferentes transformadores;
- Cone da nacele: variáveis relacionadas à temperatura média e posição angular ;
- Pás: variáveis relacionadas à posição angular (mínimo, médio, máximo e desvio-padrão)
- Controladores: variáveis relacionadas à temperatura média no topo do controlador da nacele, temperatura média no topo do controlador do eixo, temperatura média na placa VCP, temperatura média nas bobinas de estrangulamento e temperatura média no resfriador de água;
- Nariz do cone: variável relacionada à temperatura média no nariz do cone.

Este conjunto de dados é processado segundo as metodologias dos fluxos de trabalho discutidas no Capítulo 6. A Seção 7.2 apresenta a discussão dos resultados do Fluxo de Trabalho 1.

7.2- Fluxo de Trabalho 1

O objetivo do algoritmo genético multiobjetivo NSGA II é encontrar o conjunto ótimo de Pareto, contendo soluções que respeitam o princípio de não dominância. No problema formulado na Seção 6.3, este conjunto será formado por soluções que englobam a seleção de características e de parâmetros de processamento das classificações. Estas soluções são o resultado da otimização envolvendo dois objetivos:

- F_1: minimização da quantidade de características selecionadas para o subconjunto de características $\mathcal{F}'$;
- F_2: maximização do F-*score* calculado na classificação binária processada pelas máquinas de vetores de suporte utilizando o conjunto de teste.

Baseado no diagrama da Figura 34, o algoritmo NSGA II executou durante 500 gerações a existência de uma população de 100 cromossomos. Os operadores de recombinação e mutação promoveram o aumento da diversidade e aprimoramento dessa população. A atuação desses operadores ocorre segundo algum valor de probabilidade. O operador de recombinação pelo método de um ponto atuou com probabilidade de 0,9; já o operador de mutação baseada no processo de troca de *bits* atuou com probabilidade igual a 0,1.

Avaliamos diferentes componentes da turbina eólica para produzir os resultados. Começamos pelo transformador da turbina 01. De acordo com a Figura 38, o conjunto de dados registrou 3 eventos de falha nesse componente no período. As anotações do SCADA esclarecem sobre as causas dos eventos que provocaram a paralisação do sistema: elevação da temperatura do transformador, reparo do sistema de refrigeração e danos no sistema de ventilação.

A fronteira ótima de Pareto fornece as soluções apresentadas na Tabela 2. Note que 31 características contidas em $\mathcal{F}'$ oferecem F-*score* de 0,91, com precisão 1,0 e sensitividade de 0,83. Esta solução é capaz de predizer a falha com 31,2 horas de antecedência.

No contexto de análise de um conjunto ótimo de Pareto, nenhuma solução é integralmente melhor do que a outra. Através da análise dessas soluções, podemos selecionar aquela que oferece resultado superior para o F-*score*, o que significa um

Tabela 2 – Conjunto ótimo de Pareto das soluções do transformador da turbina 01.

Kernel	Função de Perda	Horizonte de Predição	N	F-score
Linear	**S.H.L**	**31,2**	**31**	**0,91**
RBF	S.H.L.	12,0	26	0,17

relaxamento em relação ao objetivo F_1. A Figura 40 apresenta o gráfico da fronteira ótima de Pareto, na qual podemos visualizar outras soluções não relevantes para os objetivos do problema.

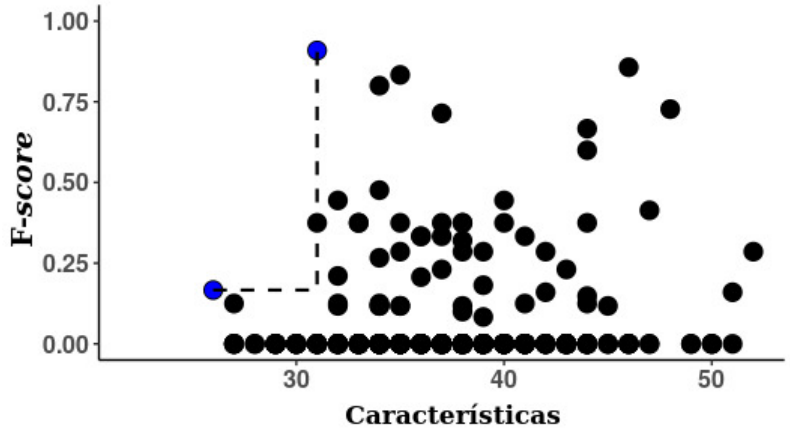

Figura 40 – Fronteira ótima de Pareto das soluções obtidas durante o processamento dos dados do transformador da turbina 01.

Procedemos para o resultado da caixa de velocidade da turbina 1, cujas soluções do conjunto ótimo de Pareto são apresentadas na Tabela 3. Este processamento abrange o registro de 2 eventos de falha, de acordo com a Figura 38. As causas da paralisação anotadas pelo SCADA são: bomba danificada e reparos na caixa de velocidade. Entre as soluções do conjunto ótimo de Pareto, destacamos a que fornece maior F-*score*. A solução com 39 características no subconjunto $\mathcal{F}'$ oferece F-*score* de 0,75, com precisão e sensitividade de 0,75, o que indica alguma dificuldade em identificar todas as anomalias considerando um horizonte de predição de 21,6 horas.

A Figura 41 apresenta graficamente a fronteira de Pareto que essas soluções desenham.

Os geradores ganharam atenção nos próximos resultados. Recorrendo novamente ao gráfico da Figura 38, observamos que é o componente com maior número de eventos de falhas, o que é um grande problema dado o elevado investimento demandado para a sua substituição e o extenso tempo de paralisação do sistema. Segundo os registro do SCADA, a turbina 06 apresentou falhas que paralisaram o sistema de maneira mais

Tabela 3 – Conjunto ótimo de Pareto das soluções da caixa de velocidade da turbina 01.

Kernel	Função de Perda	Horizonte de Predição	N	F-score
Linear	S.H.L	40,8	36	0,52
Linear	S.H.L.	40,8	37	0,64
Linear	**S.H.L.**	**21,6**	**39**	**0,75**
RBF	S.H.L.	36,0	25	0
Linear	S.H.L.	19,2	34	0,44
Linear	S.H.L.	21,6	29	0,17
RBF	H.L.	28,8	30	0,33

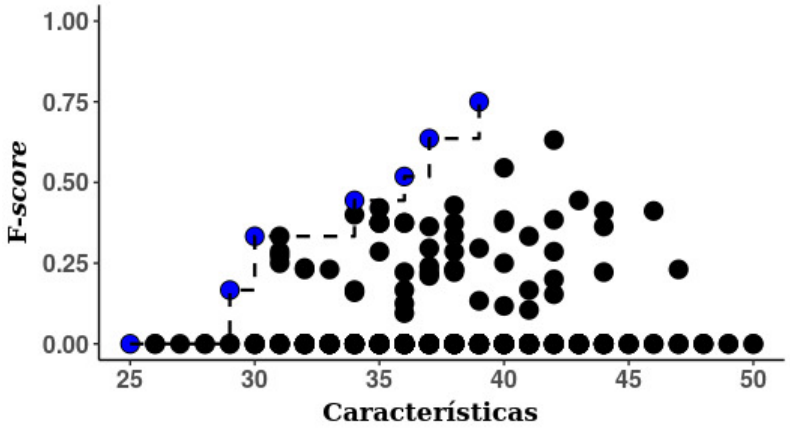

Figura 41 – Fronteira ótima de Pareto das soluções obtidas durante o processamento dos dados da caixa de velocidade da turbina 01.

recorrente, com 5 eventos: substituição do gerador, erro do sensor de temperatura, elevada temperatura do gerador e outra substituição do gerador. Decidimos não considerar a falha envolvendo o sensor de temperatura no processamento. A Tabela 4 apresenta as soluções do conjunto ótimo de Pareto do processamento envolvendo o gerador da turbina 6. A classificação alcança F-*score* máximo, com precisão e sensitividade 1,0, selecionando 46 características para o subconjunto $\mathcal{F}'$. Note que essa solução adota um horizonte de predição de 12 horas.

Tabela 4 – Conjunto ótimo de Pareto das soluções do gerador da turbina 06.

Kernel	Função de Perda	Horizonte de Predição	N	F-score
Linear	S.H.L.	31,2	40	0,43
RBF	S.H.L.	38,4	36	0,33
Linear	H.L.	36,0	42	0,6
RBF	S.H.L.	33,6	35	0,29
RBF	S.H.L.	26,4	31	0,23
Linear	**H.L.**	**12,0**	**46,0**	**1,0**

A Figura 42 apresenta a visualização gráfica da fronteira ótima de Pareto determi-

nada durante esse processamento.

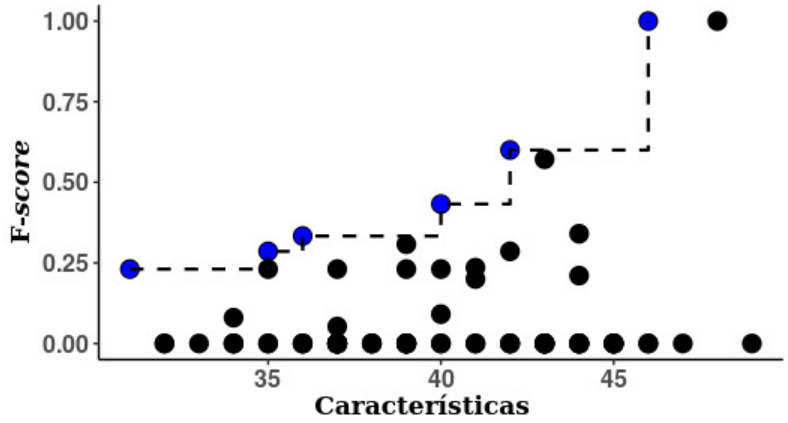

Figura 42 – Fronteira ótima de Pareto das soluções obtidas durante o processamento dos dados do gerador da turbina 06.

O gerador da turbina 7 apresentou um único registro de evento de falha, consistindo em um dano não detalhado. O processamento envolvendo seus dados alcançou também resultado máximo para o F-*score*, com precisão e sensitividade $1,0$, considerando a adoção de 32 características no subconjunto $\mathcal{F}'$ e horizonte de predição de 24 horas, valor comum em todas as soluções do conjunto.

Tabela 5 – Conjunto ótimo de Pareto das soluções do gerador da turbina 07.

Kernel	Função de Perda	Horizonte de Predição	N	F-score
Linear	S.H.L.	24,0	24	0,22
Linear	S.H.L.	24,0	25	0,33
Linear	H.L.	24,0	23	0
Linear	**H.L.**	**24,0**	**32**	**1,0**
Linear	H.L.	24,0	28	0,89

A Figura 43 apresenta graficamente a fronteira ótima de Pareto para esse processamento.

Por fim, analisamos o gerador da turbina 11, que registrou também um único evento, relacionado a problemas no circuito elétrico. A Tabela 6 apresenta o conjunto ótimo de Pareto, do qual extraímos a solução que fornece F-*score* de 0,8, com precisão 0,86 e sensitividade 0,75, considerando 37 características no subconjunto $\mathcal{F}'$ e horizonte de predição de 48 horas. Diferente das outras turbinas, o método neste processamento não obteve o valor máximo da métrica.

A Figura 44 apresenta a fronteira ótima de Pareto do processamento.

Concluída a análise exploratória, podemos comparar o método desenvolvido neste

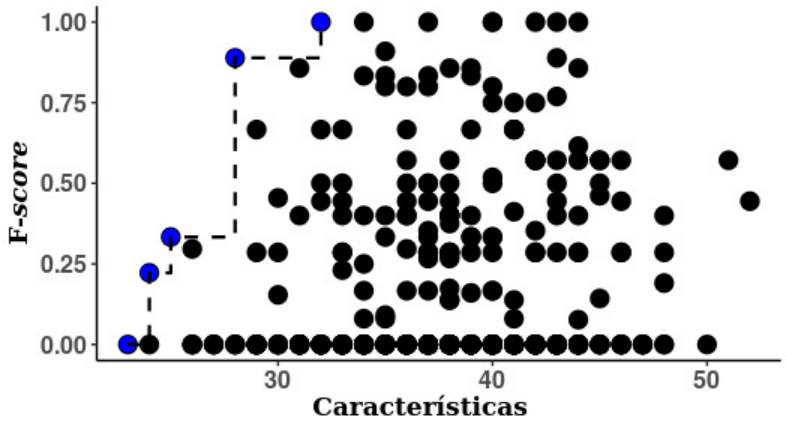

Figura 43 – Fronteira ótima de Pareto das soluções obtidas durante o processamento dos dados do gerador da turbina 07.

Tabela 6 – Conjunto ótimo de Pareto das soluções do gerador da turbina 11.

Kernel	Função de Perda	Horizonte de Predição	N	F-score
Linear	S.H.L.	43,2	30	0,5
Linear	**H.L.**	**48,0**	**37**	**0,8**
Linear	H.L.	43,2	34	0,67
Linear	H.L.	43,2	32	0,54
RBF	S.H.L.	40,8	27	0

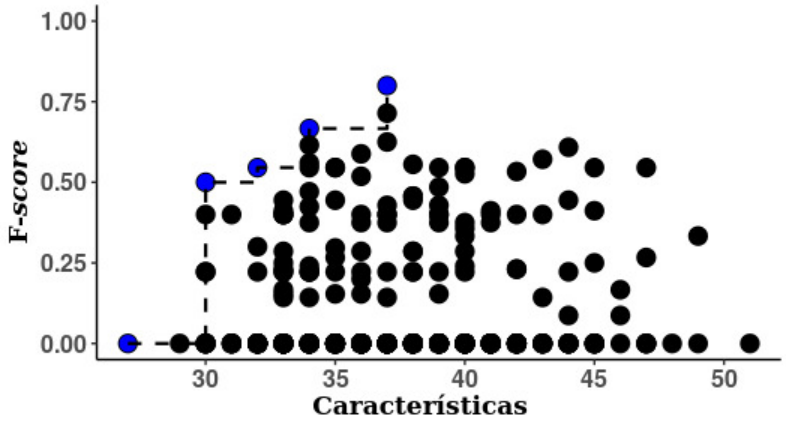

Figura 44 – Fronteira ótima de Pareto das soluções obtidas durante o processamento dos dados do gerador da turbina 11.

trabalho com o apresentado pelos autores Al Iqbal et al. [2018], como mostra a Tabela 7 . Para isso, convertemos a métrica dos nossos resultados para a mesma métrica adotada pelo trabalho de referência, AUC (do inglês *Area Under Curve*) [Wu and Flach, 2005]. O método de aprendizado de ponta a ponta desenvolvido neste trabalho mostrou-se mais eficiente na detecção de anomalias do processo, com menor erro de classificação, sendo capaz de realizar prognóstico de falhas entre 12 – 48 horas antes da sua ocorrência.

Estes modelos poderiam posteriormente ser transplantados para integrar um sistema de detecção de anomalias, diagnóstico e prognóstico, o que resultaria na redução de custos e menor tempo ocioso devido às atividades de manutenção reativa.

Tabela 7 – Comparação entre abordagens.

Turbina	**Componente**	**Este trabalho**		Al Iqbal et al. [2018]
		AUC	**AUC + desv. padrão**	**AUC + desv. padrão**
06	Gerador	1,0	0,95 + 0,087	0,841 + 0,15
07		1,0		
11		0,85		
01	Transformador	0,917	0,883 + 0,047	
	Caixa de Veloc.	0,85		

A Seção 7.3 apresenta os resultados de detecção e diagnóstico obtidos pelo processamento dos modelos ocultos de Markov.

7.3- Fluxo de Trabalho 2

Nesta seção discutiremos os resultados do processamento envolvendo os modelos ocultos de Markov na realização da detecção e diagnóstico de anomalias, cujo método foi abordado na Seção 6.4. Iniciamos a discussão pela seleção de características, que compreendeu uma etapa anterior e independente do processamento dos demais resultados. A Tabela 8 apresenta as características selecionadas.

Tabela 8 – Características selecionadas.

Componente	**Característica**
Caixa de Velocidade	Temperatura média do ambiente [℃]
Gerador	Menor valor médio de rotação do gerador [rpm] no período
Rolamento do Gerador	Direção absoluta média do vento [m/s]
Grupo Hidráulico	Temperatura média no centro do controlador [℃]
Transformador	Potência ativa do gerador [Wh]

Notamos que algumas características da Tabela 8 possuem de fato uma relação intuitiva com o componente que representará na construção da amostra τ_i, enquanto outras não gozam essa propriedade.

Baseado no histórico de eventos de falhas em componentes apresentado na

Figura 38, selecionamos aleatoriamente numa proporção de 70% e 30%, respectivamente, aqueles registros que pertenceriam aos conjuntos de treinamento e teste. Esta divisão resultou em um conjunto de teste contendo 8 amostras τ_i, que serão avaliadas posteriormente. Construímos as amostras τ_i a partir da combinação das subsequências contidas em cada conjunto. Essa combinação respeitou a arquitetura apresentada na Tabela 9, que relaciona os estados S_i aos processos e à dimensão das subsequências nos conjuntos de treinamento e teste.

Tabela 9 – Descrição das amostras τ_i utilizadas para o treinamento e teste.

Estado	**Processo**	**Dados de Treinamento**	**Dados de Teste**
S_1	Operação normal	$\Re^{2000\times5}$	$\Re^{2000\times5}$
S_2	Falha da caixa de velocidade	$\Re^{2000\times5}$	$\Re^{2000\times5}$
S_3	Falha do gerador	$\Re^{2000\times5}$	$\Re^{2000\times5}$
S_4	Falha do rolamento do gerador	$\Re^{2000\times5}$	$\Re^{2000\times5}$
S_5	Falha do grupo hidráulico	$\Re^{2000\times5}$	$\Re^{2000\times5}$
S_6	Falha do transformador	$\Re^{2000\times5}$	$\Re^{2000\times5}$

O processamento envolveu os problemas de aprendizado, avaliação e decodificação dos modelos ocultos de Markov, que foram processadas pelo *framework* hmmlearn[1] a partir de um conjunto de parâmetros iniciais:

- Número de estados: o processamento envolve 6 estados, relacionados aos processos que serão classificados, de acordo com a Tabela 9;
- Tipo de matriz de covariância: o hmmlearn estima que os estados compreendem distribuições gaussianas multivariadas. Selecionamos a opção "*diag*", pela qual cada estado utiliza somente a diagonal de uma matriz de covariância;
- Parâmetros atualizados: um conjunto de letras controla quais parâmetros serão atualizados ao longo do processamento. Definimos o conjunto "*cmt*", segundo o qual as matrizes de covariância, os valores das médias de cada estado e a matriz de transição terão seus valores atualizados;
- Algoritmo de decodificação: definimos o uso do algoritmo de Viterbi.

As matrizes de transição foram inicializadas segundo o modelo esquerda-direita. Segundo este modelo, o estado ou permanece no estado atual ou transita para o próximo estado maior, sem poder transitar para um estado inferior ou saltar para um estado

[1] Disponível em: https://github.com/hmmlearn/hmmlearn. Acessado em: 08/11/2020

que não seja o imediatamente superior. A Tabela 10 apresenta a matriz de transição que utilizamos na inicialização do problema, que devido à parametrização do *framework* hmmlearn sofreu atualizações durante o processamento.

Tabela 10 – Exemplo de uma matriz de transição segundo o método esquerda-direita para definição das probabilidades de transição entre os estados.

	S_1	S_2	S_3	S_4	S_5	S_6
S_1	**0,5**	**0,5**	0	0	0	0
S_2	0	**0,5**	**0,5**	0	0	0
S_3	0	0	**0,5**	**0,5**	0	0
S_4	0	0	0	**0,5**	**0,5**	0
S_5	0	0	0	0	**0,5**	**0,5**
S_6	0	0	0	0	0	**1**

Procedemos para os resultados da classificação, considerando as amostras do conjunto de teste. A amostra τ_1 forneceu o F-*score* macro de $0,85$. Esta métrica foi extraída da matriz de confusão apresentada na Tabela 11, na qual observamos que o desempenho da classificação é excelente em relação a alguns processos. O método foi capaz de classificar sem erros as amostras do processo de pré-falha da caixa de velocidade (S_2) e do transformador (S_6). Alguma dificuldade foi encontrada nas amostras do gerador (S_3) e do rolamento do gerador (S_4), pois possuem erros de classificação maiores em comparação com os demais estados.

Tabela 11 – Resultado do processamento da classificação da amostra de teste τ_1.

	S_1	S_2	S_3	S_4	S_5	S_6	**Sensitividade**
S_1	**1938**	**62**	0	0	0	0	0,97
S_2	0	**2000**	0	0	0	0	1
S_3	0	**705**	**1295**	0	0	0	0,65
S_4	0	0	0	**1076**	**924**	0	0,54
S_5	0	0	0	0	**1972**	**28**	0,99
S_6	0	0	0	0	0	**2000**	1
Precisão	1	0,72	1	1	0,68	0,99	

A amostra τ_2 forneceu F-*score* macro de $0,89$, a qual também extraímos da matriz de confusão, apresentada na Tabela 12. Neste processamento, todas as amostras da operação normal (S_1) foram corretamente classificadas, bem como as amostras do processo de pré-falha do gerador (S_3). Notamos que o grupo hidráulico (S_5) forneceu para esta amostra o maior erro de classificação.

A Tabela 13 apresenta os resultados da métrica F-*score* obtidos para as demais amostras τ_i do conjunto de teste.

Tabela 12 – Resultado do processamento da classificação da amostra de teste τ_2.

	S_1	S_2	S_3	S_4	S_5	S_6	**Sensitividade**
S_1	**2000**	**0**	0	0	0	0	1
S_2	0	**1954**	**46**	0	0	0	0,98
S_3	0	**0**	**2000**	0	0	0	1
S_4	0	0	0	**2000**	**0**	0	1
S_5	0	0	0	**861**	**1139**	0	0,57
S_6	0	0	0	0	**330**	**1670**	0,84
Precisão	1	1	0,98	0,70	0,78	1	

Tabela 13 – Resultado do processamento dos modelos ocultos de Markov obtido para as demais amostras do conjunto de teste.

Amostra	**F-*score***
τ_3	0,80
τ_4	0,78
τ_5	0,78
τ_6	0,87
τ_7	0,80
τ_8	0,84

Em comparação com a nossa referência [Kouadri et al., 2020], obtemos métricas compatíveis com o apresentado pelos autores, considerando que nossos resultados baseiam-se em dados reais, o que implica desafios não tratados pelos autores. Vale ressaltar que o processamento possui elevado custo computacional, o que provocou a abdicação de um melhor resultado possível.

8- Conclusão

Em um cenário de marcante transição energética para a era pós-petróleo, a expansão do uso da energia eólica é um fato. Muito dessa realidade se deve ao nível de maturidade das tecnologias da energia eólica frente a outras fontes alternativas de energia. No cerne dessa revolução energética, as turbinas eólicas desempenham o protagonismo, uma vez que seu sistema é responsável pela transformação da energia mecânica do vento em energia elétrica. Contudo, as turbinas eólicas são sistemas complexos e caros que consistem de diferentes subsistemas interdependentes, cuja exposição a condições operacionais extremas, submetidas a toda sorte de eventos climáticos, exigem, além de resiliência estrutural, recursos que atestem a garantia de sua segurança operacional. Neste trabalho, abordamos a execução dessa tarefa pelo sistema de monitoramento e supervisão SCADA. Com isso, o processamento bem sucedido de técnicas de detecção de anomalias na operação da turbina eólica baseada no SCADA converge para o alcance dos objetivos do sistema: ser eficiente e viável economicamente.

O conjunto de dados ao qual nos referimos neste trabalho é resultado do monitoramento realizado pelo SCADA durante a operação de 5 turbinas eólicas no período de 2016 – 2017. Esse conjunto de dados reais possui elevada dimensionalidade, consistindo em um conjunto de dados imperfeito, dada a ausência de rótulo em cada amostra.

Pelas características do conjunto de dados, identificamos a classificação semi-supervisionada como capaz de atender os critérios do processamento. Em uma arquitetura adotando o método *wrapper*, o algoritmo de otimização multiobjetivo NSGA II seleciona características e parâmetros de processamento dos algoritmos de aprendizado. Depois disso, a classificação semi-supervisionada realizada pelas máquinas de vetores de suporte com rótulos difusos atribuiu rótulos binários aos dados. Uma segunda classificação processada, utilizando o algoritmo convencional das máquinas de vetores de suporte, forneceu a avaliação do modelo. Diferentes componentes do sistema da turbina eólica integraram o processamento dos resultados, que se mostraram superiores aos obtidos pela referência, com prognóstico de falha no horizonte de 12 – 48 horas. Também destacamos que alguns resultados obtiveram valor máximo na métrica de avaliação do processamento para a detecção de anomalias no componente. Assim, nossa abordagem

ponta-a-ponta mostrou-se eficiente no tratamento de problemas de detecção de anomalias apesar das limitações inicialmente impostas pelo conjunto de dados disponível.

Uma segunda abordagem, baseada nos modelos ocultos de Markov, realizou o processamento classificação multiclasse de componentes em processos de pré-falha e da operação normal do sistema, associando-os aos estados ocultos do modelo. Portanto, a decodificação dos estados utilizando o algoritmo de Viterbi definiu a qual dessas classes cada amostra pertence. O resultado abrangeu 8 amostras de testes que alcançaram bons resultados na média, obtendo F-*score* máximo de 0,89 no processamento da amostra τ_2. Em relação à referência, nossa abordagem lidou com dados reais e adotou uma metodologia baseada na maximização de verossimilhança para selecionar as características relevantes. Com essa ponderação, avaliamos que o resultado alcançou os objetivos ao classificar, muitas vezes com máxima precisão, as amostras à sua respectiva classe, além de estender a aplicação o método original para dados reais.

8.1- Artigos Publicados

Além das contribuições mencionadas na seção anterior, a abordagem utilizando a técnica SVM apresentada neste trabalho gerou o seguinte artigo:

- IWSSIP 2020 - “Wind Turbine Fault Detection: A Semi-Supervised Learning Approach With Automatic Evolutionary Feature Selection”

Um outro artigo está em avaliação na revista “International Journal of Innovative Computing and Applications”, sendo uma versão em que a metodologia apresentada no artigo anterior é mais detalhada e aplicada em diferentes componentes da turbina eólica.

Por fim, a abordagem utilizando o HMM apresentou resultados interessantes e deverá ser foco para o desenvolvimento de um novo artigo, onde compararemos as duas abordagens apresentadas.

8.2- Trabalhos Futuros

Futuros trabalhos se beneficiarão da curva de aprendizado trilhada durante o desenvolvimento das duas abordagens desenvolvidas. A premissa do trabalho em desenvolvimento consiste no uso dos modelos ocultos de Markov com espaço de estados infinito em associação com redes neurais recorrentes. O objetivo deste fluxo de trabalho será a realização de prognóstico de falhas com elevada precisão e tratamento do problema no processamento em tempo real. Uma outra abordagem consiste na avaliação de cada componente dos fluxos de trabalhos desenvolvidos, sendo proposto a utilização de diferentes técnicas em cada etapa. Neste contexto, novas técnicas evolutivas poderão ser estudadas e comparadas com o NSGA II, como exemplo, citamos os Algoritmos Evolucionários [Back, 1996; Jirapech-Umpai and Aitken, 2005; Tan et al., 2014] e os Algoritmos de Chaves Aleatórias Viciadas (BRKGA) [Gonçalves and Resende, 2011; Toso and Resende, 2015].

Por fim, acreditamos que outros trabalhos poderão ser desenvolvidos no mesmo sentido, buscando um modelo generalista porém eficiente na detecção de anomalias em diferentes componentes.

Referências Bibliográficas

EDP Fazendas Eólicas. https://www.edprnorthamerica.com/farms/wind-farms. Acessado: 2019-08-10.

EDP Open Data. https://opendata.edp.com/pages/homepage/. Accessed: 2020-02-17.

Anomaly detection of energy consumption in buildings: A review, current trends and new perspectives.

Adra, S. F. and Fleming, P. J. (2010). Diversity management in evolutionary many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 15(2):183–195.

Al Iqbal, M. R., Zhao, R., Ji, Q., and Bennett, K. P. (2018). A generalized method for fault detection and diagnosis in scada sensor data via classification with uncertain labels. In *International Conference on Data Science ICDATA*, volume 18.

Alelyani, S., Tang, J., and Liu, H. (2018). Feature selection for clustering: A review. In *Data Clustering*, pages 29–60. Chapman and Hall/CRC.

Almalawi, A., Fahad, A., Tari, Z., Alamri, A., AlGhamdi, R., and Zomaya, A. Y. (2015). An efficient data-driven clustering technique to detect attacks in scada systems. *IEEE Transactions on Information Forensics and Security*, 11(5):893–906.

Amirat, Y., Benbouzid, M. E. H., Al-Ahmar, E., Bensaker, B., and Turri, S. (2009). A brief status on condition monitoring and fault diagnosis in wind energy conversion systems. *Renewable and sustainable energy reviews*, 13(9):2629–2636.

Arghira, N., Hossu, D., Fagarasan, I., Iliescu, S. S., and Costianu, D. R. (2011). Modern scada philosophy in power system operation-a survey. *University"Politehnica"of Bucharest Scientific Bulletin, Series C: Electrical Engineering*, 73(2):153–166.

Ayad, A. (2009). *Maximum Power Point Tracking for Wind Energy Conversion System Using Fuzzy Modeling and Control*. PhD thesis.

Back, T. (1996). *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford university press.

Baker, T., Asim, M., MacDermott, Á., Iqbal, F., Kamoun, F., Shah, B., Alfandi, O., and Hammoudeh, M. (2020). A secure fog-based platform for scada-based iot critical infrastructure. *Software: Practice and Experience*, 50(5):503–518.

Bangalore, P. and Tjernberg, L. B. (2015). An artificial neural network approach for early fault detection of gearbox bearings. *IEEE Transactions on Smart Grid*, 6(2):980–987.

Basseville, M., Nikiforov, I. V., et al. (1993). *Detection of abrupt changes: theory and application*, volume 104. prentice Hall Englewood Cliffs.

Baum, L. E. and Petrie, T. (1966). Statistical inference for probabilistic functions of finite state markov chains. *The annals of mathematical statistics*, 37(6):1554–1563.

Bilmes, J. A. et al. (1998). A gentle tutorial of the EM algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. *International Computer Science Institute*, 4(510):126.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.

Byrne, G., Dornfeld, D., Inasaki, I., Ketteler, G., König, W., and Teti, R. (1995). Tool condition monitoring (tcm)—the status of research and industrial application. *CIRP annals*, 44(2):541–567.

Canizo, M., Onieva, E., Conde, A., Charramendieta, S., and Trujillo, S. (2017). Real-time predictive maintenance for wind turbines using big data frameworks. In *2017 IEEE International Conference on Prognostics and Health Management (ICPHM)*, pages 70–77. IEEE.

Cassandras, C. G. and Lafortune, S. (2009). *Introduction to discrete event systems*. Springer Science & Business Media.

Çelik, M., Dadaşer-Çelik, F., and Dokuz, A. Ş. (2011). Anomaly detection in temperature data using dbscan algorithm. In *2011 International Symposium on Innovations in Intelligent Systems and Applications*, pages 91–95. IEEE.

Chandola, V., Banerjee, A., and Kumar, V. (2009). Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):15.

Chawla, N. V., Japkowicz, N., and Kotcz, A. (2004). Special issue on learning from imbalanced data sets. *ACM SIGKDD explorations newsletter*, 6(1):1–6.

Chen, L., Xu, G., Zhang, Q., and Zhang, X. (2019). Learning deep representation of imbalanced scada data for fault detection of wind turbines. *Measurement*, 139:370–379.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3):273–297.

Council, G.-G. W. E. (2019). Global wind report 2018, 2019. *URL: www. gwec. net.(Cited on page 5.)*.

Daneels, A. and Salter, W. (1999). What is SCADA?

Dash, M. and Liu, H. (1997). Feature selection for classification. *Intelligent data analysis*, 1(1-4):131–156.

Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms*, volume 16. John Wiley & Sons.

Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2):182–197.

Derrick, T., Bates, B., and Dufek, J. (1994). Evaluation of time-series data sets using the pearson product-moment correlation coefficient. *Medicine and science in sports and exercise*, 26(7):919—928.

Douzas, G., Bacao, F., and Last, F. (2018). Improving imbalanced learning through a heuristic oversampling method based on k-means and smote. *Information Sciences*, 465:1–20.

Durillo, J. J. and Nebro, A. J. (2011). jMetal: A Java Framework for Multi-objective Optimization. *Adv. Eng. Softw.*, 42(10):760–771.

E. Goldberg, D. and Henry Holland, J. (1988). Genetic algorithms and machine learning. *Machine Learning*, 3.

Echevarría, L. C., Santiago, O. L., de Campos Velho, H. F., and da Silva Neto, A. J. (2019). *Fault Diagnosis Inverse Problems: Solution with Metaheuristics*. Springer.

Edelkamp, S. and Schroedl, S. (2011). *Heuristic search: theory and applications*. Elsevier.

Emary, E., Zawbaa, H. M., and Hassanien, A. E. (2016). Binary grey wolf optimization approaches for feature selection. *Neurocomputing*, 172:371–381.

Gao, J., Liu, J., Rajan, B., Nori, R., Fu, B., Xiao, Y., Liang, W., and Philip Chen, C. (2014). Scada communication and security issues. *Security and Communication Networks*, 7(1):175–194.

García, S., Luengo, J., and Herrera, F. (2015). *Data preprocessing in data mining*. Springer.

Gather, U. and Rauhut, B. (1990). The outlier behaviour of probability distributions. *Journal of Statistical Planning and Inference*, 26(2):237–252.

Gertler, J. (1998). *Fault detection and diagnosis in engineering systems*. CRC press.

Ghahramani, Z. (2001). An introduction to hidden markov models and bayesian networks. In *Hidden Markov models: applications in computer vision*, pages 9–41. World Scientific.

Ghareb, A. S., Bakar, A. A., and Hamdan, A. R. (2016). Hybrid feature selection based on enhanced genetic algorithm for text categorization. *Expert Systems with Applications*, 49:31–47.

Ghojogh, B., Karray, F., and Crowley, M. (2019). Hidden markov model: Tutorial.

Gikhman, I. I. and Skorokhod, A. *The theory of stochastic processes II*.

Gonçalves, J. F. and Resende, M. G. (2011). Biased random-key genetic algorithms for combinatorial optimization. *Journal of Heuristics*, 17(5):487–525.

Grant, M., Boyd, S., and Ye, Y. (2009). Cvx: Matlab software for disciplined convex programming.

Gu, Q., Zhu, L., and Cai, Z. (2009). Evaluation measures of the classification performance of imbalanced data sets. In *International symposium on intelligence computation and applications*, pages 461–471. Springer.

Guo, G. and Chen, H. (2009). Semi-supervised learning applied to large data sets with very few labeled examples. In *Fuzzy Systems and Knowledge Discovery, Fourth International Conference on*, volume 1, pages 281–285, Los Alamitos, CA, USA. IEEE Computer Society.

Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182.

Habibi, H., Howard, I., and Simani, S. (2019). Reliability improvement of wind turbine power generation using model-based fault detection and fault tolerant control: A review. *Renewable Energy*, 135:877 – 896.

Hamdani, T. M., Won, J.-M., Alimi, A. M., and Karray, F. (2007). Multi-objective feature selection with nsga ii. In *International conference on adaptive and natural computing algorithms*, pages 240–247. Springer.

Hawkins, D. M. (1980). *Identification of outliers*, volume 11. Springer.

Hawkins, S., He, H., Williams, G., and Baxter, R. (2002). Outlier detection using replicator neural networks. In *International Conference on Data Warehousing and Knowledge Discovery*, pages 170–180. Springer.

Hayes, D. (1977). Rays of hope: the transition to a post-petroleum world.

He, Z., Xu, X., and Deng, S. (2003). Discovering cluster-based local outliers. *Pattern Recognition Letters*, 24(9-10):1641–1650.

Hodge, V. and Austin, J. (2004). A survey of outlier detection methodologies. *Artificial intelligence review*, 22(2):85–126.

Hor, C.-L. and Crossley, P. A. (2005). Knowledge extraction from intelligent electronic devices. In *Transactions on Rough Sets III*, pages 82–111. Springer.

Hossin, M. and Sulaiman, M. (2015). A review on evaluation metrics for data classification evaluations. *International Journal of Data Mining & Knowledge Management Process*, 5(2):1.

Huang, B., Buckley, B., and Kechadi, T.-M. (2010). Multi-objective feature selection by using nsga-ii for customer churn prediction in telecommunications. *Expert Systems with Applications*, 37(5):3638–3646.

Isermann, R. (2005). *Fault-diagnosis systems: an introduction from fault detection to fault tolerance*. Springer Science & Business Media.

Jacinto, E. d. A. S. (2016). Determinação do potencial eólico do município de barreirinhas/ma.

Jain, P. (2011). *Wind energy engineering*. New York: McGraw-Hill,.

Jirapech-Umpai, T. and Aitken, S. (2005). Feature selection and classification for microarray data analysis: Evolutionary methods for identifying predictive genes. *BMC bioinformatics*, 6(1):148.

John, G. H., Kohavi, R., and Pfleger, K. (1994). Irrelevant features and the subset selection problem. In *Machine Learning Proceedings 1994*, pages 121–129. Elsevier.

Jordan, M. I. (2003). An introduction to probabilistic graphical models.

Jović, A., Brkić, K., and Bogunović, N. (2015). A review of feature selection methods with applications. In *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 1200–1205. IEEE.

Kaldellis, J. K. and Zafirakis, D. (2011). The wind energy (r) evolution: A short review of a long history. *Renewable energy*, 36(7):1887–1901.

Kannan, S. and Somasundaram, K. (2015). A review of outlier prediction techniques in data mining. *Research Journal of Applied Sciences, Engineering and Technology*, 10(9):1021–1028.

Kautz, T., Eskofier, B. M., and Pasluosta, C. F. (2017). Generic performance measure for multiclass-classifiers. *Pattern Recognition*, 68:111–125.

Keivanpour, S., Ramudhin, A., and Ait Kadi, D. (2017). The sustainable worldwide offshore wind energy potential: A systematic review. *Journal of Renewable and Sustainable Energy*, 9(6):065902.

Khalid, S., Khalil, T., and Nasreen, S. (2014). A survey of feature selection and feature extraction techniques in machine learning. In *2014 Science and Information Conference*, pages 372–378. IEEE.

Kidam, K. and Hurme, M. (2013). Analysis of equipment failures as contributors to chemical process accidents. *Process safety and environmental protection*, 91(1-2):61–78.

Kobbacy, K. A. H. and Murthy, D. P. (2008). *Complex system maintenance handbook*. Springer Science & Business Media.

Kohavi, R. and John, G. H. (1997). Wrappers for feature subset selection. *Artificial intelligence*, 97(1-2):273–324.

Koller, D. and Friedman, N. (2009). *Probabilistic graphical models: principles and techniques*. MIT press.

Kouadri, A., Hajji, M., Harkat, M.-F., Abodayeh, K., Mansouri, M., Nounou, H., and Nounou, M. (2020). Hidden markov model based principal component analysis for intelligent fault diagnosis of wind energy converter systems. *Renewable Energy*, 150:598–606.

Kusiak, A. and Verma, A. (2012). Analyzing bearing faults in wind turbines: A data-mining approach. *Renewable Energy*, 48:110–116.

Lawrence, D. (1991). Handbook of genetic algorithms. *Van Nostrand Reinhold*.

Lee, J., Kao, H.-A., and Yang, S. (2014). Service innovation and smart analytics for industry 4.0 and big data environment. *Procedia Cirp*, 16:3–8.

Letcher, T. M. (2017). *Wind energy engineering: a handbook for onshore and offshore wind turbines*. Academic Press.

Li, A.-D., He, Z., and Zhang, Y. (2016). Bi-objective variable selection for key quality characteristics selection based on a modified nsga-ii and the ideal point method. *Computers in Industry*, 82:95–103.

Li, D., Ho, S.-C. M., Song, G., Ren, L., and Li, H. (2015). A review of damage detection methods for wind turbine blades. *Smart Materials and Structures*, 24(3):033001.

Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R. P., Tang, J., and Liu, H. (2018). Feature selection: A data perspective. *ACM Computing Surveys (CSUR)*, 50(6):94.

Li, J., Zhang, X., Zhou, X., and Lu, L. (2019). Reliability assessment of wind turbine bearing based on the degradation-hidden-markov model. *Renewable Energy*, 132:1076 – 1087.

Lu, L., Yan, J., and de Silva, C. W. (2015). Dominant feature selection for the fault diagnosis of rotary machines using modified genetic algorithm and empirical mode decomposition. *Journal of Sound and Vibration*, 344:464–483.

Lu, X., McElroy, M. B., and Kiviluoma, J. (2009). Global potential for wind-generated electricity. *Proceedings of the National Academy of Sciences*, 106(27):10933–10938.

Lucena, J. d. A. Y. and Lucena, K. Â. A. (2019). Wind energy in brazil: an overview and perspectives under the triple bottom line. *Clean Energy*, 3(2):69–84.

Mao, W., Tian, S., Fan, J., Liang, X., and Safian, A. (2020). Online detection of bearing incipient fault with semi-supervised architecture and deep feature representation. *Journal of Manufacturing Systems*, 55:179–198.

Márquez, F. P. G., Tobias, A. M., Pérez, J. M. P., and Papaelias, M. (2012). Condition monitoring of wind turbines: Techniques and methods. *Renewable Energy*, 46:169–178.

Maxion, R. A. (1990). Anomaly detection for diagnosis. In *Digest of Papers. Fault-Tolerant Computing: 20th International Symposium*, pages 20–21. IEEE Computer Society.

Mukhopadhyay, A., Maulik, U., Bandyopadhyay, S., and Coello, C. A. C. (2013). A survey of multiobjective evolutionary algorithms for data mining: Part i. *IEEE Transactions on Evolutionary Computation*, 18(1):4–19.

Niu, Z., Shi, S., Sun, J., and He, X. (2011). A survey of outlier detection methodologies and their applications. In *International Conference on Artificial Intelligence and Computational Intelligence*, pages 380–387. Springer.

Pandit, R. K. and Infield, D. (2018). Scada-based wind turbine anomaly detection using gaussian process models for wind turbine condition monitoring purposes. *IET Renewable Power Generation*, 12(11):1249–1255.

Pathan, A.-S. K. (2014). *The state of the art in intrusion prevention and detection*. CRC press.

Poritz, A. B. (1988). Hidden markov models: A guided tour. In *Proceedings of the IEEE Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7–13.

Qiao, W. and Lu, D. (2015). A survey on wind turbine condition monitoring and fault diagnosis—part i: Components and subsystems. *IEEE Transactions on Industrial Electronics*, 62(10):6536–6545.

Quinlan, J. (1993). Program for machine learning. *C4. 5*.

Rabiner, L. and Juang, B. (1986). An introduction to hidden markov models. *ieee assp magazine*, 3(1):4–16.

Ramaswamy, S., Rastogi, R., and Shim, K. (2000). Efficient algorithms for mining outliers from large data sets. In *ACM Sigmod Record*, volume 29, pages 427–438. ACM.

Rocha, P. A. C., de Sousa, R. C., de Andrade, C. F., and da Silva, M. E. V. (2012). Comparison of seven numerical methods for determining weibull parameters for wind energy generation in the northeast region of brazil. *Applied Energy*, 89(1):395–400.

Ross, S. M. (2002). *Probability models for computer science*. Harcourt Academic Press San Diego.

SANTOS, L. (2017). Avanços da energia eólica no brasil: Uma análise das políticas e seus resultados. Master's thesis, Universidade Federal do Espírito Santo.

Schlechtingen, M., Santos, I. F., and Achiche, S. (2013). Using data-mining approaches for wind turbine power curve monitoring: a comparative study. *IEEE Transactions on Sustainable Energy*, 4(3):671–679.

Seymore, K., McCallum, A., Rosenfeld, R., et al. (1999). Learning hidden markov model structure for information extraction. In *AAAI-99 workshop on machine learning for information extraction*, pages 37–42.

Siedlecki, W. and Sklansky, J. (1993). A note on genetic algorithms for large-scale feature selection. In *Handbook of Pattern Recognition and Computer Vision*, pages 88–107. World Scientific.

Singh, D. and Singh, B. (2019). Investigating the impact of data normalization on classification performance. *Applied Soft Computing*, page 105524.

Singh, K. and Upadhyaya, S. (2012). Outlier detection: applications and techniques. *International Journal of Computer Science Issues (IJCSI)*, 9(1):307.

Singh, U. and Singh, S. N. (2017). Optimal feature selection via nsga-ii for power quality disturbances classification. *IEEE Transactions on Industrial Informatics*, 14(7):2994–3002.

Sittón-Candanedo, I., Alonso, R. S., Corchado, J. M., Rodríguez-González, S., and Casado-Vara, R. (2019). A review of edge computing reference architectures and a new global edge proposal. *Future Generation Computer Systems*, 99:278–294.

Soufan, O., Kleftogiannis, D., Kalnis, P., and Bajic, V. B. (2015). Dwfs: a wrapper feature selection tool based on a parallel genetic algorithm. *PloS one*, 10(2):e0117988.

Soyel, H., Tekguc, U., and Demirel, H. (2011). Application of nsga-ii to feature selection for facial expression recognition. *Computers & Electrical Engineering*, 37(6):1232–1240.

Srinivas, N. and Deb, K. (1994). Muiltiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary computation*, 2(3):221–248.

Stetco, A., Dinmohammadi, F., Zhao, X., Robu, V., Flynn, D., Barnes, M., Keane, J., and Nenadic, G. (2018). Machine learning methods for wind turbine condition monitoring: A review. *Renewable energy*.

Tan, C. J., Lim, C. P., and Cheah, Y.-N. (2014). A multi-objective evolutionary algorithm-based ensemble optimizer for feature selection and classification with neural network models. *Neurocomputing*, 125:217–228.

Tang, J., Alelyani, S., and Liu, H. (2014). Feature selection for classification: A review. *Data classification: Algorithms and applications*, page 37.

Tang, K.-S., Man, K.-F., Kwong, S., and He, Q. (1996). Genetic algorithms and their applications. *IEEE signal processing magazine*, 13(6):22–37.

Tautz-Weinert, J. and Watson, S. J. (2016). Using scada data for wind turbine condition monitoring–a review. *IET Renewable Power Generation*, 11(4):382–394.

Tavner, P. (2012). *Offshore Wind Turbines: Reliability, availability and maintenance*. Energy Engineering. Institution of Engineering and Technology.

Tekguc, U., Soyel, H., and Demirel, H. (2009). Feature selection for person-independent 3d facial expression recognition using nsga-ii. In *2009 24th International Symposium on Computer and Information Sciences*, pages 35–38. IEEE.

Thiel, C., Scherer, S., and Schwenker, F. (2007). Fuzzy-input fuzzy-output one-against-all support vector machines. In *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, pages 156–165. Springer.

Thomas, M. S. and McDonald, J. D. (2017). *Power system SCADA and smart grids*. CRC press.

Tong, W. (2010). *Wind power generation and wind turbine design*. WIT press.

Toso, R. F. and Resende, M. G. (2015). A c++ application programming interface for biased random-key genetic algorithms. *Optimization Methods and Software*, 30(1):81–93.

Vafaie, H. and De Jong, K. (1992). Genetic algorithms as a tool for feature selection in machine learning. In *Proceedings Fourth International Conference on Tools with Artificial Intelligence TAI'92*, pages 200–203. IEEE.

Vafaie, H. and De Jong, K. (1997). Genetic algorithms as a tool for feature selection in machine learning.

Van Veldhuizen, D. A. (1999). Multiobjective evolutionary algorithms: classifications, analyses, and new innovations. Technical report, AIR FORCE INST OF TECH WRIGHT-PATTERSONAFB OH SCHOOL OF ENGINEERING.

Vapnik, V. (2013). *The nature of statistical learning theory*. Springer science & business media.

Vapnik, V. N. (1999). An overview of statistical learning theory. *IEEE transactions on neural networks*, 10(5):988–999.

Wang, C.-H., Liu, J.-F., Hong, T.-P., and Tseng, S.-S. (1999). A fuzzy inductive learning strategy for modular rules. *Fuzzy sets and Systems*, 103(1):91–105.

Wang, H., Bah, M. J., and Hammad, M. (2019). Progress in outlier detection techniques: A survey. *IEEE Access*, 7:107964–108000.

Wang, K.-S., Sharma, V. S., and Zhang, Z.-Y. (2014). Scada data based condition monitoring of wind turbines. *Advances in Manufacturing*, 2(1):61–69.

Wang, L., Wang, Y., and Chang, Q. (2016). Feature selection methods for big data bioinformatics: A survey from the search perspective. *Methods*, 111:21–31.

Wasserman, L. (2013). *All of statistics: a concise course in statistical inference*. Springer Science & Business Media.

Wei, L. and Keogh, E. (2006). Semi-supervised time series classification. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 748–753.

Welikala, R. A., Fraz, M. M., Dehmeshki, J., Hoppe, A., Tah, V., Mann, S., Williamson, T. H., and Barman, S. A. (2015). Genetic algorithm based feature selection combined with dual classification for the automated detection of proliferative diabetic retinopathy. *Computerized Medical Imaging and Graphics*, 43:64–77.

Whitley, D. (1994). A genetic algorithm tutorial. *Statistics and computing*, 4(2):65–85.

Widodo, A. and Yang, B.-S. (2007). Support vector machine in machine condition monitoring and fault diagnosis. *Mechanical systems and signal processing*, 21(6):2560–2574.

Williamson, T. (2002). *Vagueness*. Routledge.

Wu, S. and Flach, P. (2005). A scored auc metric for classifier evaluation and selection. In *Second Workshop on ROC Analysis in ML, Bonn, Germany*.

Xue, B., Zhang, M., Browne, W. N., and Yao, X. (2015). A survey on evolutionary computation approaches to feature selection. *IEEE Transactions on Evolutionary Computation*, 20(4):606–626.

Yang, W., Tavner, P. J., Crabtree, C. J., Feng, Y., and Qiu, Y. (2014). Wind turbine condition monitoring: technical and commercial challenges. *Wind Energy*, 17(5):673–693.

Yu, L. and Liu, H. (2004). Efficient feature selection via analysis of relevance and redundancy. *Journal of machine learning research*, 5(Oct):1205–1224.

Zaher, A., McArthur, S., Infield, D., and Patel, Y. (2009). Online wind turbine fault detection through automated scada data analysis. *Wind Energy: An International Journal for Progress and Applications in Wind Power Conversion Technology*, 12(6):574–593.

Zhang, Q. and Li, H. (2007). Moea/d: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on evolutionary computation*, 11(6):712–731.

Zhang, Z. (2014). Comparison of data-driven and model based methodologies of wind turbine fault detection with scada data. *EWEA March*.

Zhao, R., Al Iqbal, M. R., Bennett, K. P., and Ji, Q. (2016). Wind turbine fault prediction using soft label svm. In *2016 23rd International conference on pattern recognition (ICPR)*, pages 3192–3197. IEEE.

Zucchini, W., MacDonald, I. L., and Langrock, R. (2017). *Hidden Markov models for time series: an introduction using R*. CRC press.

A- Máquinas de Vetores de Suporte

Máquinas de vetores de suporte são uma aplicação da teoria do aprendizado estatístico (TAE), cujas bases teóricas foram lançadas por Vapnik ainda na década de 1960, mas permaneceu no campo da análise teórica até meados da década de 1990 [Vapnik, 1999]. O desenvolvimento de algoritmos derivados da teoria do aprendizado estatístico possibilitou o tratamento de muitos problemas do mundo real, dos quais observamos os valores de entrada e saída, mas não compreendemos o mecanismo que estabeleceu esta relação [Widodo and Yang, 2007]. Isso foi possível porque Vapnik [2013] fundou o modelo de aprendizado a partir dos dados como um problema de estimação de funções que devem explicar a dependência entre entrada e saída, além de ter habilidade de generalização. O corpo desse modelo de aprendizado é composto por três componentes:

- Um gerador (G) de vetor de dados aleatórios $\boldsymbol{x} \in \Re^n$, amostrados independentemente de uma distribuição de probabilidade desconhecida porém fixa $P(x)$;
- Um supervisor (S) que atribui um valor de saída y a cada dado de entrada x, de acordo com a distribuição fixa e desconhecida $P(y|x)$;
- Uma máquina de aprendizado (LM) que implementa um conjunto de funções $f(\boldsymbol{x}, \wr),\ \wr \in \Lambda$, onde Λ é um conjunto arbitrário de parâmetros que rege o comportamento da função. Em algum estágio do processamento do aprendizado, ocorre a seleção da função $f(\boldsymbol{x}, \wr),\ \wr \in \Lambda$ que melhor aproxima a saída fornecida pelo supervisor y dos dados do conjunto de treinamento $\boldsymbol{x}$.

Um conjunto finito de l amostras *i.i.d.* emitido pelo gerador e rotulado pelo supervisor é utilizado para o treinamento:

$$(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_l, y_l),\ \boldsymbol{x} \in \Re^n,\ y \in \{-1, +1\}$$

Este conjunto de treinamento é obtido de forma aleatória e independente de acordo com uma distribuição de probabilidade conjunta $P(\boldsymbol{x}, y) = P(\boldsymbol{x})P(y|\boldsymbol{x})$. A Figura 45 esquematiza o modelo de aprendizado estruturado pela TAE.

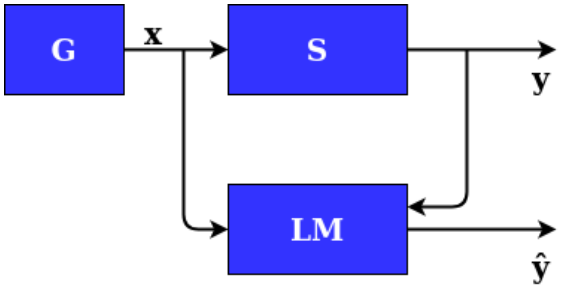

Figura 45 – Modelo de aprendizado a partir de exemplos utilizado pela teoria do aprendizado estatístico com a discriminação das componentes de geração de dados (G), supervisão (S) e máquina de aprendizado (LM). Adaptado de [Vapnik, 2013].

Como apresenta a Figura 45, supomos que a máquina de aprendizado (LM) possui a tarefa de executar o mapeamento $\boldsymbol{x}_i \longmapsto y_i$. O mecanismo de seleção que induz a aproximação da dependência funcional entre os pares $(\boldsymbol{x}, y)$ utiliza uma função de perda ou discrepância $\mathrm{L}(y, f(\boldsymbol{x}, \wr))$ para avaliar as funções $f(\boldsymbol{x}, \wr),\ \wr \in \Lambda$ no conjunto de possíveis mapeamentos $\boldsymbol{x} \longmapsto f(\boldsymbol{x}, \wr)$. Esta avaliação não necessariamente utiliza todos os pontos do conjunto $\boldsymbol{x} \in X \subset \Re^n$ (que pode ser contínuo ou discreto), mas pode utilizar somente pontos de interesse $\boldsymbol{x}^* \in X^*$, onde $X^* \in \Re^m$, com $\Re^m \subset \Re^n$, onde $m < n$.

A.1- Minimização do erro empírico

O processamento orientado pela minimização do risco empírico (ERM, do inglês *emprirical risk minimization*) R_{emp} é centrado na minimização do erro associado às amostras contidas no subconjunto de treinamento, para uma medida de probabilidade $P(\boldsymbol{x}, y)$ desconhecida. Este princípio é empregado em muitos algoritmos de aprendizado supervisionado [Vapnik, 1999].

O problema de aprendizado apresenta diferentes formulações para os três principais problemas: reconhecimento de padrões, estimação da regressão e estimação da densidade [Vapnik, 1999]. As máquinas de vetores de suporte enquadram-se em problemas de reconhecimento de padrões e de estimação da regressão. Em sua formulação para problemas de reconhecimento de padrões, o supervisor atribui valores binários $y = \{-1, 1\}$ aos dados $\boldsymbol{x}$ enquanto $f(\boldsymbol{x}, \wr),\ \wr \in \Lambda$ é um conjunto de funções indicadoras sob a forma:

$$\mathrm{L}(y, f(\boldsymbol{x}, \wr)) = \begin{cases} 0, & \text{se } y = f(\boldsymbol{x}, \wr) \\ 1, & \text{se } y \neq f(\boldsymbol{x}, \wr) \end{cases} \tag{54}$$

De acordo com Vapnik [2013], o objetivo é selecionar a função que minimize o risco esperado, apresentado na Equação (55).

$$R(\wr) = \int \mathrm{L}(y, f(\boldsymbol{x}, \wr)) dP(\boldsymbol{x}, y) \tag{55}$$

A função perda ou discrepância $\mathrm{L}(y, f(\boldsymbol{x}, \wr))$ fornece, dessa forma, a probabilidade do erro de classificação, determinado pela lógica apresentada na Equação (54). Com isso, o problema de aprendizado consiste na identificação da função $f(\boldsymbol{x}, \wr),\ \wr \in \Lambda$ que fornece o menor erro de classificação quando a medida de probabilidade $P(\boldsymbol{x}, y)$ é desconhecida, mas os dados são fornecidos. A Equação (56) calcula o erro empírico para o problema de reconhecimento de padrões, considerando o cálculo do erro de classificação no conjunto de treinamento.

$$R_{emp}(\wr) = \frac{1}{l} \sum_{i=1}^{l} \mathrm{L}(y_i, f(\boldsymbol{x}, \wr)) \tag{56}$$

Em extensão, a Equação (56) é base de construção de outros problemas clássicos como, por exemplo, o método do erro mínimo quadrado utilizado em problemas de regressão, como apresentado na Equação (57)

$$R_{emp}(\wr) = \frac{1}{l} \sum_{i=1}^{l} \mathrm{L}(y_i - f(\boldsymbol{x}, \wr))^2, \tag{57}$$

ou o método de máximo verossimilhança em problemas de estimação de densidade $p(\boldsymbol{x}, \wr)$, cujo objetivo é a minimização da Equação (58)

$$R_{emp}(\wr) = -\frac{1}{l} \sum_{i=1}^{l} \ln p(\boldsymbol{x}_i, \wr) \tag{58}$$

A.2- Minimização do erro estrutural

Contudo, máquinas de vetores de suporte são guiadas pelo princípio de minimização do risco estrutural (SRM, do inglês *structural risk minimization*), que objetiva minimizar o erro de generalização do modelo [Vapnik, 2013]. Esta característica é alcançada pela determinação de um limite superior do teste de erro, como expresso na Equação (59).

$$R(\wr) = R_{emp}(\wr) + \phi(\frac{l}{h_k}) \tag{59}$$

onde $R_{emp}(\wr)$ é o erro associado ao conjunto de treinamento e $\phi(\frac{l}{h_k})$ estabelece um intervalo de confiança do erro de generalização de acordo com a quantidade de amostras l e o índice h da dimensão VC (Vapnik e Chervonenkis) [Cortes and Vapnik, 1995].

A dimensão VC é uma propriedade compartilhada pelas funções $f(\boldsymbol{x}, \wr),\ \wr \in \Lambda$ pertencentes a cada um dos conjuntos aninhados $\mathbf{S}_1 \subset \mathbf{S} \subset \ldots \mathbf{S}_n \ldots$, onde $\mathbf{S}_k = \{f(\boldsymbol{x}, \wr),\ \wr \in \Lambda_k\}$. Inicialmente, poderíamos relacionar a dimensão VC ao número de parâmetros embarcados em $\wr$, contudo há contraprovas que impedem esta conclusão [Vapnik, 1999]. A dimensão VC mede a capacidade das funções $f(\wr),\ \wr \in \Lambda_k$ contidas em $\mathbf{S}_k$. Essa capacidade é definida como o número de pontos que a função pode separar perfeitamente em classes distintas. Quanto maior o índice h_k da dimensão VC associado a cada conjunto $\mathbf{S}_k$, mais complexas são as funções $f(\wr),\ \wr \in \Lambda_k$ contidas neste conjunto, dado que $h_1 \leq h_2 \ldots \leq h_n$.

Em um conjunto de treinamento com l amostras, há 2^l formas diferentes de atribuir rótulos binários $y = \{-1, +1\}$ aos dados $\boldsymbol{x}$. A Figura 46 apresenta disposições arbitrárias de uma reta, ou seja, a representação gráfica de uma função $f(\wr),\ \wr \in \Lambda$ suficiente para separar os 3 pontos não-colineares no conjunto em $\Re^2$ em duas classes nas $2^3 = 8$ formas possíveis. O índice da dimensão VC do conjunto em $\Re^n$ coincide com o resultado da equação $h = n + 1$. Assim, uma reta em $\Re^2$ não é capaz de dividir uma quantidade maior do que 3 pontos.

O gráfico da Figura 47 ilustra o *tradeoff* fornecido pela Equação (59) adotando o critério da minimização do risco estrutural, que intenciona minimizar ambas as parcelas dessa equação. Em destaque na Figura, o índice h^* identifica a dimensão VC que contêm a função $f(\wr),\ \wr \in \Lambda$ que alcança boa qualidade de aproximação dos dados sem incorrer

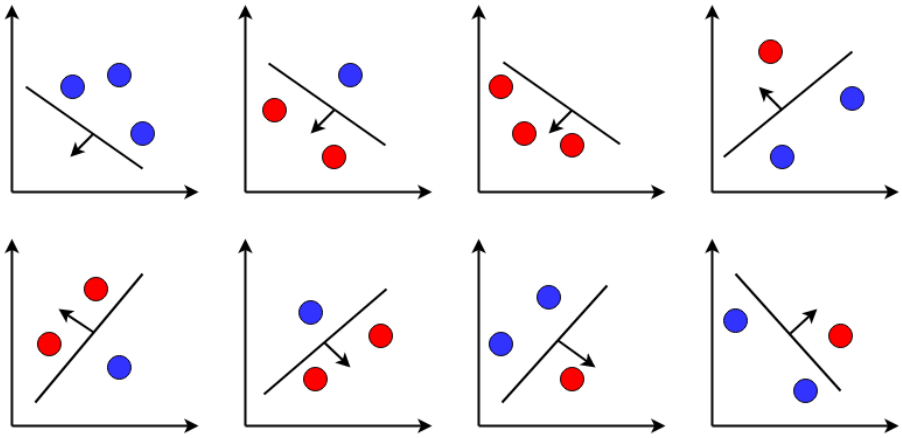

Figura 46 – Exemplo de aplicação da teoria da dimensão VC no espaço $\Re^2$, no qual uma reta é capaz de separar 3 pontos.

em efeitos indesejados como sobreajuste ou subajuste, causados pela adoção de funções muito e pouco complexas, respectivamente.

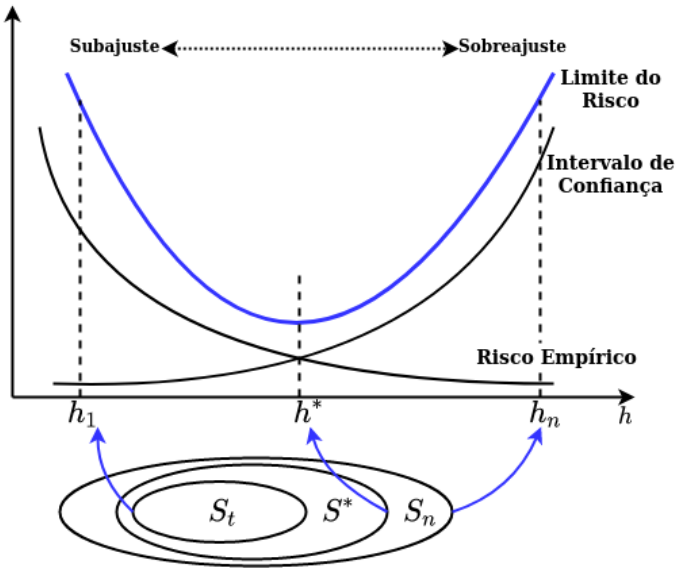

Figura 47 – O risco $R(\wr)$ é a soma do risco empírico e o intervalo de confiança do erro de generalização. O risco empírico decresce à medida que aumenta o índice da dimensão do problema. Este *tradeoff* permite avaliar a qualidade da aproximação entre a função $\{f(\wr),\ \wr \in \Lambda_k$ e os dados. O menor valor do risco $R(\wr)$ determina o melhor índice h^* da dimensão VC para o problema. Adaptado de [Vapnik, 2013].

No escopo das máquinas de vetores de suporte, os conceitos da minimização do erro estrutural se manifestam na determinação da função $f(\wr),\ \wr \in \Lambda$ que constrói o hiperplano ótimo na separação das amostras positivas das negativas, em uma classificação binária. Na Seção A.3, discutiremos a aplicação das máquinas de vetores de suporte em problemas de classificação binária.

A.3- Classificação Binária utilizando máquinas de vetores de suporte

Máquinas de vetores de suporte adotam como estratégia de redução do risco $R(\wr)$ a manutenção de um valor fixo para o erro empírico $R_{emp}(\wr)$ e a minimização do intervalo de confiança do erro de generalização [Vapnik, 2013]. A intenção do algoritmo é construir um hiperplano ótimo $(\boldsymbol{w}, b)$ que separa linearmente as amostras em duas classes. Este hiperplano está contido numa região limítrofe de separação dessas classes denominada margem. A maximização dessa margem de separação das classes corresponde à otimização do intervalo de confiança do erro de generalização apresentado na Equação (59) [Vapnik, 2013]. A Figura 48 ilustra o hiperplano ótimo $(\boldsymbol{w}^T\boldsymbol{x}) - b = 0$ construído na classificação das amostras do conjunto de treinamento.

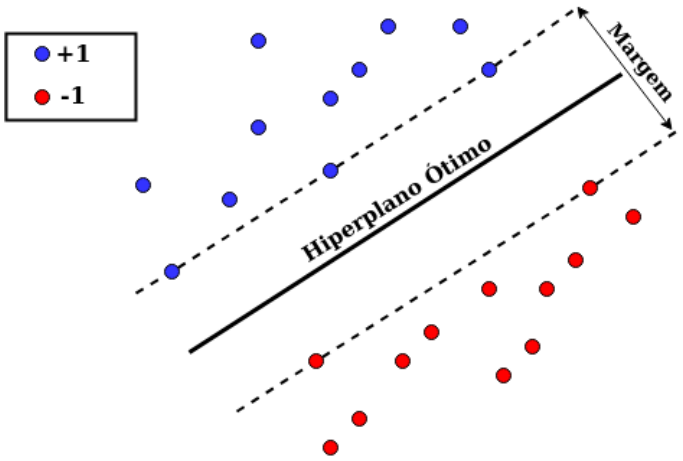

Figura 48 – Hiperplano ótimo de separação de classes determinado pela maximização da margem durante o processamento das máquinas de vetores de suporte.

Os pontos posicionados sobre as margens são denominados vetores de suporte. Já os dados dispostos acima e abaixo do hiperplano atendem ao seguinte critério de classificação:

$$\begin{cases} (\boldsymbol{w}^T\boldsymbol{x}_i) - b \geq \Delta, & \text{se } y_i = +1 \\ (\boldsymbol{w}^T\boldsymbol{x}_i) - b \leq -\Delta, & \text{se } y_i = -1 \end{cases}$$

onde $\Delta = 1/|\boldsymbol{w}|$ é a largura da margem e $i = 1, \ldots, l$.

Se o conjunto de treinamento é linearmente separável, o hiperplano é maximal com $|\boldsymbol{w}_*| = 1$ e realiza a separação das classes sem erros [Vapnik, 2013]. Assim, a

formulação anterior do hiperplano maximal pode ser simplificada sob a forma:

$$y_i[(\boldsymbol{w}_*^T \boldsymbol{x}_i) - b] \geq 1,\ i = 1, \ldots, l \tag{60}$$

A dimensão VC do hiperplano é determinada a partir da premissa de que o vetor de dados $\boldsymbol{x} \in X$ está contido em uma esfera de raio R. A Equação (61) demonstra que a dimensão VC do hiperplano pode ser menor do que a enunciada anteriormente pela regra geral, cujo índice foi calculado pela equação $h = n + 1$ no espaço $\Re^n$. Este resultado guia a aplicação do princípio SRM para generalizar o modelo [Vapnik, 1999].

$$h \leq min\left(\left[\frac{R^2}{\Delta^2}\right], n\right) + 1 \tag{61}$$

onde R é o raio da esfera na qual está contido o vetor de dados $\boldsymbol{x}$, Δ é a largura da margem e n é a dimensão do espaço $\Re^n$.

De acordo com a Equação (61), observamos que o valor da margem Δ deve ser maximizado para ocorrer a redução da dimensão VC que favorece o *tradeoff* da Figura 47. Como enunciado, $\Delta = 1/|\boldsymbol{w}|$, logo o valor dos coeficientes $\boldsymbol{w}$ deve ser minimizado. A construção do hiperplano ótimo maximal das máquinas de vetores de suporte é um problema de programação quadrática [Vapnik, 2013]. A Equação (62) apresenta a forma primal desse problema de otimização.

$$\begin{aligned} &\text{Mininimizar} \quad \frac{1}{2}||\boldsymbol{w}^2|| \\ &\text{sujeito a:} \\ &\qquad y_i[(\boldsymbol{w}^T \boldsymbol{x}_i) - b] \geq 1,\ i = 1, \ldots, l \end{aligned} \tag{62}$$

A Equação (62) pode ser reformulada para a obtenção da forma dual do problema. Substituindo a restrição da formulação primal pelos multiplicadores de Lagrange, obtemos:

$$\mathcal{L}(\boldsymbol{w}, b, \alpha) \equiv \frac{1}{2}||\boldsymbol{w}||^2 - \sum_{i=1}^{l} \alpha_i y_i(\boldsymbol{w}^T \boldsymbol{x}_i + b) + \sum_{i=1}^{l} \alpha_i \tag{63}$$

onde α_i são multiplicadores de Lagrange.

A função de Lagrange $\mathcal{L}(\boldsymbol{w}, b, \alpha)$ deve ser minimizada com relação a $\boldsymbol{w}$ e b e maximizada em relação a $\alpha_i > 0$. No ponto de sela, as seguintes derivadas de $\mathcal{L}(\boldsymbol{w}, b, \alpha)$

sobre essas variáveis devem ser obtidas:

$$\frac{\partial \mathcal{L}(\boldsymbol{w}_0, b_0, \alpha^0)}{\partial b} = \mathbf{0}$$
$$\frac{\partial \mathcal{L}(\boldsymbol{w}_0, b_0, \alpha^0)}{\partial \boldsymbol{w}} = \mathbf{0}.$$

- Para o cálculo do hiperplano ótimo, os coeficientes α_i^0 devem satisfazer à seguinte restrição:

$$\sum_{i=1}^{l} \alpha_i^0 y_i = 0, \quad \alpha_i^0 \geq 0, \quad i = 1, \ldots, l \tag{64}$$

- Os coeficientes $\boldsymbol{w}_0$ do hiperplano ótimo são formados pela combinação linear utilizando o vetor de dados de treinamento:

$$\boldsymbol{w}_0 = \sum_{i=1}^{l} y_i \alpha_i^0 x_i, \quad \alpha_i^0 \geq 0 \tag{65}$$

Observamos que somente os multiplicadores de Lagrange α_i não nulos no ponto de sela atendem a restrição da forma primal $\alpha_i[y_i[(\boldsymbol{w}^T \boldsymbol{x}_i) + b) - 1] = 0$ [Vapnik, 2013]. Os multiplicadores α_i que atendem este critério determinam os pontos pertencentes aos vetores de suporte, construídos em respeito à igualdade da Equação (65). A margem é determinada pelos pontos $\boldsymbol{x}_i$ dos vetores de suporte, sendo os demais pontos do conjunto de treinamento irrelevantes, uma vez que para estes pontos $\alpha_i = 0$.

Com as condições do teorema Kühn-Tucke satisfeitas, podemos extrair a forma dual do problema de otimização, como expressa na Equação (66).

$$\begin{aligned} &\text{Maximizar} \quad \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j}^{l} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) \\ &\text{sujeito a:} \\ &\sum_{i=1}^{l} \alpha_i y_i = 0, \\ &\alpha_i \geq 0, \quad i = 1, \ldots, l \end{aligned} \tag{66}$$

Muitos conjuntos de dados não possibilitam a separação linear das classes sem erros, como discutido no desenvolvimento das formulações primal e dual do problema

de otimização, como visto nas equações (62) e (66). A generalização dessas equações introduz variáveis de folga não negativas $\xi_i \geq 0$ que permitem a manipulação desses conjuntos de dados e garantem que a solução viável sempre existe [Vapnik, 2013]. A Equação (67) apresenta a formulação primal do problema de minimização que constrói o hiperplano de margem suave, que tolera erros na separação das classes.

$$\begin{aligned} &\text{Mininimizar} \quad \frac{1}{2}||\boldsymbol{w}||^2 + C\left(\sum_{i=1}^{l}\xi_i\right) \\ &\text{sujeito a:} \\ &\qquad y_i[(\boldsymbol{w}^T\boldsymbol{x}_i) - b] \geq 1 - \xi_i,\ i = 1, \ldots, l \end{aligned} \tag{67}$$

onde a constante C é o custo de uso da variável de folga ξ.

A Figura 49 ilustra a interação da variável de folga ξ na geração do hiperplano ótimo durante a classificação dos pontos. Todos os pontos $\boldsymbol{x}_i$ para os quais a variável de folga ξ é diferente de zero correspondem a erros em relação à margem. Quanto maior o segundo termo da função objetivo da Equação (67), maior a fração de erros durante a classificação. Assim, a determinação de uma constante C adequada é primordial para o direcionamento da extração do hiperplano ótimo.

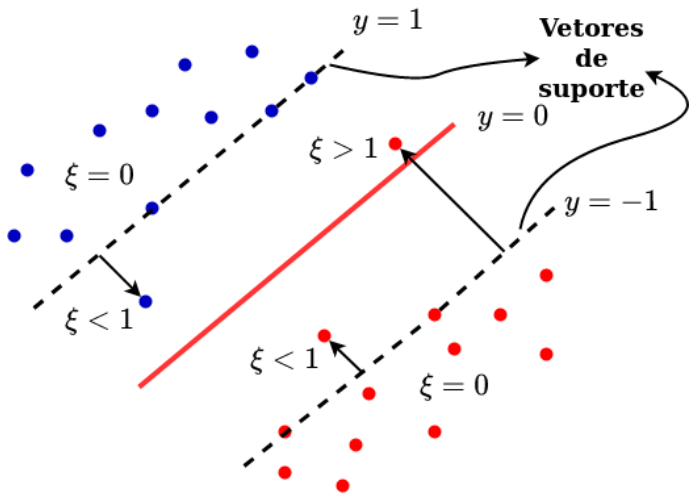

Figura 49 – Determinação do hiperplano de margem suave, que tolera erros de classificação através do relaxamento das variáveis de folga ξ.

A função de Lagrange é novamente utilizada para a derivação da formulação dual do problema.

$$\mathcal{L}(\boldsymbol{w}, \xi, b, \alpha) \equiv \frac{1}{2}||\boldsymbol{w}||^2 - \sum_{i=1}^{l} \alpha_i y_i (\boldsymbol{w}^T \boldsymbol{x}_i + b) + \sum_{i=1}^{l} \alpha_i + C \sum_{i=1}^{l} \xi_i \tag{68}$$

onde α_i são multiplicadores de Lagrange e C é uma constante que determina o custo de uso da variável de folga ξ.

A forma dual é obtida a partir da função de Lagrange $\mathcal{L}(\boldsymbol{w}, \xi, b, \alpha)$ minimizando em relação a $\boldsymbol{w}$, ξ e b e maximizando em relação a α [Vapnik, 2013]. No ponto de sela, as seguintes derivadas em relação a essas variáveis devem ser correspondidas:

$$\begin{aligned} \frac{\partial \mathcal{L}(\boldsymbol{w}_0, \xi_0, b_0, \alpha^0)}{\partial b} &= \mathbf{0}, \\ \frac{\partial \mathcal{L}(\boldsymbol{w}_0, \xi_0, b_0, \alpha^0)}{\partial \boldsymbol{w}} &= \mathbf{0} \\ \frac{\partial \mathcal{L}(\boldsymbol{w}_0, \xi_0, b_0, \alpha^0)}{\partial \xi} &= \mathbf{0} \end{aligned} \tag{69}$$

Novamente, após satisfeitas as condições do teorema Kühn-Tucke, a forma dual do problema de otimização é obtida, de acordo com a Equação (70).

$$\begin{aligned} \text{Maximizar} \quad & \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j}^{l} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) \\ \text{sujeito a:} \quad & \\ & \sum_{i=1}^{l} \alpha_i y_i = 0, \\ & 0 \leq \alpha_i \leq C, \;\; i = 1, \ldots, l \end{aligned} \tag{70}$$

Como discutido no caso da classificação considerando o conjunto de dados linearmente separável, os coeficientes do hiperplano são identificados pela Equação (65). Estes pontos $\boldsymbol{x}_i$, para os quais o multiplicador de Lagrange $\alpha_i \neq 0$, atendem precisamente o critério $y_i[(\boldsymbol{w}^T \boldsymbol{x}_i) - b] \geq 1 - \xi_i$ contida na formulação primal do problema.

A.4- Funções kernel

O mapeamento do dados de entrada x em um espaço de características de elevada dimensão Z é outro recurso utilizado para tratar conjunto de dados que não são linearmente separáveis [Vapnik, 2013]. Como ilustrado na Figura 50, a representação bidimensional do conjunto de dados em (a) não possibilita a construção de um hiperplano para separar as classes. Contudo, se a dimensão de representação desses dados por tridimensional, torna-se possível determinar um hiperplano que separa corretamente estes dados em (b).

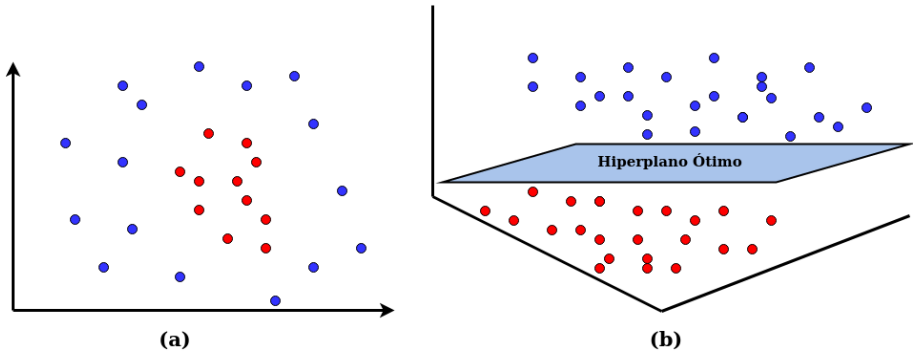

Figura 50 – (a) Representação bidimensional dos dados não-linearmente separáveis; (b) Transformação da representação dos dados para o espaço tridimensional, quando torna-se possível a separação linear pelo hiperplano.

Este mapeamento $\Re^2 \longrightarrow \Re^3$ é expresso como um produto interno no espaço de Hilbert, de acordo com a Equação (71).

$$(z_i \cdot z_j) = K\langle x_i, x_j \rangle \tag{71}$$

onde z é a imagem no espaço de características do vetor de dados de entrada x.

A transformação $K\langle x_i, x_j \rangle$ pode ser qualquer função simétrica $\phi(\cdot)$ que deve cumprir condições necessárias e suficientes determinadas pelo teorema de Mercer, descrito nas Equações (72) e (73).

$$K\langle u, v \rangle = \sum_{k=1}^{\infty} \alpha_k \phi_k(u) \phi_k(v) \tag{72}$$

com a condição necessária e suficiente:

$$\int\int K(u,v)g(u)g(v)dudv > 0, \quad \forall g \tag{73}$$

válida para todo $g(\cdot) \neq 0$, onde $\int g^2(u)du < \infty$.

Reescrevemos a Equação (66) com a inclusão da função kernel $K\langle x_i, x_j\rangle$.

$$\begin{aligned} \text{Maximizar} \quad & \sum_{i=1}^{l} \alpha_i - \frac{1}{2}\sum_{i,j}^{l} \alpha_i\alpha_j y_i y_j K\langle x_i, x_j\rangle \\ \text{sujeito a:} \quad & \\ & \sum_{i=1}^{l} \alpha_i y_i = 0, \\ & 0 \leq \alpha_i \leq C, \quad i = 1, \ldots, l \end{aligned} \tag{74}$$

Diferentes funções $\phi(\cdot)$ podem ser implementadas na transformação:

- Linear: $K(x_i, x_j) = (x_i \cdot x_j)$;
- Gaussiana: $K(x_i, x_j) = \exp\left(-\frac{||x_i - x_j||^2}{2\sigma^2}\right)$, onde σ é o desvio padrão ;
- Polinomial: $K(x_i, x_j) = (x_i \cdot x_j)^p$, onde p é a ordem do polinômio;
- RBF: $K(x_i, x_j) = exp\left(-\gamma||x_i - x_j||^2\right)$, onde $\gamma = \frac{1}{2\sigma^2}$ e σ é o desvio padrão;
- entre outras.

A aplicação de funções kernel além da linear permite a construção de fronteiras de classes não-lineares, o que amplia a capacidade de generalização das máquinas de vetor de suporte.

B- Cadeias de Markov

Algumas propriedades da cadeia de Markov serão enunciadas. A principal propriedade diz respeito ao próximo estado do processo $q_{t+1} = S_j$ ser dependente somente do estado atual $q_t = S_i$, o que confere à cadeia de Markov a característica de ser um processo de tempo discreto sem memória, ao passo que durante a sua evolução o passado é desprezado [Ross, 2002]. A Equação (75) formaliza esta propriedade.

$$a_{ij}(t) = P(q_{t+1} = S_j | q_t, \ldots, q_1) = P(q_{t+1} = S_j | q_t = S_i) \tag{75}$$

onde $t = 1, 2, \ldots$ e $\sum_j a_{ij}(t) = 1$, assim cada linha da matriz de transição soma 1.

De acordo com Zucchini et al. [2017], as cadeias de Markov podem ser compreendidas como um um primeiro nível de relaxamento em direção à independência condicional. Outra propriedade relevante diz sobre a transição entre os estados ser independente em relação ao tempo, o que determina a cadeia de Markov homogênea. Para formalizar esta propriedade, nos apropriamos da Equação (75) e ressaltamos que (75) se refere à transição de passo unitário entre estados [Cassandras and Lafortune, 2009]. Naturalmente, podemos estender esta noção para transições que ocorrem em n-passos, de acordo com a Equação (76).

$$a_{ij}(t, t+n) = P(q_{t+n} = S_j | q_t = S_i) \tag{76}$$

Podemos assumir, baseado na Equação (76), que existe algum estado intermediário k durante a transição do estado S_i para o estado S_j, dada a possível distância entes estes passos determinada pelo valor de n. A Equação (77) desenvolve este raciocínio utilizando a regra da probabilidade total [Cassandras and Lafortune, 2009].

$$a_{ij}(t, t+n) = \sum_{r=1}^{R} P(q_{t+n} = S_j | q_u = S_r, q_t = S_i) P(q_u = S_r | q_t = S_i) \tag{77}$$

onde $t \leq u \leq t + n$.

A Figura 51 expressa a relação discutida na Equação (77). Note que o somatório

$\sum_{r=1}^{R} P(q_{t+n} = S_j | q_u = r, q_t = S_i)$ contempla todas as R transições intermediárias viáveis em algum passo u.

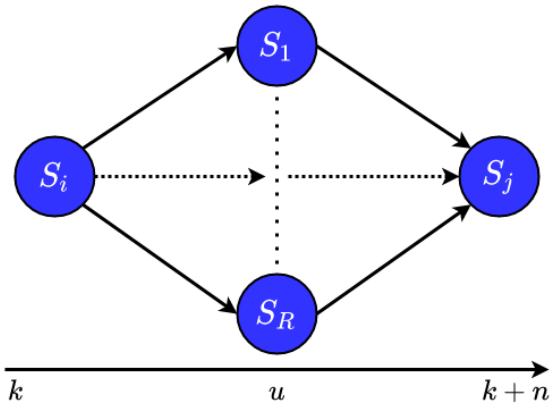

Figura 51 – Transições intermediárias que podem ocorrer entre um estado i e j. Em algum passo u, há R estados intermediários contemplados na Equação (77).

Pela aplicação da propriedade da perda de memória apresentada em (75), obtemos a partir de (77) a Equação geral de Chapman-Kolmogorov (78), de acordo com Cassandras and Lafortune [2009].

$$a_{ij}(t, t+n) = \sum_{r=1}^{R} a_{ir}(t, u) a_{rj}(u, t+n), \quad t \leq u \leq t+n \tag{78}$$

Em uma cadeia de Markov homogênea, a Equação (76) é reescrita como:

$$a_{ij}^{n}(t, t+n) = P(q_{t+n} = S_j | q_t = S_i), \quad n = 1, 2, \ldots \tag{79}$$

Baseado na Equação (79), estendemos a Equação de Chapman-Kolmogorov (78) para o caso homogêneo. Fazendo $u = k + m$ e posteriormente $m = n - 1$, obtemos:

$$a_{ij}^{n} = \sum_{r=1}^{R} a_{ir}^{n-1} a_{rj} \tag{80}$$

Ou, em notação matricial:

$$H(n) = H(n-1)H(1) \tag{81}$$

onde $H(n) \equiv [p_{ij}^{n}]$.

Vamos avaliar a transformação dos valores de probabilidade dos estados $\pi_j(t) \equiv P[q_t = S_j]$ após um tempo [Cassandras and Lafortune, 2009]. Esta é uma das principais

aplicações da cadeia de Markov. Seja o vetor de estados $\boldsymbol{\pi(t)}$ no instante t, como vemos na Equação (82).

$$\boldsymbol{\pi(t)} = [\pi_1(t), \pi_2(t), \ldots, \pi_j(t)],\ t = 1, 2, \ldots, T \tag{82}$$

Para determinar os valores futuros para $t > 0$, a partir de uma distribuição inicial $\boldsymbol{\pi(0)}$, utilizamos a Equação de kolmogorov (78). Seja $\pi_k(t+1)$ um elemento do vetor $\boldsymbol{\pi(t)}$, cuja probabilidade do evento condicional $P(q_{t+1} = S_j | q_t = S_i)$ para todos os valores possíveis de i é dada por:

$$\begin{aligned} \pi_j(t+1) = P(q_{t+1} = S_j) &= \sum_i P(q_{t+1} = S_j | q_t = S_i) \cdot P(q_t = S_i) \\ &= \sum_i a_{ij} \cdot \pi_i(t) \end{aligned} \tag{83}$$

Pela definição apresentada na Equação (83), alcançamos a solução utilizando a notação matricial:

$$\boldsymbol{\pi}(t) = \boldsymbol{\pi}(0)\boldsymbol{A}^t,\ t = 1, 2, \ldots \tag{84}$$

onde $\boldsymbol{A}$ é a matriz de transição.

O que acontece quando impomos $\boldsymbol{\pi}_j = \lim_{t \to \infty} \boldsymbol{\pi}(t)$? Nesta situação, dadas algumas condições que discutiremos a seguir, é atingida a distribuição estacionária, quando as probabilidades dos estados não variam mais sob a influência da passagem do tempo. Discutimos a seguir algumas propriedades da cadeia de Markov que devem ser atendidas para que o estado estacionário seja viável, de acordo com Cassandras and Lafortune [2009]:

- Alcançabilidade de estados: um estado j é alcançável a partir de um estado i se $a_{ij}^n > 0$ para algum $n = 1, 2, \ldots$;
- Clausura do subconjunto de estados: um subconjunto $\mathrm{S} \subset \boldsymbol{S}$ é fechado se $a_{ij} = 0$ para qualquer $i \in \mathrm{S},\ j \notin \mathrm{S}$. Ou seja, não ocorrem transições de entre qualquer estado a_{ij} pertencente ao subconjunto S e quaisquer outros estados fora de S;
- Estados absorventes: um estado i é absorvente se ele próprio forma um subconjunto fechado. Assim, se um estado é absorvente, temos $a_{ij} = 1$;

- Irredutibilidade da cadeia de Markov: a cadeia de Markov é irredutível se cada estado pode ser alcançado por qualquer outro estado. A cadeia de Markov é redutível se existe ao menos um conjunto fechado de estados;

- Recorrência de estados: seja ρ_i^n a probabilidade do primeiro retorno para o estado i após exatamente n passos $(n \geq 1)$. Sendo a probabilidade de sempre retornar para o estado i igual a $\rho_i = \sum_{n=1}^{\infty} \rho_i^{(n)}$, o tempo médio de recorrência é determinado por $M_i = \sum_{n=1}^{\infty} \rho_i^{(n)}$. Com isso, o estado i é transiente se a probabilidade de retorno para este estado é < 1 (ou $\rho_i < 1$); em contraste, o estado i é denominado recorrente se $\rho_i = 1$. Concluindo, um estado i é recorrente positivo se seu tempo médio de recorrência é finito (ou $M_i < \infty$); e o estado i é nulamente recorrente se o seu tempo médio de recorrência é infinito (ou $M_i = \infty$);

- Periodicidade de estados: um estado i é periódico se o seu período $d(i)$ é o maior divisor comum do conjunto $\{n > 0 : a_{ii}^n > 0\}$.

Agora, podemos continuar com a discussão sobre a estacionariedade dos estados da cadeia de Markov. Um teorema estabelecido por Kolmogorov assegura que em cadeias de Markov irredutíveis, aperiódicas e homogêneas o limite

$$\boldsymbol{\pi}_j = \lim_{t \to \infty} \boldsymbol{\pi}(t) \tag{85}$$

sempre existe e independe do vetor $\boldsymbol{\pi}(\boldsymbol{t})$, $t = 1, 2, \ldots$ utilizado para determinar o estado inicial [Cassandras and Lafortune, 2009].

Este teorema garante a existência do limite de $\boldsymbol{\pi}_j$, mas não legitima a existência de uma distribuição de probabilidade de estado estacionária, quando $\sum_j \boldsymbol{\pi}_j = 1$. Assim, especificamos condições complementares que asseguram ou não a existência da estacionariedade para a cadeia de Markov [Cassandras and Lafortune, 2009]. Em uma cadeia de Markov irredutível e aperiódica na qual todos os estados são transientes ou nulamente recorrentes, que atende a igualdade

$$\boldsymbol{\pi}_j = \lim_{t \to \infty} \boldsymbol{\pi}(t) = 0, \ \forall j, \tag{86}$$

a distribuição de probabilidade de estado estacionária não existe. Por sua vez, em uma cadeia de Markov irredutível e aperiódica, na qual todos os estados são recorrentes positivos, existe um único vetor de probabilidade de estado estacionário $\boldsymbol{\delta}$, tal que $\boldsymbol{\pi}_j > 0$

e

$$\delta_j = \lim_{t \to \infty} \boldsymbol{\pi}(t) = \frac{1}{M_j}, \tag{87}$$

onde M_j é o tempo médio de recorrência do estado j.

Assim, este teorema assegura que, atendidas as condições, o cálculo do vetor de distribuição de estados estacionária $\boldsymbol{\delta}$ consiste na solução de um problema com equações lineares, de acordo com a Equação (88) [Cassandras and Lafortune, 2009].

$$\begin{aligned} \boldsymbol{\delta} &= \boldsymbol{\delta} A \\ \sum_j \boldsymbol{\pi}_j &= 1 \end{aligned} \tag{88}$$