



Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Departamento de Engenharia Elétrica

Trabalho de Conclusão de Curso

Desenvolvimento de modelos de predição de falhas em
turbogeradores por meio da análise multivariada de
séries temporais

Marianna Barbosa Brito Dias

Campina Grande, Paraíba, Brasil

©Marianna Barbosa Brito Dias, 22 de outubro de 2021

Marianna Barbosa Brito Dias

Desenvolvimento de modelos de predição de falhas em
turbogeradores por meio da análise multivariada de
séries temporais

Trabalho de Conclusão de Curso submetida à Co-ordenadoria de Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande como parte dos requisitos necessários para obtenção do grau de Bacharel em Ciências no Domínio da Engenharia Elétrica.

Orientador: Prof. George Rossany Soares de Lira, Dr.

Campina Grande, Paraíba, Brasil

©Marianna Barbosa Brito Dias, 22 de outubro de 2021

Marianna Barbosa Brito Dias

Desenvolvimento de modelos de predição de falhas em
turbogeradores por meio da análise multivariada de
séries temporais

Trabalho de Conclusão de Curso de Bacharelado
submetida à Coordenadoria de Graduação em En-
genharia Elétrica da Universidade Federal de Cam-
pina Grande como parte dos requisitos necessários
para obtenção do grau de Bacharel em Ciências no
Domínio da Engenharia Elétrica.

Aprovado em ___/___/___

Prof. George Rossany Soares de Lira, Dr.

Orientador

Prof. Edson Guedes da Costa, Dr.

Avaliador

Campina Grande, Paraíba, Brasil

©Marianna Barbosa Brito Dias, 22 de outubro de 2021

Agradecimentos

Agradeço, primeiramente, a Deus, por ter me concedido saúde e disposição para enfrentar essa longa jornada da graduação. Também sou grata a Ele por estar comigo em todos os momentos difíceis, nos quais pensei em desistir.

Meus agradecimentos à minha família, em especial, aos meus pais, por todo apoio, amor, preocupação e por todo o esforço para proporcionar o melhor que podiam à mim e aos meus irmãos. Reconheço toda a luta e empenho para que eu tivesse a melhor educação possível, e, certamente sem o apoio deles, eu não estaria aqui hoje.

Agradeço ao meu namorado Victor, por compartilhar comigo os meus momentos mais felizes, mas também por me dar todo o apoio e confiança para superar os momentos tristes. Agradeço por sempre ter ficado do meu lado, por ter entendido as minhas fraquezas e instabilidades. Obrigada por todos os momentos de alegria genuína e pelos seus abraços reconfortantes que me cobrem por inteira.

Sou grata a Maxwell, por toda ajuda concedida ao longo da graduação, compartilhando seu conhecimento e tempo. Agradeço a todos os meus amigos, em especial a Ana Paula, André, Artur, Clara, Débora, Mirelle, Pedro, Sara, Samuel, Stayner, Vagne, Victor e Vinícius, por serem um refúgio nos momentos de dificuldade e me fazerem uma pessoa mais feliz. Obrigada por toda ajuda e por todos os momentos que vivemos juntos, levarei essas memórias para sempre comigo.

Agradeço a Vandilson, por todo o conhecimento compartilhado no Laboratório de Alta Tensão. Também sou grata a Patrício, por sua colaboração e orientação, sem as quais não seria possível realizar este trabalho.

Por fim, agradeço aos professores que contribuíram direta ou indiretamente para minha formação. Em especial ao Professor George Lira, por todas as experiências compartilhadas, por ser meu orientador durante tantos anos e me dar a oportunidade de enfrentar novos desafios. Agradeço também ao Professor Edson Costa, pelo apoio e dedicação com os alunos.

Resumo

Este trabalho trata da documentação de um estudo sobre análise de séries temporais multivariadas aplicadas a modelos de classificação. Como estudo de caso, foi desenvolvido um modelo de predição de falhas em turbogeradores de um sistema de geração isolado. Para isso, foram utilizados dados operacionais e registros de monitoramento de um dos turbogeradores do sistema. Esses dados passaram por uma etapa de pré-processamento, na qual é realizada uma limpeza nos dados e seleção dos atributos mais relevantes para o desenvolvimento do modelo de detecção de falhas. Ao longo do trabalho foi implementado um modelo de previsão de séries temporais multivariadas baseado em VAR (do inglês *Vector AutoRegression*) para constituir uma base de dados artificial de modo a contribuir com questões de desbalanceamento na detecção de falhas. Já o classificador foi desenvolvido com o uso de um modelo de aprendizado de máquina baseado redes neurais recorrentes do tipo LSTM (do inglês *Long Short Term Memory*). Os testes realizados até o momento indicam que as técnicas implementadas apresentam resultados satisfatórios na detecção de falhas do turbogerador, contribuindo no processo de gerenciamento de ativos.

Palavras-chave: Séries temporais multivariadas, turbogeradores, detecção de falhas, *Vector AutoRegression*, *Long Short Term Memory*.

Abstract

This report presents the documentation of a study on multivariate time series classification models. As a case study, a failure prediction model in turbogenerators of an isolated generation system was developed. For this, operational data and monitoring records from one of the system's turbogenerators were used. This data went through a pre-processing stage, in which the data was cleaned and the model's most relevant attributes were selected. Throughout the work, a multivariate time series prediction model based on VAR (Vector AutoRegression) was implemented to constitute an artificial database in order to contribute with unbalance issues in the failures detection. The fault classifier was developed using a machine learning model based on a special kind of recurrent neural networks called LSTM (Long Short Term Memory). The tests carried out so far indicate that the implemented techniques present satisfactory results in the detection of turbogenerator failures, contributing to the asset management process.

Keywords: Multivariate time series, Failure detection, Vector AutoRegression, Long Short Term Memory.

Sumário

Lista de símbolos e abreviaturas	iv
Lista de Tabelas	vii
Lista de Figuras	viii
1 Introdução	1
1.1 Objetivo geral	3
1.2 Objetivos específicos	3
1.3 Organização do documento	3
2 Sistema de geração de uma FPSO	5
2.1 Contextualização histórica	5
2.2 Unidade de geração de uma FPSO	6
2.3 Considerações finais	10
3 Pré-processamento de dados	11
3.1 Análise de dados	11
3.1.1 Limpeza de dados	12
3.1.2 Dados faltantes	13
3.1.3 Seleção de variáveis	14
3.2 Considerações finais	16
4 Previsão de séries temporais multivariadas	18
4.1 Propriedade das séries temporais	18
4.2 Modelo VAR	19

4.2.1	Formulação matemática do modelo VAR	20
4.2.2	Estimação do modelo VAR	21
4.2.3	Seleção de defasagem do modelo VAR	21
4.3	Considerações finais	22
5	<i>Deep Learning</i> em problemas de classificação	23
5.1	Classificação de Séries Temporais	23
5.1.1	Redes neurais recorrentes	25
5.1.2	LSTM	27
5.1.3	Uso de CPU vs GPU em modelos de <i>Deep Learning</i>	30
5.2	Considerações finais	31
6	Metodologia de pesquisa	32
6.1	Sistema Elétrico da FPSO	32
6.2	Pré-processamento	34
6.3	Modelo de detecção de falhas	37
6.4	Considerações finais	39
7	Resultados e análises	41
7.1	Pré-processamento	41
7.2	Modelo de detecção de falhas	53
7.3	Considerações finais	57
8	Conclusão	58
8.1	Trabalhos futuros	59
	Referências bibliográficas	60

Lista de símbolos e abreviaturas

AIC	<i>Akaike</i>	22
AR	<i>Autoregressive</i>	19
BIC	<i>Schwarz-Bayesian</i>	22
CPU	<i>Graphics Processing Unit</i>	30
CST	<i>Classificação de Séries Temporais</i>	24
DL	<i>Deep Learning</i>	23
DM	<i>Data Mining</i>	3
DS	<i>Data Science</i>	3
EDA	<i>Exploratory Data Analysis</i>	11
FN	<i>False Negative</i>	39
FP	<i>False Positive</i>	39
FPR	<i>False Positive Rate</i>	57
FPSO	<i>Floating Production Storage and Offloading</i>	3
GPU	<i>Graphics Processing Unit</i>	4
HQ	<i>Hannan-Quinn</i>	22
LSTM	<i>Long Short-Term Memory</i>	4
MAPE	<i>Mean Absolute Percentage Error</i>	52
MIMD	<i>Multiple Instruction, Multiple Data</i>	31
MISD	<i>Multiple Instruction, Single Data</i>	31

ML	<i>Machine Learning</i>	3
MSE	<i>Mean Squared Error</i>	38
PST	Previsão de Séries Temporais	18
SIMD	<i>Single Instruction, Multiple Data</i>	31
SISD	<i>Single Instruction, Single Data</i>	31
ST	Séries Temporais	3
TN	<i>True Negative</i>	39
TP	<i>True Positive</i>	39
TPR	<i>True Positive Rate</i>	57
UPS	<i>Data Science</i>	8
VAR	<i>Vector AutoRegression</i>	4

Lista de Tabelas

6.1	Causas das falhas registradas no turbogerador em análise.	33
6.2	Matriz de confusão genérica.	38
7.1	Sensores seleccionados ao final no processo de seleção de atributos.	44
7.2	Avaliação quantitativa do modelo de detecção de falhas	53

Lista de Figuras

2.1	Modelo de plataforma tipo FPSO.	7
2.2	Principais subsistemas de uma plataforma tipo FPSO.	7
2.3	Diagrama unifilar da FPSO.	9
3.1	Fluxograma de coleta de dados à tomada de decisão.	12
3.2	Elementos de um diagrama de caixa.	13
3.3	Comparação entre o gráfico de caixa e o de violino.	13
3.4	Mapa de calor expressando a correlação de Pearson entre quatro variáveis genéricas.	16
5.1	Fluxograma do processo de classificação de séries temporais	25
5.2	Comparação das arquiteturas entre uma rede neural do tipo <i>Feedforward</i> e do tipo Recorrente.	26
5.3	Arquitetura LSTM original	29
5.4	Arquitetura LSTM com <i>Input Gate</i> , <i>Forget Gate</i> o <i>Output Gate</i>	29
6.1	Etapa de pré-processamento do estudo de análise multivariada de séries temporais para detecção de falhas em turbogeradores de uma plataforma FPSO.	35
6.2	Fluxograma do modelo de detecção de falhas em turbogeradores de uma plataforma FPSO utilizando LSTM.	40
7.1	Informação mútua entre os 55 sensores e o vetor de falhas.	42
7.2	Seleção dos sensores com informação mútua acima do valor <i>threshold</i>	43
7.3	Matriz de correlação linear entre os 32 sensores pré-selecionados.	46
7.4	Matriz de correlação linear entre os sensores selecionados.	47

7.5	Gráfico de violino para cada um dos sensores selecionados	48
7.6	Leitura dos sensores para uma instância da base de dados artificial	49
7.7	MAPE (%) referente a previsão do modelo VAR para cada sensor	52
7.8	Acurácia (a) e MSE (b) para os dados de treinamento ao longo das épocas .	54
7.9	Acurácia (a) e MSE (b) para os dados de validação ao longo das épocas . . .	55
7.10	Matriz de confusão com os valores de teste do modelo de detecção de falhas.	56
7.11	Curva ROC do modelo	56

Capítulo 1

Introdução

A garantia do fornecimento de energia se torna imprescindível no setor elétrico e de petróleo. Assegurar a continuidade operacional dos equipamentos é de grande relevância, pois tem relação direta com as necessidades dos mercados e consumidores. Ademais, a interrupção do fornecimento de energia leva a aplicação de multas onerosas pelas agências reguladoras.

De acordo com Castro et al.[1], plataformas de produção de petróleo e gás tipo FPSO (do inglês *Floating Production Storage and Offloading*) realizam produções superiores a 150.000 barris de petróleo por dia. Essas plataformas apresentam um lucro diário médio de US\$ 6.000.000,00 [2]. Tomando por base esse cenário, o custo de interrupção das atividades dessas plataformas é muito oneroso. Nesse sentido, identificar fragilidades operacionais no sistema torna-se uma questão cada vez mais relevante.

Em uma FPSO existem subsistemas de compressão e de processamento de gás, de processamento de óleo, de tratamento e de produção de água, de geração de energia, de utilidades, de tocha de segurança (*flare*), de controle e automação e de ancoragem dos *riseres*. Todos subsistemas dependem da energia gerada pelo sistema de geração de energia [3]. O sistema de geração de energia é composto por um conjunto de 3 ou 4 turbogeradores, que produzem eletricidade a partir da combustão de gás natural associado ou mesmo de diesel.

Os turbogeradores são constantemente monitorados através de instrumentos de medição, os quais são chamados de sensores ou medidores. Um sensor é um elemento transdutor que recebe como entrada uma grandeza física (vazão, pressão, temperatura, etc.) e fornece um

sinal elétrico, analógico ou digital, com valor proporcional ao da grandeza medida [3]. Estes instrumentos são utilizados para monitorar uma série de variáveis, dentre elas, temperatura, pressão, chama, etc. As medições advindas destes sensores constituem exemplos de séries temporais (STs).

As séries temporais são utilizadas em vários campos de estudos nas mais diversas áreas desde previsões meteorológicas até medições de sinais psicológicos. Uma série temporal consiste em uma sequência de leituras de dados igualmente espaçadas no tempo.

Nas ultimas décadas, houve um aumento significativo de dados de série temporal coletados por meio de sensores. Um conjunto de dados de série temporal é considerado univariado quando a sequência de medições é realizada sobre a mesma variável. Já quando são coletadas medições de múltiplas variáveis, ou sensores, são constituídas séries temporais multivariadas. Ultimamente, a análise de séries temporais multivariadas nos processos de classificação de falhas vem ganhando cada vez mais destaque [4].

O processo de classificação de falhas em equipamentos por meio de uma abordagem multivariada pode contribuir na construção de um modelo que represente melhor a dinâmica do equipamento, uma vez que será analisada a influência de múltiplos fatores na detecção do problema. Huang e Yoo[4] destacam que a análise multivariada pode ter vantagens práticas relevantes. Uma vez que se tenha um número de eventos de falha relacionados a dados temporais multivariados, pode-se extrair os atributos mais relevantes para a falha, além de poder identificar padrões discriminativos antes dos eventos ocorrerem.

A análise de falhas em máquinas é um processo que pode se beneficiar significativamente de técnicas de mineração de dados (DM) e aprendizagem de máquina (ML) [4]. O DM consiste no processo de coleta, identificação e extrações de informações, como padrões ou correlações em grandes conjuntos de dados, para que com isso possamos antecipar e definir resultados. ML consiste em um conjunto de técnicas baseadas na ideia de que máquinas são capazes de assimilar dados por conta própria. Com isso, padrões de dados e conexões entre eles seriam rapidamente identificados. A aplicação de técnicas de análise preditiva pode auxiliar nas tomadas de decisão de uma empresa, auxiliando na identificação de falhas dispendiosas. Dessa forma, busca-se minimizar a redução nos lucros devido as perdas de produção.

De modo geral, é presumido que existam padrões discriminativos que ocorrem antes dos

eventos de falha [4]. Conhecer esses padrões não apenas ajuda a encontrar as causas raízes da falha, mas também torna possível a detecção precoce no futuro. Todavia, extrair esses padrões a partir da leitura de um grande volume de dados coletados de múltiplos sensores ainda é um grande desafio. Assim, pretende-se, no trabalho, desenvolver um modelo capaz de identificar padrões e projetar um sistema acurado de classificação da ocorrência de falhas tomando como estudo de caso sensores em turbogeradores de um sistema de geração de uma FPSO.

1.1 Objetivo geral

Desenvolvimento de modelos baseados em ML e *Data Science* (DS) com o intuito de identificar possíveis falhas em turbogeradores instalados em uma FPSO. Para isso, será realizada uma análise multivariada proveniente de dados advindos de sensores de monitoramento do sistema de geração da plataforma energética analisada.

1.2 Objetivos específicos

Para a realização do objetivo geral, são definidos os seguintes objetivos específicos:

- Realizar o levantamento do estado da arte de modelos de detecção de falhas baseados em séries temporais multivariadas;
- Desenvolver uma análise exploratória dos dados de modo a selecionar os atributos mais influentes para alimentar os modelos de previsão;
- Implementar algoritmos de ML para a detecção da ocorrência de falhas em turbogeradores de uma FPSO;
- Quantificar as métricas de desempenho dos modelos de previsão desenvolvidos.

1.3 Organização do documento

O Capítulo 2 trata do detalhamento de uma FPSO, destacando seus principais componentes e subsistemas.

O Capítulo 3 traz uma explicação de técnicas utilizadas no pré-processamento dos dados brutos, deste a limpeza dos dados, seleção de atributos, análise de correlação, entre outros.

O Capítulo 4 aborda o modelo VAR (do inglês *Vector AutoRegression*), um dos algoritmos mais utilizados na previsão de valores futuros de séries temporais multivariados.

O Capítulo 5 trata da explicação teórica do modelo de ML baseado em LSTM (do inglês *Long Short-Term Memory*), que será implementado na detecção de falhas do turbogeradores. São detalhados o funcionamento do algoritmo, bem como a melhoria do seu desempenho junto ao uso de GPUs (do inglês *Graphics Processing Unit*) quando se trabalha de um grande volume de dados.

O Capítulo 6 trata da metodologia utilizada desde o pré-processamento dos dados até o desenvolvimento do modelo.

O Capítulo 7 aborda os resultados obtidos por meio da implementação do modelo LSTM utilizado na detecção de falhas em turbogeradores de uma FPSO, bem como análise desses resultados, por meio do cálculo de métricas de desempenho.

Por fim, o Capítulo 8 trata das conclusões deste estudo e das contribuições desta documentação para futuros trabalhos.

Capítulo 2

Sistema de geração de uma FPSO

Este capítulo trata do detalhamento da unidade de geração de uma plataforma FPSO, destacando a importância da mesma para a plataforma.

2.1 Contextualização histórica

Existem diversos tipos de unidade de produção de petróleo, sendo cada uma delas utilizadas de acordo com as necessidades e características que cada área de operação exige [5].

As primeiras descobertas de depósitos de petróleo em regiões marinhas ocorreram nos Estados Unidos no final do século XX, levando a instalação de plataformas oceânicas. As primeiras plataformas foram instaladas em águas rasas e a uma pequena distância da costa [6].

Com o avanço do processo de exploração de petróleo em águas cada vez mais profundas, surgiu uma demanda por novos equipamentos e técnicas que permitissem a pesquisa geológica, a perfuração de poços submarinos e a exploração dos mesmos, além das atividades de transporte do petróleo da unidade de produção até os locais de distribuição [7].

Entre as décadas de 70 e 80, a maioria das reservas submarinas apresentavam profundidades num limite máximo 140 metros. De modo geral, esse tipo de campo era explorado com sistemas fixos no fundo do mar, nos quais o petróleo geralmente era transportado para a costa através de dutos submarinos [7].

Com a exploração de petróleo em águas cada vez mais profundas e em posições muito

longe do continente ficou praticamente inviável a utilização de plataformas tipo fixas para extração de petróleo. A descoberta de petróleo em lâmina d'água superior a 1000 metros levou ao desenvolvimento de novas técnicas de exploração. Dessa forma, surgiram as plataformas flutuantes que são estruturas complacentes instaladas através de um sistema de ancoragem [8].

Na atualidade, grande parte da produção de petróleo e gás natural acontece em alto mar (também conhecido pelo termo em inglês *offshore*). Ademais, com as descobertas na área de pré-sal, esta operação tende a ser realizada em distâncias cada vez mais longas em relação ao continente, o que aumentaria bastante os custos de uma possível instalação de novos oleodutos e gasodutos submarinos. Com isso, as empresas do setor de petróleo e gás vêm se interessando cada vez mais pela aplicação de unidades marítimas flutuantes *offshore* [9].

Neste sentido, estes sistemas são utilizados preferencialmente quando o campo está alocado em regiões onde a instalação de dutos submarinos para condução de óleo até a costa não é economicamente viável, surgindo como uma estratégia interessante na solução deste problema a partir da conversão de navios com o casco simples em plataformas tipo FPSO. Isso possibilita que o óleo seja processado e armazenado para posteriormente ser escoado por meio de outro navio, conhecido como aliviador, que é periodicamente conectado a este para receber e transportar o óleo até os terminais petrolíferos [6]. A Figura 2.1 ilustra um modelo da plataforma FPSO em alto mar.

2.2 Unidade de geração de uma FPSO

A Figura 2.2 ilustra alguns subsistemas de uma plataforma tipo FPSO. São eles: os subsistemas de compressão e de processamento de gás, separação e tratamento, gás e água, de geração de energia, de utilidades e de tocha de segurança, mas conhecido como *flare*. Todos esses módulos são alimentados pela energia gerada no módulo de sistema de geração de energia, reiterando a importância desse módulo para o bom funcionamento da FPSO.

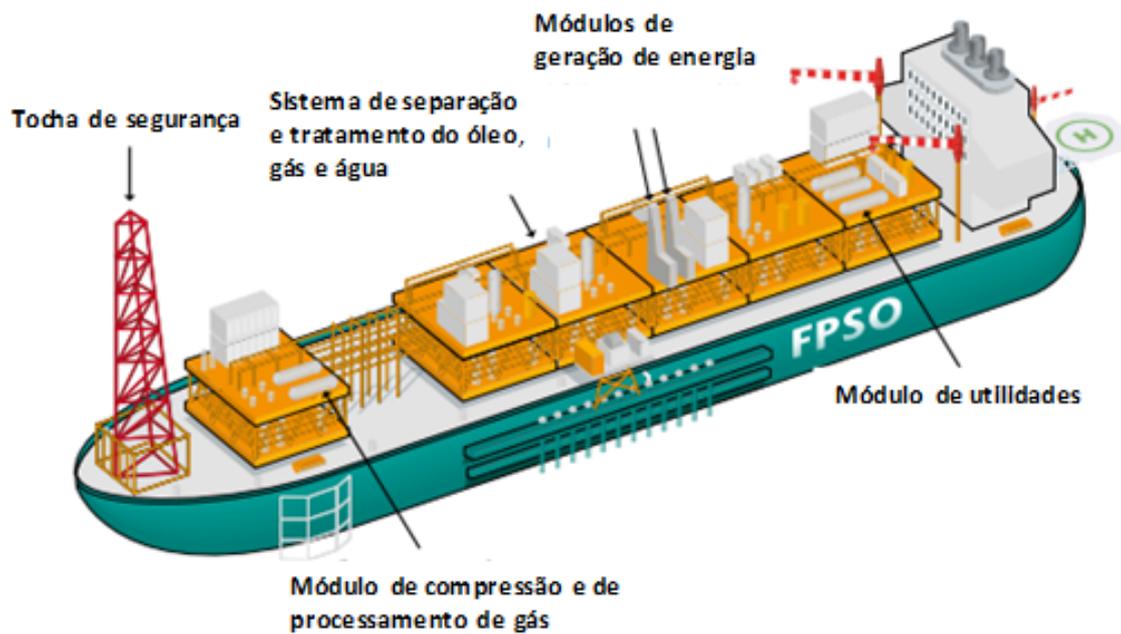
Os sistemas de geração de energia das plataformas tipo FPSOs configuram-se como unidades termelétricas flutuantes em alto mar, tendo geração de energia própria e independente para que seja assegurada a alimentação elétrica em todos os processos de produção da uni-

Figura 2.1: Modelo de plataforma tipo FPSO.



Fonte: Adaptado de SILVA[6]

Figura 2.2: Principais subsistemas de uma plataforma tipo FPSO.



Fonte: Adaptado de REMBE[10]

dade. De modo geral, num FPSO também existe um sistema de emergência para o caso de interrupção desta energia [9].

Em geral, o sistema elétrico de uma plataforma tipo FPSO, apresenta como fonte principal de energia um sistema de geração composto por três ou quatro turbogeradores, um gerador auxiliar e um gerador de emergência, além de uma fonte de alimentação ininterrupta, também conhecida como UPS (do inglês *Uninterruptible Power Supply*).

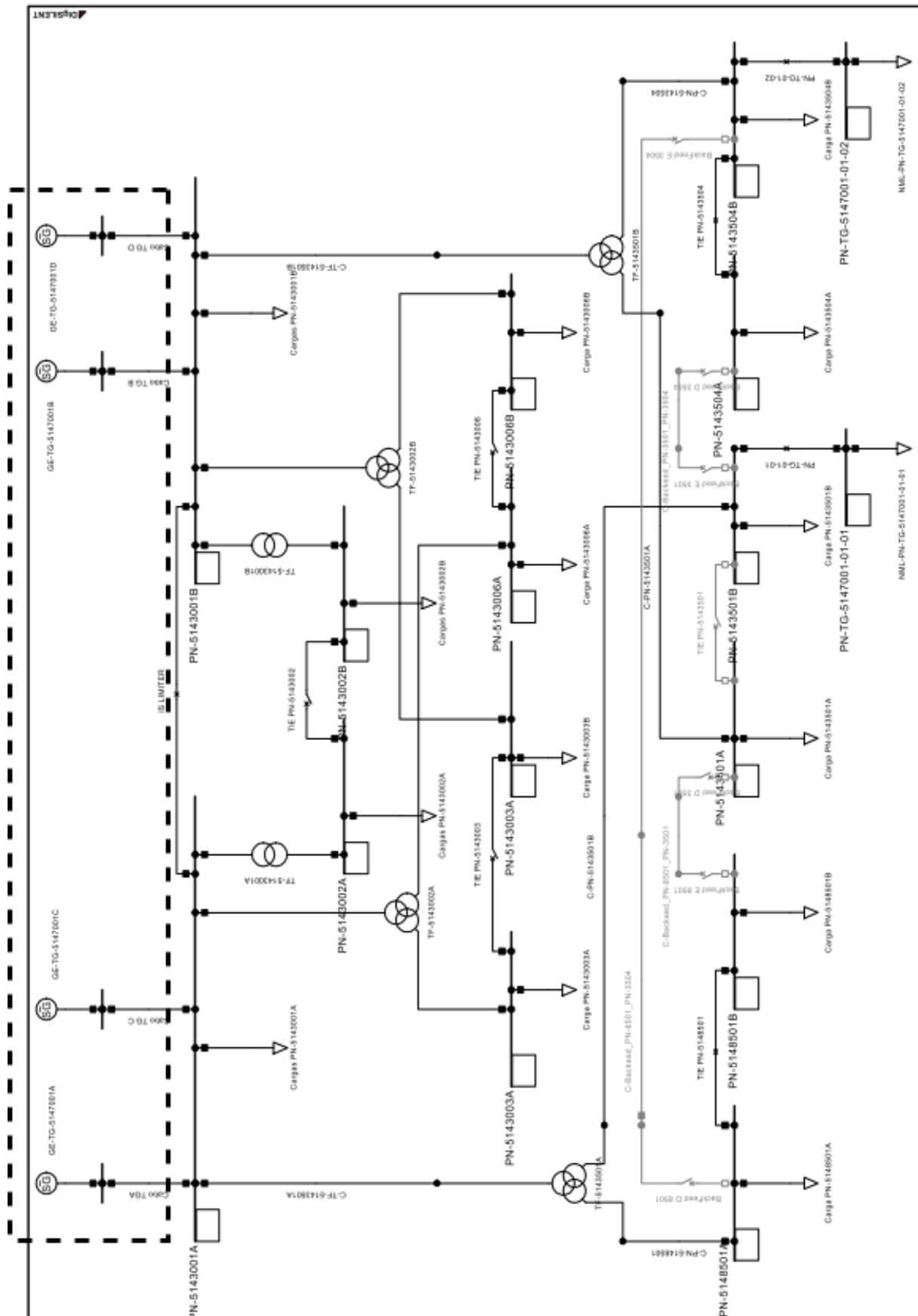
O turbogerador é uma máquina bastante pesada, cujo transporte e armazenamento não são simples. Logo, é difícil e custoso transportá-la até a plataforma, ainda mais para plataformas em águas profundas.

O sistema elétrico da plataforma é composto pelos sistemas do casco, que está relacionado a alimentação das cargas do navio e do *topside*, que está voltado para a alimentação de cargas relacionadas à produção de óleo, gás e injeção de água [1]. O funcionamento da unidade de produção, portanto, é totalmente dependente do módulo de geração de energia, sem o qual não os demais sistemas da plataforma tipo FPSO não conseguem realizar suas atividades, por isso a importância do monitoramento desse tipo de sistema e detecção de possíveis falhas [9].

O sistema de geração de energia elétrica principal dessa plataforma apresenta quatro turbogeradores (GE-TG-5147001A/B/C/D), cujo funcionamento ocorre por turbinas a gás, com potência nominal de 25MW/31,25kVA, tensão nominal de 13,8 kV e frequência nominal de 60 Hz. O diagrama unifilar do sistema elétrico do FPSO que será analisado neste trabalho é mostrado na Figura 2.3. Em tracejado é destacado o sistema de geração, composto por 4 turbogeradores.

Pretende-se realizar, neste trabalho, a predição de falhas tendo como objeto de estudo o turbogerador com o maior número de registro de falhas. Para isso, serão analisados dados operacionais e registros de sensores de monitoramento desse turbogerador. Diversos tipos de sensores serão analisados, dentre eles: sensor de chama, sensor diferencial de pressão de admissão de ar, sensor de pressão de gás combustível, sensor de pressão de óleo sintético, sensor de temperatura dos gases de exaustão, sensor de vibração do compressor, entre outros.

Figura 2.3: Diagrama unifilar da FPSO.



Fonte: Castro et al.[1]

2.3 Considerações finais

Neste capítulo foi destacado a evolução das unidades de produção de petróleo ao longo da história, apresentando as vantagens econômicas e práticas do sistema de plataformas flutuantes, em especial a FPSO diante da exploração de petróleo em águas cada vez mais profundas e em posições muito distantes do continente. Também foi destacada a importância do bom funcionamento do módulo de geração de energia em uma plataforma tipo FPSO, bem como foi descrito o sistema elétrico da unidade que será analisada.

Capítulo 3

Pré-processamento de dados

Este capítulo trata sobre a etapa de pré-processamento dos dados brutos coletados. Serão apresentadas algumas técnicas já conhecidas da análise de dados, como: limpeza de dados, correlação entre variáveis, extração de atributos e visualização de dados.

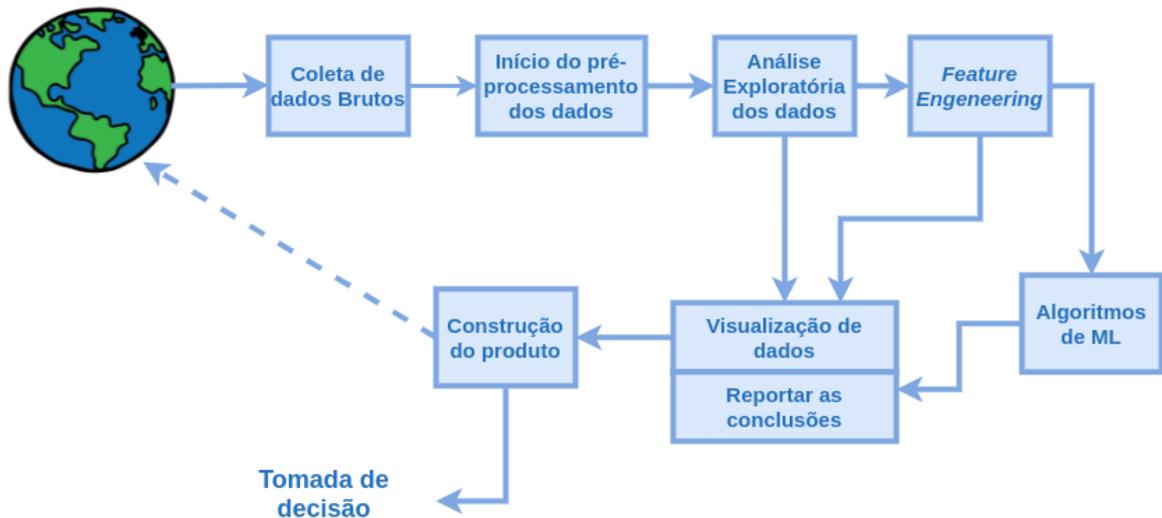
3.1 Análise de dados

A análise de dados é uma subdivisão da DS que se concentra na análise após o tratamento e limpeza de dados brutos, na etapa de pre-processamento, com o intuito de explorar o comportamento do conjunto de dados e a relação entre eles [11].

A análise exploratória de dados (EDA – do inglês *Exploratory Data Analysis*), como o próprio nome sugere, tem o foco na exploração do conjunto de dados, geralmente se aliando ao processo de visualização de dados para facilitar a compreensão de como as variáveis do conjunto se correlacionam e desenvolver ideias que auxiliem no processo de tomada de decisão. A Figura 3.1 ilustra o fluxograma desde a coleta de dados brutos, passando pela análise exploratória até a tomada de decisão.

Para realizar uma EDA, são feitas algumas etapas, como [11]:

- Avaliar graficamente a natureza das variáveis individuais;
- Realizar uma análise descritiva que permita quantificar alguns aspectos gráficos dos dados;

Figura 3.1: Fluxograma de coleta de dados à tomada de decisão.

Fonte: Autoria própria

- Examinar graficamente as relações entre as variáveis, além de quantificar o grau de inter-relação entre elas;
- Avaliar a presença de valores faltantes (*missing*);
- Preparar os dados para modelos estatísticos ou de ML;

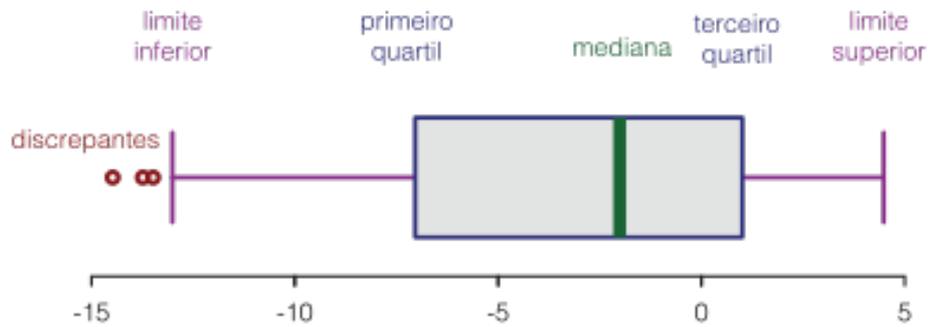
3.1.1 Limpeza de dados

Durante o processo de coleta de grandes quantidades de dados em um grande intervalo de tempo, podem ocorrer diversos problemas, como: erros de digitação; equipamentos de medição que podem falhar e problemas eletrônicos que podem levar a geração de dados surreais ou mesmo não gerar dado algum durante um certo número de observações.

Para o caso de variáveis contínuas, a aplicação de função que retorne como os dados estão distribuídos estatisticamente pode contribuir para identificar um valor não condizente com o coletado, sendo mais adequado o descarte desses valores. Assim, o uso de digramas de caixa pode ser bastante útil, conforme pode ser observado na Figura 3.2.

Um processo semelhante também pode ser feito por gráficos de violino. A principal ideia por trás desses gráficos é combinar as capacidades dos histogramas (ou gráficos de

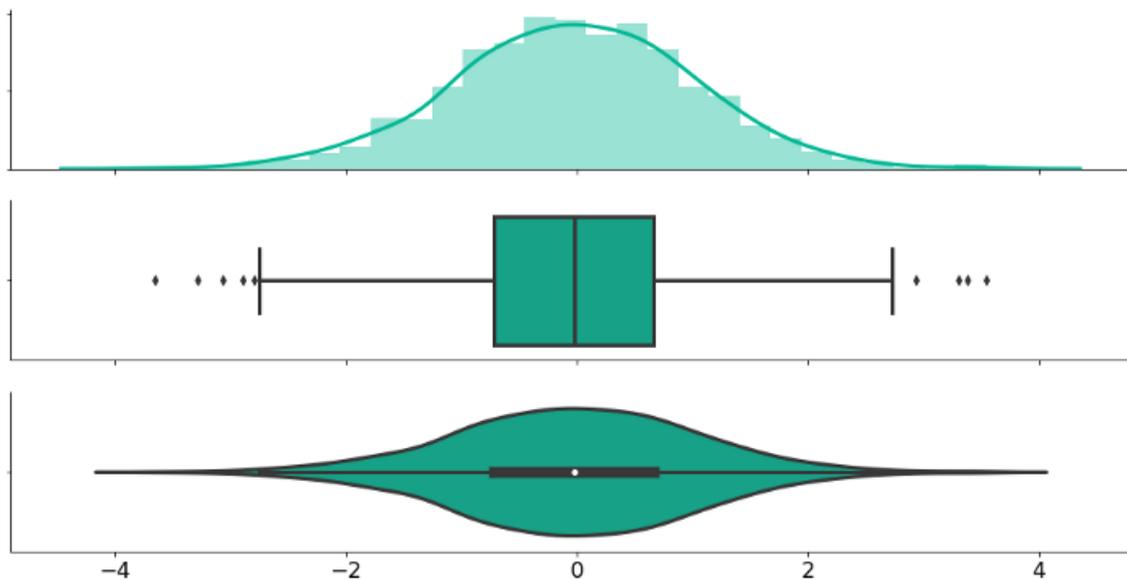
Figura 3.2: Elementos de um diagrama de caixa.



Fonte: Fonte: Radečić[12]

distribuição de *kernel*) e gráficos de caixa. A comparação entre esses gráficos é mostrado na Figura 3.3.

Figura 3.3: Comparação entre o gráfico de caixa e o de violino.



Fonte: Radečić[12]

3.1.2 Dados faltantes

Dados incompletos são um problema inevitável ao lidar com a maioria das fontes de dados do mundo real. Kotsiantis, Kanellopoulos e Pintelas[13] destacam que o especialista deve escolher entre uma série de métodos para lidar com dados faltantes, como:

- Desconsideração das instâncias de dados que apresentam pelo menos um valor de

atributo desconhecido;

- Substituição pelo valor do recurso mais comum, isto é, o valor do recurso que ocorre com mais frequência é selecionado para ser o valor para todos os valores desconhecidos do recurso;
- Substituição dos valores ausentes pela média dos valores nos demais casos;
- Substituição dos dados faltantes utilizando alternativas como métodos de interpolação calculados com os dados disponíveis;
- Utilização de modelos de ML capazes de lidar melhor com intervalos faltantes de dados, como a LSTM.

3.1.3 Seleção de variáveis

Em geral, instâncias de dados podem apresentar muitas *features* ou atributos. Todavia, as vezes apenas alguns deles podem estar relacionados ao problema em si. Pode haver redundância, na qual certos recursos são correlacionados de forma que não seja necessário incluir todos eles na modelagem; e interdependência, em que duas ou mais recursos transmitem informações fortemente dependentes entre si, de modo que não podem ser incluídos de maneira isolada[13].

O processo de seleção de atributos ou *feature selection* engloba atividades de identificação e remoção de informações irrelevantes e redundantes. Isso reduz a dimensionalidade dos dados e pode permitir que algoritmos de aprendizagem operem de forma mais rápida e eficaz.

Correlação entre atributos

Analisar a correlação entre *features* é de grande importância na análise de dados uma vez que auxilia para evitar o problema de colinearidade entre os atributos. Teoricamente, pode-se dizer que variáveis fortemente correlacionadas representam a mesma informação e, por isso, não é preciso utilizá-las em conjunto.

Além disso, a análise de correlação pode ser útil na identificação de variáveis que se explicam, contribuindo na substituição de valores faltantes em determinados atributos, por exemplo.

Em ciência de dados, a correlação entre atributos pode ser calculada pelo coeficiente de correlação de Pearson (r). Esse coeficiente mede a proximidade entre duas sequências de números, que podem ser listas, vetores ou colunas de um *DataFrame*, também conhecidas por séries.

O valor de r é um número entre -1 e 1 . Em resumo, esse coeficiente expressa numericamente se duas sequências de números (x e y) estão positivamente correlacionadas, não correlacionadas ou negativamente correlacionadas. Quanto mais próximo de 1 , mais forte é a correlação positiva. Quanto mais próximo de -1 , mais forte é a correlação negativa (ou seja, as sequências apresentam comportamento oposto). Quanto mais próximo de 0 , mais fraca é a correlação entre os atributos. A Correlação de Pearson é calculada conforme Equação 3.1.

$$r = \frac{\sum(x - \bar{x})(y - \bar{y})}{\sqrt{\sum(x - \bar{x})^2 \sum(y - \bar{y})^2}}. \quad (3.1)$$

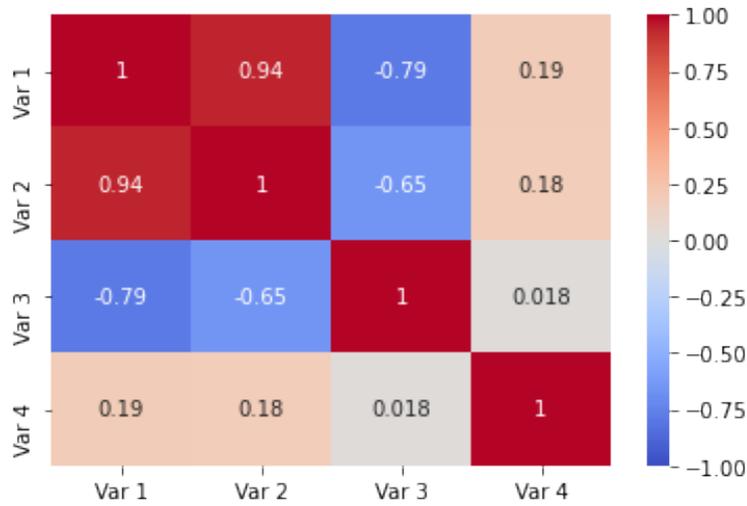
O cálculo de correlações é útil para ajudar a explorar um novo conjunto de dados, identificando aqueles atributos que são fortemente correlacionados e podem apresentar redundância de informação. Por meio de um mapa de calor dessas correlações, o cientista de dados consegue perceber visualmente quais são os atributos que apresentam as correlações mais fortes e aqueles que são fracamente correlacionados, conforme mostrado no exemplo da Figura 3.4.

Informação mútua

A informação mútua pode ser usada para avaliar a relação entre cada atributo e os valores desejados. De modo geral, pode-se dizer que a informação mútua é capaz de medir a dependência estatística entre variáveis, isto é, ela é capaz de medir “correlações não-lineares” gerais entre as variáveis. Se a informação mútua entre duas variáveis for 0 , então elas são estatisticamente independentes [14].

Por meio do cálculo da informação mútua, entre um conjunto de atributos de uma matriz

Figura 3.4: Mapa de calor expressando a correlação de Pearson entre quatro variáveis genéricas.



Fonte: Autoria Própria

X e um vetor alvo Y, é possível selecionar aqueles atributos ou *features* que apresentam maior relevância para serem utilizadas, posteriormente, em modelos de previsão.

Caso se trabalhe com valores contínuos, pode-se definir informação mútua como:

$$I(X, Y) = \int p(\mathbf{x}, \mathbf{y}) \log \left(\frac{p(\mathbf{x}, \mathbf{y})}{p(\mathbf{x})p(\mathbf{y})} \right) d\mathbf{x}d\mathbf{y}. \quad (3.2)$$

Já se os valores forem discretos, o cálculo é dado por:

$$I(X, Y) = \sum \sum p(x, y) \log \left(\frac{p(x, y)}{p(x)p(y)} \right), \quad (3.3)$$

em que $I(X, Y)$ é a informação mútua entre X e Y, $p(\mathbf{x}, \mathbf{y})$ é a distribuição de probabilidade conjunta de X e Y, e $p(\mathbf{x})$ e $p(\mathbf{y})$ são funções de probabilidade de distribuição marginal de X e Y, respectivamente.

3.2 Considerações finais

Neste capítulo foram descritas técnicas de pré-processamento de dados. Trata-se de um conjunto de atividades que envolvem preparação, organização e estruturação dos dados, consistindo em uma etapa fundamental anterior a realização de análises estatísticas e predi-

ções. Atividades de pré-processamento são determinantes para a qualidade final dos dados analisados. Dentro da etapa de pré-processamento, muitas vezes, tem-se que lidar com problemas relacionados com valores faltantes e valores discrepantes ou fora dos padrões (*outliers*). Análises visuais como o gráfico de caixa e de violino podem auxiliar no processo de identificação de *outliers*. Já no caso de valores ausentes, foram vistas técnicas como: substituição pela média; interpolação com os dados disponíveis; análise de atributos fortemente correlacionadas, entre outros. Por fim, foi estudado o processo de seleção de atributos, que é uma importante etapa do pré-processamento de dados, uma vez que contribui para selecionar as variáveis de maior impacto no modelo, além de eliminar informações redundantes.

Capítulo 4

Previsão de séries temporais multivariadas

Este capítulo analisa o modelo de previsão de séries temporais multivariada VAR (do inglês *Vector AutoRegression*), descrevendo o seu processo de previsão.

4.1 Propriedade das séries temporais

Uma série temporal é simplesmente uma série de pontos de dados ordenados no tempo, em períodos igualmente espaçados. Em uma série temporal, o tempo costuma ser a variável independente. Os modelos de previsão de séries temporais (PST) visam prever valores futuros desta série.

As séries temporais apresentam algumas propriedades importantes, das quais valem destacar:

- **Tendência:** determina se a série está crescendo, diminuindo ou se é estável ao longo do tempo. Essa propriedade ajuda no conhecimento do comportamento futuro da série à curto, médio ou longo prazo;
- **Sazonalidade:** corresponde a flutuações periódicas, isto é, descreve fenômenos que se repetem a cada período idêntico do tempo;
- **Erro Aleatório:** são flutuações inexplicáveis e inesperadas, as quais não se podem

prever.

A expressão matemática que representa a série temporal, considerando a análise clássica aditiva é expressa na Equação 4.1.

$$Y_t = T_t + S_t + a_t, \quad (4.1)$$

em que Y_t representa a série temporal, T_t a componente de tendência, S_t a componente sazonal e a_t o fator aleatório, o qual não é possível prever.

A análise de séries temporais univariadas e multivariadas vem sendo aplicadas em diversos setores para entender o comportamento futuro de um determinado fenômeno.

Ademais, modelos multivariados de previsão muitas vezes apresentam a capacidade de obter uma visão mais realista do problema quando comparados a modelos univariados, visto que analisam a influência de múltiplos fatores na previsão. Um modelo bastante usado na previsão de séries temporais multivariadas é o modelo VAR, que será discutido nas próximas seções.

4.2 Modelo VAR

O modelo VAR (do inglês *Vector AutoRegression*), proposto por Sims em 1980, é um dos modelos mais flexíveis e fáceis de implementar na análise de séries temporais multivariadas. Trata-se de uma extensão do modelo autorregressivo univariado (AR, do inglês *Autoregressive*) aplicado para séries temporais multivariadas dinâmicas.

Segundo Zivot e Wang[15], o modelo VAR tem mostrado bons resultados especialmente útil para descrever o comportamento dinâmico e realizar previsões de séries temporais no setor financeiro. Apresentando previsões superiores àquelas de modelos de séries temporais univariados. Nesse modelo de previsão, as variáveis são conjuntamente explicadas, isto é, cada variável é determinada por suas próprias defasagens no tempo e pelas defasagens das demais variáveis que compõem o conjunto dos dados de entrada do modelo. A grande vantagem no modelo VAR está no fato de que ele permite analisar o efeito da variação de determinada variável sobre as demais variáveis ao longo do tempo.

4.2.1 Formulação matemática do modelo VAR

Sendo $\mathbf{Y}_t = (y_{1t}, y_{2t}, \dots, y_{nt})'$, em que y_{nt} representa uma observação da série temporal y_n no intervalo de tempo t , um modelo VAR(p) pode ser expresso matematicamente, conforme a Equação 4.2

$$\mathbf{Y}_t = \mathbf{c} + \mathbf{M}_1 \mathbf{Y}_{t-1} + \mathbf{M}_2 \mathbf{Y}_{t-2} + \dots + \mathbf{M}_p \mathbf{Y}_{t-p} + \boldsymbol{\varepsilon}_t, t = 1, \dots, T, \quad (4.2)$$

em que $\mathbf{c} = (c_1, c_2, \dots, c_n)'$ é um vetor fixo de constantes adicionadas ao modelo, o que permite a possibilidade de média $E(\mathbf{Y}_t)$ não-nula; \mathbf{M}_p são matrizes de coeficientes, as quais representam a sensibilidade de uma variável do modelo com relação a uma defasagem de outra variável endógena do modelo no tempo; e $\boldsymbol{\varepsilon}_t = (\varepsilon_{1t}, \varepsilon_{2t}, \dots, \varepsilon_{Kt})'$ representa um vetor de ruído branco, isto é, média nula (Equação 4.3), variância constante (Equação 4.4), e não apresentam correlação serial (Equação 4.5).

$$E(\boldsymbol{\varepsilon}) = 0, \quad (4.3)$$

$$E(\varepsilon_t \varepsilon_t') = \sigma^2, \quad (4.4)$$

$$E(\varepsilon_t \varepsilon_s') = 0. \quad (4.5)$$

Assim, um modelo VAR(2), ou bivariado, poderia ser expresso por:

$$\begin{aligned} \begin{pmatrix} y_{1t} \\ y_{2t} \end{pmatrix} &= \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} + \begin{pmatrix} m_{11}^1 & m_{12}^1 \\ m_{21}^1 & m_{22}^1 \end{pmatrix} \begin{pmatrix} y_{1t-1} \\ y_{2t-1} \end{pmatrix} \\ &+ \begin{pmatrix} m_{11}^2 & m_{12}^2 \\ m_{21}^2 & m_{22}^2 \end{pmatrix} \begin{pmatrix} y_{1t-2} \\ y_{2t-2} \end{pmatrix} + \begin{pmatrix} \varepsilon_{1t} \\ \varepsilon_{2t} \end{pmatrix}, \end{aligned} \quad (4.6)$$

ou

$$\begin{aligned} y_{1t} &= c_1 + m_{11}^1 y_{1t-1} + m_{12}^1 y_{2t-1} + m_{11}^2 y_{1t-2} + m_{12}^2 y_{2t-2} + \varepsilon_{1t} \\ y_{2t} &= c_2 + m_{21}^1 y_{1t-1} + m_{22}^1 y_{2t-1} + m_{21}^2 y_{1t-2} + m_{22}^2 y_{2t-2} + \varepsilon_{2t}, \end{aligned} \quad (4.7)$$

4.2.2 Estimação do modelo VAR

De acordo com Santos[16], na estimação das equações do modelo VAR, o sistema apresenta uma estrutura fixa, com as mesmas variáveis em todas as equações e com o mesmo número de defasagens, sendo conhecido como “VAR puro”. O estimador para a representação padrão de um processo VAR(p) é definido por mínimos quadrados.

Considerando a primeira equação do VAR(p) representado pela Equação 4.2, tem-se:

$$y_{1T} = c_1 + m_{11}y_{1T-1} + m_{12}y_{2T-1} + \varepsilon_{1T}. \quad (4.8)$$

Sendo

$$\mathbf{y}_1 = [y_{11}, \dots, y_{1T}]', \quad (4.9)$$

$$\mathbf{X}_1 = \begin{bmatrix} 1 & y_{1,0} & y_{2,0} \\ 1 & y_{1,1} & y_{2,1} \\ \vdots & \vdots & \vdots \\ 1 & y_{1,T-1} & y_{2,T-1} \end{bmatrix}, \quad (4.10)$$

$$\mathbf{m}_1 = [c_1, m_{11}, m_{12}]', \quad (4.11)$$

$$\boldsymbol{\varepsilon}_1 = [\varepsilon_{11}, \dots, \varepsilon_{1T}]', \quad (4.12)$$

então:

$$\mathbf{y}_1 = \mathbf{X}\mathbf{m}_1 + \boldsymbol{\varepsilon}_1 \quad (4.13)$$

m_1 , pode então ser calculado pelo método dos mínimos quadrados:

$$m_1 = (\mathbf{X}'\mathbf{X})^{-1} \mathbf{X}'\mathbf{y}_1 \quad (4.14)$$

4.2.3 Seleção de defasagem do modelo VAR

A seleção de defasagem é um dos aspectos importantes da especificação do modelo VAR. Em aplicações práticas, geralmente é escolhido um número máximo de atrasos, p_{\max} , e após isso, é feita uma avaliação do desempenho do modelo, incluindo $p = 0, 1, \dots, p_{\max}$.

Segundo Zivot e Wang[15], Os critérios de seleção de modelo VAR(p) têm a forma:

$$IC(p) = \ln |\bar{\Sigma}(p)| + c_T \cdot \varphi(n, p), \quad (4.15)$$

onde $IC(p)$ é o critério de informação, $\tilde{\Sigma}(p) = T^{-1} \sum_{t=1}^T \hat{\varepsilon}_t \hat{\varepsilon}_t'$ é a matriz de covariância residual de um modelo VAR(p) sem correção de graus de liberdade, c_T é uma sequência indexada pelo tamanho da amostra e tamanho T , e $\varphi(n, p)$ é uma função de penalidade que penaliza modelos VAR (p) muito grandes.

O modelo VAR(p) ideal é aquele que minimiza os erros de previsão. Alguns métodos que mensuram a qualidade de um modelo estatístico e poderiam ser usados para a escolha ideal de p são listados abaixo:

- Akaike (AIC) (Equação 4.16);
- Schwarz-Bayesian (BIC) (Equação 4.17);
- Hannan-Quinn (HQ) (Equação 4.18).

Esses métodos geralmente são integrados a *softwares* de modo que a seleção de atrasos é quase que totalmente automatizada.

$$AIC(p) = \ln |\bar{\Sigma}(p)| + \frac{2}{T}pn^2, \quad (4.16)$$

$$BIC(p) = \ln |\bar{\Sigma}(p)| + \frac{\ln T}{T}pn^2, \quad (4.17)$$

$$HQ(p) = \ln |\bar{\Sigma}(p)| + \frac{2 \ln \ln T}{T}pn^2. \quad (4.18)$$

4.3 Considerações finais

Neste capítulo, foi apresentado o conceito de séries temporais, bem como a descrição matemática do modelo VAR, usado para realizar previsões de séries temporais multivariadas. Também foi descrito o processo de estimação dos parâmetros do processo, bem como alguns métodos de escolha do número ótimo de defasagem a ser utilizado pelo modelo.

Capítulo 5

Deep Learning em problemas de classificação

Este capítulo traz uma descrição do processo de análise de séries temporais multivariadas aplicadas em processos de classificação. Haverá um destaque para modelos de previsão que implementam *Deep Learning* (DL), em especial modelos LSTM.

5.1 Classificação de Séries Temporais

Durante as últimas duas décadas, a Classificação de Séries Temporais (CST) foi considerada uma das problemas mais desafiadores para o DM (do inglês, *Data Mining*) [17]. Com o aumento da disponibilidade de dados temporais, aumentou a demanda pela exploração de diversos algoritmos de CST.

Devido à sua ordenação temporal natural, os dados da série temporal estão presentes em quase todas as tarefas que requerem algum tipo de processo cognitivo humano. Na verdade, qualquer problema de classificação que utilize dados que são registrados levando em consideração algum tipo de ordenação, pode ser considerado um problema de CST [18].

As séries temporais são encontradas em muitas aplicações do mundo real, desde registros eletrônicos de saúde, até leituras de sensores de monitoramento em processos industriais. Distinguir a série temporal de operações normais daquela de um dispositivo com alguma anomalia, e reconhecer a anomalia, é um exemplo de problema de classificação de série

temporal.

Havendo um conjunto de objetos com a mesma estrutura (valores reais, vetores de mesmo comprimento ou matrizes com o mesmo tamanho) e um conjunto fixo de classes diferentes, temos o que chamamos de *dataset*. Defini-se *dataset* como um conjunto de pares (objeto, classe), ou seja, cada objeto está associada uma determinada classe.

Um problema de Classificação consiste em desenvolver um modelo que associe a um novo objeto, com a mesma estrutura dos demais, a probabilidade de pertencer às classes previamente apresentadas no processo de aprendizagem do modelo. Para isso, são identificados padrões inerentes aos dados de entrada.

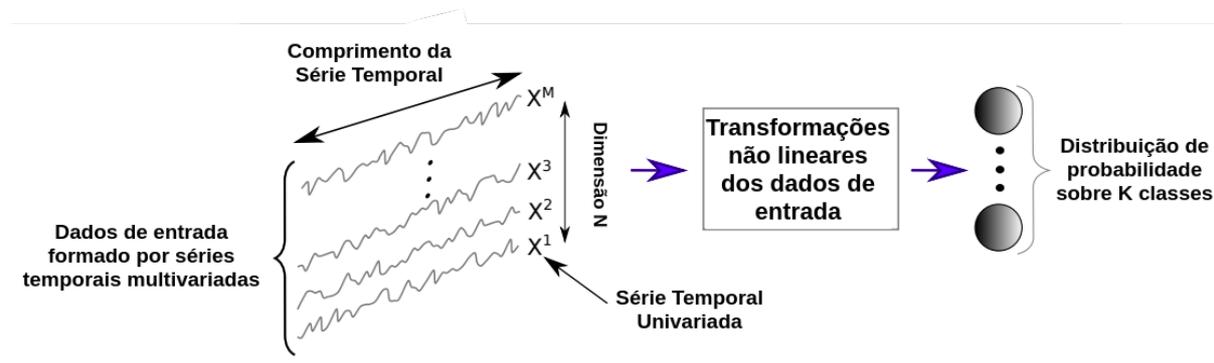
Uma série temporal univariada representa um conjunto ordenado de valores, enquanto uma série temporal multivariada de dimensão N consiste em N séries temporais univariadas distintas com o mesmo número de observações. Um problema de classificação de série temporal, portanto, consiste em um problema de classificação em que os objetos do conjunto de dados de entrada são séries temporais univariadas ou multivariadas [19].

Em resumo, podemos destacar algumas definições no processo de CST, como:

- Definição 1: Uma série temporal univariada $X = [x_1, x_2, \dots, x_T]$ é um conjunto ordenado de valores reais. O tamanho do vetor X equivale ao número de observações da variável no tempo T .
- Definição 2: Uma série temporal multivariada de dimensão N , $X = [X^1, X^2, \dots, X^N]$ consiste em N séries temporais univariadas distintas $X^i \in \mathbb{R}^T$.
- Definição 3: Um *dataset* $D = [(X_1, Y_1), (X_2, Y_2), \dots, (X_N, Y_N)]$ consiste em uma coleção de pares (X_i, Y_i) onde X_i pode ser tanto uma série temporal univariada quanto uma série temporal multivariada, sendo Y_i o rótulo ou classe correspondente.
- Definição 4: o objetivo da CST é treinar um classificador para mapear o *dataset* de entrada para uma distribuição de probabilidade sobre os valores das variáveis de classe (rótulos).

A Figura 5.1 ilustra o processo de classificação de séries temporais.

De acordo com Pra[19], os modelos de aprendizado profundo, ou DL, incorporam ferramentas de *Feature Engineering* internamente, otimizando-os e eliminando a necessidade de

Figura 5.1: Fluxograma do processo de classificação de séries temporais

Fonte: Adaptado de Fawaz et al.[18]

realiza-las manualmente.

Feature Engineering é o processo de extrair e selecionar, a partir de dados brutos, características que possam ser utilizadas de forma efetiva em modelos preditivos. Portanto, eles são capazes de extrair informações das séries temporais de forma mais rápida, direta e completa, tornando-os grandes aliados nos modelos de CST. Nas próximas seções serão descritas algumas arquiteturas de redes neurais profundas, com destaque para o modelo LSTM.

5.1.1 Redes neurais recorrentes

As redes neurais artificiais (ANNs, do inglês *Artificial Neural Network*) são artifícios que simulam o funcionamento do cérebro humano, por meio de uma rede de neurônios artificiais interconectados, visando estabelecer um processamento em paralelo similar ao que ocorre no cérebro humano [20]. Por meio dessa rede complexa de neurônios, as ANNs tentam identificar padrões nos dados de entrada, para assim conseguir realizar previsões.

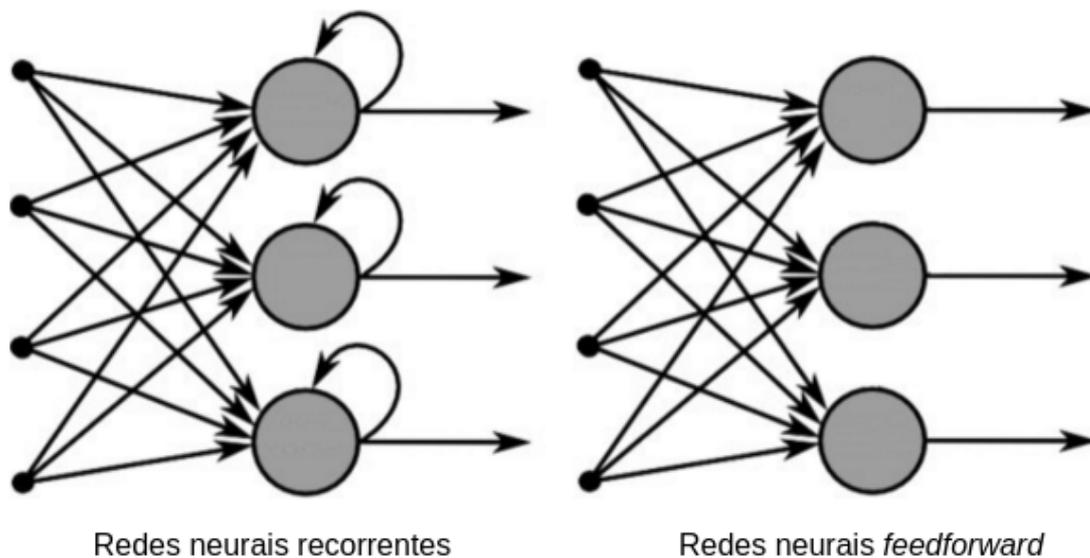
As redes neurais recorrentes (RNN, do inglês *Recurrent Neural Networks*) são tipos de redes neurais artificiais inspiradas na conectividade cíclica de neurônios que ocorre no cérebro. Esse tipo de rede utiliza *loops* de funções iterativas para armazenar informações. A Figura 5.2 ilustra a diferença entre a arquitetura de uma RNN, e uma ANN com arquitetura tradicional *feedforward*.

As RNNs foram projetadas para reconhecer padrões em sequências de dados, tais como: texto, genomas, caligrafia, palavra falada ou dados numéricos advindos de leituras de sen-

sores, bolsa de valores, entre outros.

As RNNs surgiram na busca de superar as dificuldades enfrentadas com relação ao aprendizado de dependências. Isto é, nas soluções de problemas que envolvem vários estágios de processamento [21]. Essa questão está associada ao fato de que muita informação relevante do começo de uma sequência pode ser esquecida nas etapas intermediárias. Redes Neurais Recorrentes são redes com *loops*, permitindo que as informações persistam, sendo assim, as RNNs possuem o que um certo tipo de memória, que faz analogia a memória do cérebro humano.

Figura 5.2: Comparação das arquiteturas entre uma rede neural do tipo *Feedforward* e do tipo Recorrente.



Fonte: Adaptado de Pai[22]

As RNNs apresentam vantagens frente a uma ANN do tipo *feedforward* pelo *loop* de *feedback*, que conecta suas próprias saídas momento após momento como entrada. Com isso, esse tipo de rede apresenta o que é conhecido por memória. A importância disso é que há informações na própria sequência que podem ser usadas pelas RNNs para executar tarefas, o que não é permitido nas redes do tipo *feedforward*[22].

5.1.2 LSTM

Conforme explanado anteriormente, a principal característica das RNNs é uma conexão cíclica, que permite ao modelo possuir a capacidade de atualizar o estado atual com base nos estados anteriores e nos dados de entrada atuais [23]. Todavia, quando a lacuna entre os dados de entrada é relativamente grande, as RNNs muitas vezes são incapazes de conectar informações relevantes. Para lidar com as “dependências de longo prazo”, Hochreiter e Schmidhuber[24] propuseram a rede de memória de longa de curto prazo ou LSTM (do inglês *Long Short-Term Memory*).

A LSTM é uma arquitetura de RNN que apresenta a vantagem de lembrar de valores em intervalos arbitrários. Essa arquitetura se mostrou bem adequada para classificar e prever séries temporais que apresentam intervalos de tempo de duração desconhecida.

Normalmente RNNs são constituídas por células ocultas ou células recorrentes padrão, como células sigma (σ) e células *tanh*, que estão ilustradas em vermelho nas Figuras 5.3 e 5.4. As expressões matemáticas da célula sigma recorrente padrão são escritas da seguinte forma:

$$h_t = \sigma(W_h h_{t-1} + W_x x_t + b), \quad (5.1)$$

$$y_t = h_t, \quad (5.2)$$

em que x_t , h_t e y_t denotam, respectivamente, a entrada, a informação recorrente e a saída da célula no tempo t , W_h e W_x são os pesos; e b é o fator *bias*.

Neste tipo de arquitetura, conforme a lacuna entre as entradas relacionadas aumenta, é difícil aprender as informações de conexão. Hochreiter e Schmidhuber[24] analisaram as razões fundamentais para o problema das dependências de longo prazo e identificaram que sinais de erro que fluem para trás no tempo (*backpropagation*) tendem a desaparecer.

Os autores melhoraram a capacidade de memorização da célula recorrente padrão, introduzindo um “*gate*” na célula. A informação é retida pelas células e as manipulações de memória são feitas pelos portões (*gates*). Existem três portões:

- *Forget Gate*: As informações que não são mais úteis no estado da célula são removidas com o *Forget Gate*. Duas entradas: x_t (entrada no tempo t) e h_{t-1} (saída de célula

anterior) são alimentadas ao *gate*, multiplicadas por matrizes de peso e somadas do bias. Em seguida, são passadas por uma função de ativação de saída binária, na qual caso a saída seja 1, a informação será retida na célula para ser utilizada em outro momento.

- *Input Gate*: A adição de informações úteis ao estado da célula é feita pelo *Input Gate*. Primeiramente, uma função tipo sigmoide filtra os valores a serem lembrados de maneira similar ao que acontece no *Forget Gate*, usando as entradas h_{t-1} e x_t . Esses valores são multiplicados por um vetor, que é criado com a função *tanh*, regulando todos os valores possíveis de h_{t-1} e x_t num intervalo de de -1 a +1.
- *Output Gate*: É responsável por extrair informações úteis do estado da célula atual e transformá-las em um vetor de saída. Os valores do vetor criado pela função *tanh* e os valores regulados pela função sigmoide são multiplicados para serem enviados como uma saída e entrada para a próxima célula.

Existem algumas variações de arquiteturas LSTM, existindo aquelas que incluem o *Forget Gate* e aquelas que apresentam apenas os *gates* de entrada e saída. De modo geral, o termo célula LSTM se refere a arquitetura que apresenta os três *gates* listados anteriormente, incluindo o *Forget Gate*[23].

Primeiro, será apresentado o modelo LSTM original que possui apenas o *Input Gate* e o *Output Gate*. As expressões matemáticas que expressam o funcionamento desse tipo de LSTM são dadas conforme as equações a seguir:

$$i_t = \sigma(W_{ih}h_{t-1} + W_{ix}x_t + b_i), \quad (5.3)$$

$$\tilde{c}_t = \tanh(W_{\tilde{c}}h_{t-1} + W_{\tilde{c}x}x_t + b_{\tilde{c}}), \quad (5.4)$$

$$c_t = c_{t-1} + i_t \cdot \tilde{c}_t, \quad (5.5)$$

$$o_t = \sigma(W_{oh}h_{t-1} + W_{ox}x_t + b_o), \quad (5.6)$$

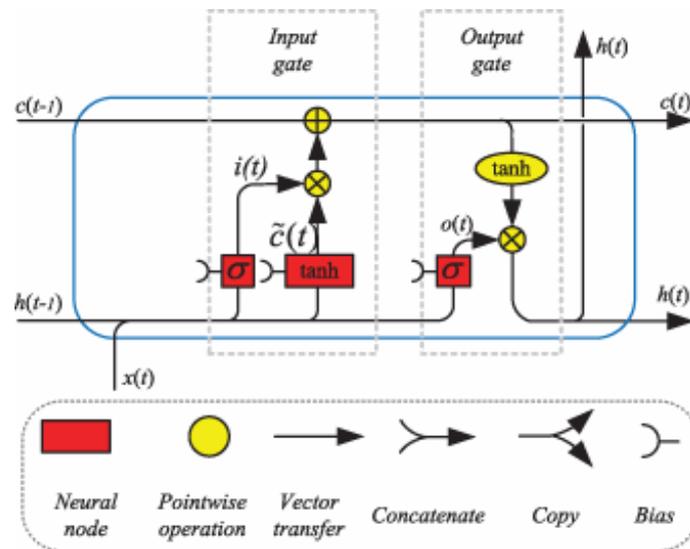
$$h_t = o_t \cdot \tanh(c_t), \quad (5.7)$$

nas quais, c_t representa o estado da célula de LSTM. i_t e o_t representam a saída da célula

sigma no *Input Gate* e *Output Gate* respectivamente. W_i , W_c e W_o são os pesos, e o operador $'\cdot'$ denota a multiplicação ponto a ponto de dois vetores. Ao atualizar o estado da célula, a porta de entrada pode decidir quais novas informações podem ser armazenadas no estado da célula e a porta de saída determina quais informações podem ser enviadas baseado no estado da célula.

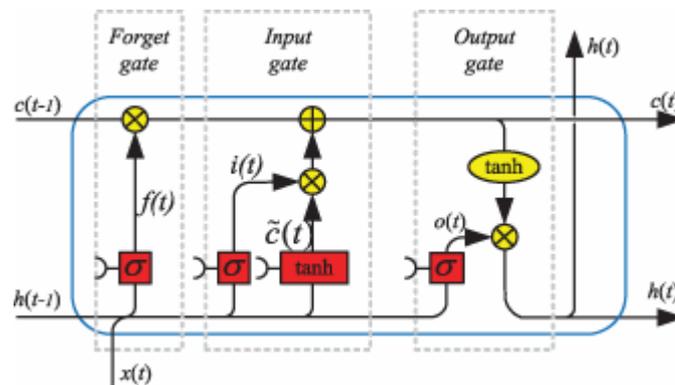
Na Figura 5.4, é percebida uma arquitetura de LSTM que apresenta o *Input Gate*, *Forget Gate* o *Output Gate*, enquanto a Figura 5.3 ilustra a arquitetura original apenas com o *Input Gate* e o *Output Gate*.

Figura 5.3: Arquitetura LSTM original



Fonte: Yu et al.[23]

Figura 5.4: Arquitetura LSTM com *Input Gate*, *Forget Gate* o *Output Gate*.



Fonte: Yu et al.[23]

Existem diversas plataformas que facilitam a aplicação de redes LSTM em modelos de

previsão voltados para a classificação. Dentre eles, é possível destacar o `keras` e o `pytorch`.

5.1.3 Uso de CPU vs GPU em modelos de *Deep Learning*

Uma unidade de processamento central, ou CPU (do inglês *Central Processing Unit*), consiste em uma unidade eletrônica responsável por executar instruções de máquina. As CPUs são a parte principal do *hardware* de um computador ou controlador lógico programável [25].

Embora muitos processadores modernos possuam vários núcleos, cada um dos quais pode executar vários *threads*, as CPUs, na prática vem se mostrando com desempenho inferior no aprendizado de máquina quando comparadas com GPUs (do inglês *Graphics Processing Unit*). Isso acontece principalmente devido ao número limitado de operações que podem ser realizadas simultaneamente na CPU [25].

Uma unidade de processamento gráfico ou GPU, consiste em um dispositivo separado de um computador pessoal que executa a renderização de gráficos. De acordo com Cao, Balasubramanian e Balasubramanian[26], os recursos de processamento paralelo das GPUs apresentam a possibilidade de acelerar o treinamento de modelos de DL, como aqueles baseados em LSTM, quando comparada com as implementações de CPU.

As GPUs apresentam alta taxa de eficiência na multiplicação de matrizes, a qual basicamente forma o núcleo do aprendizado de máquina. O que acontece é que as GPUs podem realizar cálculos múltiplos e simultâneos. Isso permite a distribuição de processos de treinamento e pode acelerar significativamente as operações de DL.

A principal contribuição das GPUs, portanto, é o paralelismo ou processamento simultâneo de partes de um todo. De modo geral, há quatro arquiteturas usadas para implementações de processamento paralelo, como:

- Instrução única, dados únicos (SISD, do inglês *Single Instruction, Single Data*);
- Instrução única, dados múltiplos (SIMD, do inglês *Single Instruction, Multiple Data*);
- Várias instruções, dados únicos (MISD, do inglês *Multiple Instruction, Single Data*);
- Várias instruções, vários dados (MIMD, do inglês *Multiple Instruction, Multiple Data*).

Geralmente, grande parte das CPUs são processadores *multi-core*, operando com uma arquitetura do tipo MIMD. Em contrapartida, algumas GPUs usam uma arquitetura SIMD. Dessa maneira, enquanto uma CPU é dividida em vários núcleos para que possam realizar várias tarefas ao mesmo tempo, a GPU terá centenas ou milhares de núcleos, todos dedicados a uma única tarefa. Esses cálculos podem ser simples, executados com maior frequência e independentes uns dos outros.

Outra contribuição do uso de GPUs está relacionada ao fato de que, em geral, apresentam maior escalabilidade do que as CPUs, permitindo processar conjuntos de dados maiores com mais rapidez. Uma GPU pode apresentar milhares de unidades lógicas aritméticas, cada uma delas realizando uma computação paralela. Esse fato acaba por conferir um rendimento mais alto e uma maior velocidade na execução de modelos que lidam com um grande volume de dados [25].

5.2 Considerações finais

Nesta seção foi descrito como ocorre o processo de classificação de séries temporais utilizando algoritmos de *Deep Learning*. Também foi esclarecida as vantagens do modelo LSTM para tratar com dados sequenciais (como no caso de séries temporais) e como ocorre o seu processamento. Por fim, foram apresentadas algumas vantagens da GPU em relação a CPU na implementação de modelos de DL, visto que as operações podem ser aceleradas com o paralelismo da distribuição de processos de treinamento.

Capítulo 6

Metodologia de pesquisa

Este capítulo descreve a metodologia de pesquisa desenvolvida para realizar a análise de séries temporais multivariadas para detecção de falhas em turbogeradores de uma plataforma tipo FPSO.

6.1 Sistema Elétrico da FPSO

Conforme visto no Capítulo 2 deste trabalho, os Sistemas de Geração de energia das plataformas tipo FPSOs configuram-se como unidades termelétricas flutuantes em alto mar, apresentando geração de energia própria e independente para que seja assegurada a alimentação elétrica em todos os processos de produção da unidade [9].

Nesse sentido, o funcionamento da unidade de produção da plataforma, é totalmente dependente do módulo de geração de energia, reiterando a relevância do monitoramento desse tipo de sistema e detecção de possíveis falhas.

Nesta pesquisa, analisou-se o sistema elétrico de uma plataforma tipo FPSO, que apresenta como fonte principal de energia um sistema de geração composto quatro turbogeradores do tipo GE-TG-5147001A/B/C/D, cujo funcionamento ocorre por turbinas a gás, com potência nominal de 25MW/31,25kVA, tensão nominal de 13,8 kV e frequência nominal de 60 Hz.

Como objeto base de estudo para esse trabalho, tem-se o turbogerador com maior número de ocorrências de falhas registradas num período desde o dia dia 01/05/2020 às 00:00:00 e

a até o dia 01/05/2021 às 00:00:00. A aquisição foi feita para um período de amostragem de um minuto. Para isso, serão analisados dados operacionais e registros de sensores de monitoramento desse turbogerador. A tabela Tabela 6.1 denota as causas de falha desse turbogerador no período de aquisição referido.

Tabela 6.1: Causas das falhas registradas no turbogerador em análise.

Data da Parada	Causa da parada
2020-05-25 08:33:00	Falha no sistema de controle da FCV-501A
2020-07-05 00:00:00	Parada por Trip por vibração alta crítica na turbina de potência - VT-502A
2020-07-11 09:50:00	Parada por Trip por vibração alta crítica na turbina de potência - VT-502A
2020-07-11 20:00:00	Parada por Trip por vibração alta crítica na turbina de potência.
2020-08-02 22:39:43	Parada por Trip - perda de controle da metering de gás
2020-08-25 04:02:00	Parada por Trip por vibração alta crítica na turbina de potência - VT-501
2020-08-25 22:35:00	Parada por Trip por vibração alta crítica na turbina de potência - VT-501
2020-12-14 01:44:00	Parada por Trip - perda de controle da metering de gás
2021-01-08 20:48:00	Parada por Trip - Temperatura do Óleo Mineral
2021-03-21 04:12:00	Trip por falha na Metering de gás FCV-501A e na válvula de Vent XV-004
2021-03-26 00:00:00	Trip por queda geral do CCM PN-TG-5147001-01-01
2021-04-20 00:00:00	Trip por falha no Cartão YAIC-1D1A

Em seus estudos, Castro et al.[1] selecionaram as variáveis de monitoramento relacionadas a esses tipos de falhas. Assim, de um total de 142 sensores de monitoramento para esse turbogerador foram selecionados 55 sensores para o estudo de detecção de falhas.

Com o grande número de registros de sensores que monitoram esse tipo de equipamento, a análise de falhas nessas máquinas é um processo que pode se beneficiar significativamente de técnicas de DM e ML.

Por meio dessas técnicas, pode-se identificar e extrair informações ou padrões em grandes conjuntos de dados, para que assim seja possível antecipar e definir resultados, como a identificação de falhas dispendiosas.

Uma vez que existam padrões discriminativos anteriores aos eventos de falha, pode ser realizada a detecção precoce da mesma no futuro. Nesse trabalho, desenvolveu-se um modelo de classificação de ocorrência ou não de falhas em turbogeradores de um sistema de geração de uma plataforma tipo FPSO.

Para isso, após coleta dos dados brutos dos 55 sensores de monitoramento, os dados passaram por uma etapa de pré-processamento anterior ao desenvolvimento do modelo, de modo a definir o conjunto de dados que formaria as séries temporais multivariadas utilizadas como entrada para o modelo de classificação e ocorrência ou não de falhas. As etapas de pré-processamento e desenvolvimento do modelo são descritas nas próximas seções.

6.2 Pré-processamento

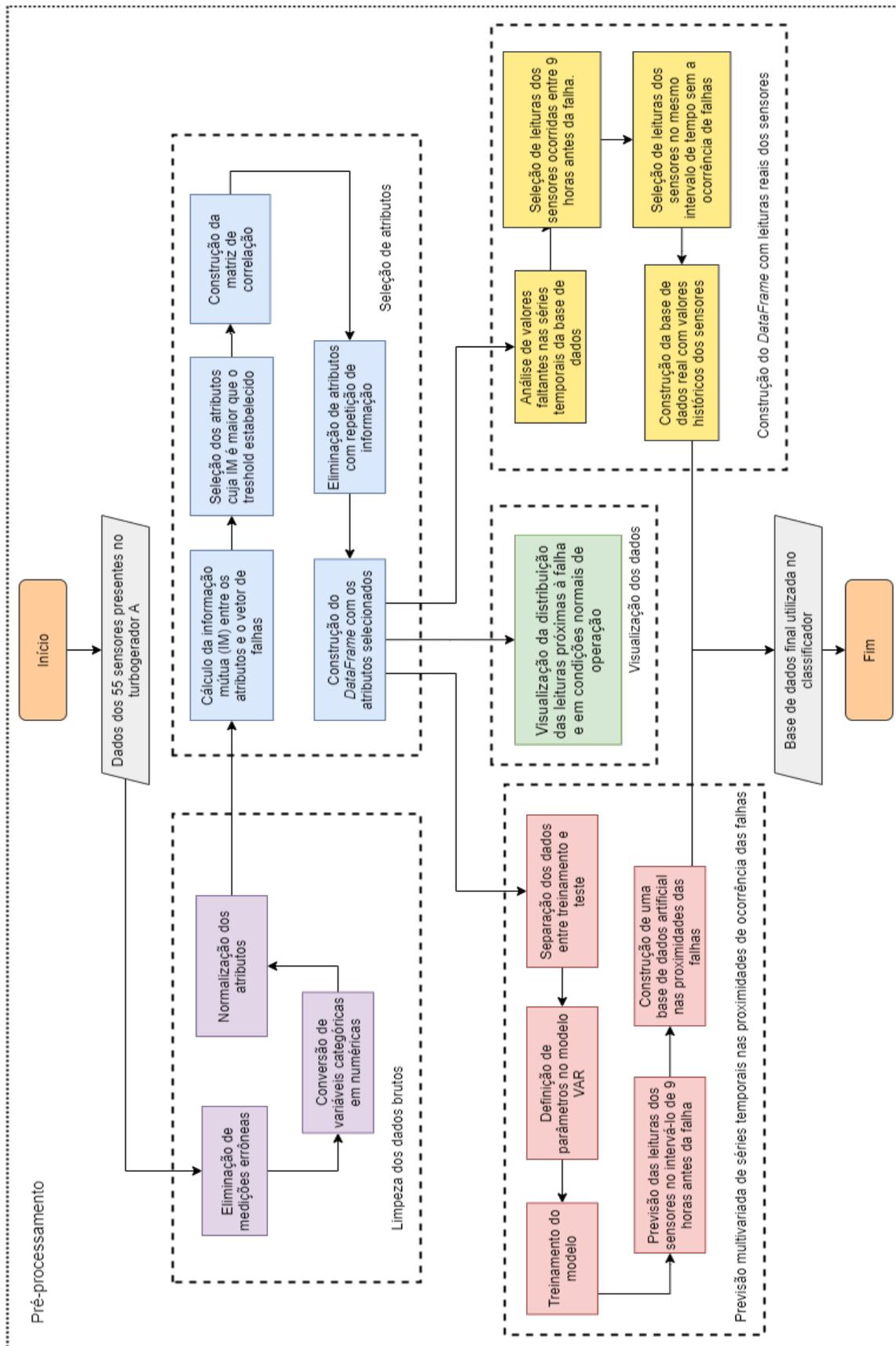
A metodologia do trabalho desenvolvido foi dividida em duas etapas: a etapa de pré-processamento e a etapa de desenvolvimento do modelo de classificação. Nesta seção será descrita a etapa de pré-processamento dos dados de entrada.

Nesta etapa, foram realizadas algumas atividades como eliminação de dados inconsistentes advindos de erros de medição na leitura de alguns sensores de monitoramento, seleção de atributos de maior importância no processo de identificação de falhas no turbogerador e identificação de informações redundantes entre os atributos do conjunto de dados por meio da análise da correlação linear.

Na Figura 6.1 é apresentado o fluxograma do pré-processamento dos dados brutos.

Conforme descrito no fluxograma, inicialmente foi realizada a etapa de limpeza dos dados

Figura 6.1: Etapa de pré-processamento do estudo de análise multivariada de séries temporais para detecção de falhas em turbogeradores de uma plataforma FPSO.



Fonte: Autoria própria.

brutos. Nesta etapa, medições errôneas advindas dos sensores de monitoramento foram eliminadas. Ademais, houve a conversão de variáveis categóricas em *labels* numéricos. No caso da leitura dos sensores de chama, por exemplo, valores categóricos "On" e "Off" foram convertidos para "1" e "0" respectivamente.

Para realizar este processo, foi utilizada a linguagem de programação `python`. O `pandas` é uma biblioteca para uso em `python`, *open-source*, que fornece ferramentas muito úteis para análise e manipulação de dados. Toda a organização do `pandas` é voltada para um objeto *DataFrame*, que consiste em uma estrutura de dados bidimensional com os dados alinhados de forma tabular, com linhas e colunas. A biblioteca `pandas` permite que sejam realizadas diversas operações entre *DataFrames*, como: exclusão de valores faltantes, concatenação de *DataFrames* distintos, análises estatísticas, entre outras.

Após a etapa de limpeza, conjunto de dados foram normalizados para valores entre "0" e "1". A Normalização é uma técnica aplicada com frequência na etapa de preparação dos dados com o objetivo de colocá-los em um intervalo de valores comuns. Não se trata de uma etapa obrigatória, porém pode impactar na acurácia do modelo de alguns algoritmos de aprendizagem profunda. A normalização é dada pela fórmula abaixo:

$$X_{\text{novo}} = \frac{X - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}}, \quad (6.1)$$

em que X é um elemento de um vetor, X_{min} e X_{max} são, respectivamente, o valor mínimo e máximo deste mesmo vetor, e X_{novo} corresponde ao valor normalizado da variável X .

O conjunto de dados analisados era composto ainda pelas 55 variáveis relacionadas aos sensores de monitoramento, associados a uma coluna chamada de vetor de falhas, com valores discretos (0, para condições normais de operação e 1 para indicação de falha). Após a etapa de limpeza dos dados brutos, iniciou-se a etapa de seleção de atributos. Essa etapa possui o intuito de selecionar os sensores de maior relevância para a identificação de falhas e eliminar aqueles que, estatisticamente, trazem a mesma informação.

Para este propósito, realizou-se o cálculo da informação mútua entre os atributos do conjunto de dados e o vetor de falhas. Para isso, recorreu-se a função `mutual_info_classif()` da biblioteca `feature_selection` do pacote `sklearn` do `python`.

Foi definido um *threshold* como critério de eliminação para os atributos que possuíssem

valor menor que 0,6 em uma escala de 0 a 1. Em seguida, construiu-se a matriz de correlação entre os atributos selecionados, eliminando aqueles que apresentavam correlação linear iguais a 1.

Com isso, montou-se um *DataFrame* com os atributos selecionados. Esses dados passaram por análises visuais, através de um mapa de calor com seus valores de correlações e plotagem do gráfico de violino.

Por fim, foi feita uma análise dos valores faltantes das séries temporais multivariadas que compõem o banco de dados. Após essa análise, selecionou-se as leituras dos sensores desde nove horas antes da falha até o momento da falha em medições de 1 em 1 minuto. Também selecionou-se medições no mesmo intervalo de tempo para condições normais de operação do turbogerador. Com isso construiu-se um *DataFrame* com leituras reais dos sensores de monitoramento em condições imediatamente anteriores a falha e em condições normais.

Na análise dos dados, percebe-se que há, majoritariamente, informações sobre o período de condição normal de operação do equipamento do que em momentos de falha. Com isso, para diminuir esse desbalanceamento entre as classes, recorreu-se a um processo de previsão multivariada de séries temporais nas proximidades de ocorrência de falhas com o auxílio do modelo VAR. Desse modo, construiu-se uma base de dados artificial para compor os dados de entrada anteriormente selecionados. Por fim, constituiu-se a base de dados final a ser usada no classificador.

6.3 Modelo de detecção de falhas

Antes do desenvolvimento do classificador, ou detector de falhas do turbogerador, é interessante que os dados previamente pré-processados sejam transformados para o formato de sequência. Ou seja, os dados foram transformados em *DataFrames* menores com dados sequenciais (dados ordenados em intervalos de 9 horas, com granularidade de 1 minuto).

Nesse sentido, foram construídos *DataFrames* com registros de 9 horas imediatamente antes da falha no turbogerador (com passo de 1 minuto). Mas também foram obtidas sequências com intervalos de tempo de mesmo comprimento, em condições normais de operação.

Posteriormente, houve a separação dos dados entre treinamento, validação e teste. Sendo

55% dos dados utilizados para treinamento, 10% para validação e 35% para teste.

A partir daí, iniciou-se o desenvolvimento de um modelo LSTM que seria utilizado para a classificação de STs, auxiliando no processo de identificação de falhas no equipamento por meio da análise multivariada de STs. Configurou-se os parâmetros do modelo LSTM, o número de épocas de treinamento e as condições de parada do mesmo.

Na etapa de treinamento do modelo, foi utilizada uma GPU com o objetivo de acelerar o treinamento da LSTM. Após o treinamento, o modelo foi armazenado, bem como os cálculos do erro médio quadrático (MSE) nas etapas de treinamento e validação.

Após o treinamento, o modelo realizou a previsão para os dados de teste, indicando ou não a ocorrência de falhas para os dados de entrada.

Por fim, foi feita uma avaliação quantitativa do modelo de classificação de STs através de métricas como: acurácia, *precision*, *recall* e *f1-score*, além da visualização da matriz de confusão. Essas métricas são definidas a seguir:

$$\text{acurácia} = \frac{TP + TN}{TP + TN + FP + FN}, \quad (6.2)$$

$$\text{precision} = \frac{TP}{TP + FP}, \quad (6.3)$$

$$\text{recall} = \frac{TP}{TP + FN}, \quad (6.4)$$

$$\text{f1-score} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}, \quad (6.5)$$

Tabela 6.2: Matriz de confusão genérica.

Valor previsto \ Valor real	Positivo	Negativo
Positivo	Verdadeiro Positivo (TP)	Falso Positivo (FP)
Negativo	Falso Negativo (FN)	Verdadeiro negativo (TN)

em que:

- Positivo Verdadeiro (*True Positive* – TP): a classe prevista e observada fazem parte da classe positiva;
- Falso Positivo (*False Positive* – FP): a classe prevista retornou positivo, mas a original observada era negativa;

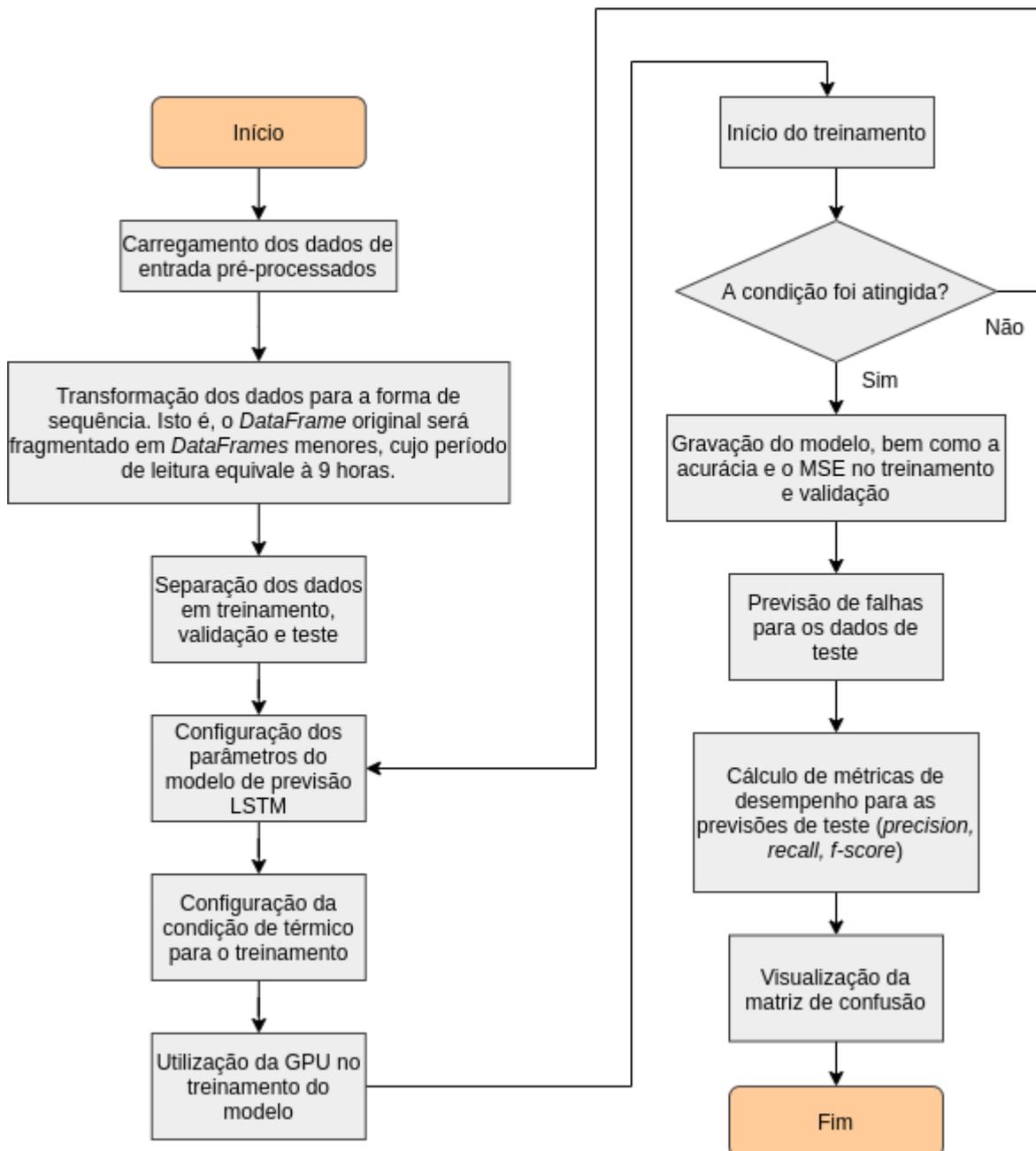
- Negativo Verdadeiro (*True Negative* – TN) os valores previstos e observados são da classe negativa;
- Falso Negativo (*False Negative* – FN): o valor previsto correspondeu a classe negativa, mas o original observado era da classe positivo.

A Figura 6.2 mostra o passo a passo da metodologia de desenvolvimento do modelo de detecção de falhas do turbogerador da FPSO.

6.4 Considerações finais

Neste capítulo foi vista a metodologia de pesquisa do trabalho de conclusão de curso. Foram mostradas as etapas realizadas no pré-processamento dos dados brutos dos sensores de monitoramento do turbogerador que foi objeto de estudo do trabalho. Dentre as etapas, é possível destacar: a limpeza de dados; a seleção de atributos; a construção de uma base de dados artificial para dados anteriores a falha utilizando o modelo VAR; e a junção dos *DataFrames* contendo os valores das bases de dados real e artificial das leituras dos sensores. No processo de desenvolvimento do modelo, foi implementado uma LSTM para classificação de séries temporais. Nesta etapa utilizou-se uma GPU no intuito de acelerar o treinamento do modelo. Por fim, realizou-se a previsão de falhas com os dados de teste e o modelo foi avaliado quantitativamente.

Figura 6.2: Fluxograma do modelo de detecção de falhas em turbogeradores de uma plataforma FPSO utilizando LSTM.



Fonte: Autoria própria.

Capítulo 7

Resultados e análises

Este capítulo descreve os principais resultados obtidos desde o pré-processamento dos dados brutos advindos dos sensores de monitoramento do turbogerador até os resultados dos testes no modelo de detecção de falhas baseado na avaliação de séries temporais multivariadas anteriores ao momento da falha.

7.1 Pré-processamento

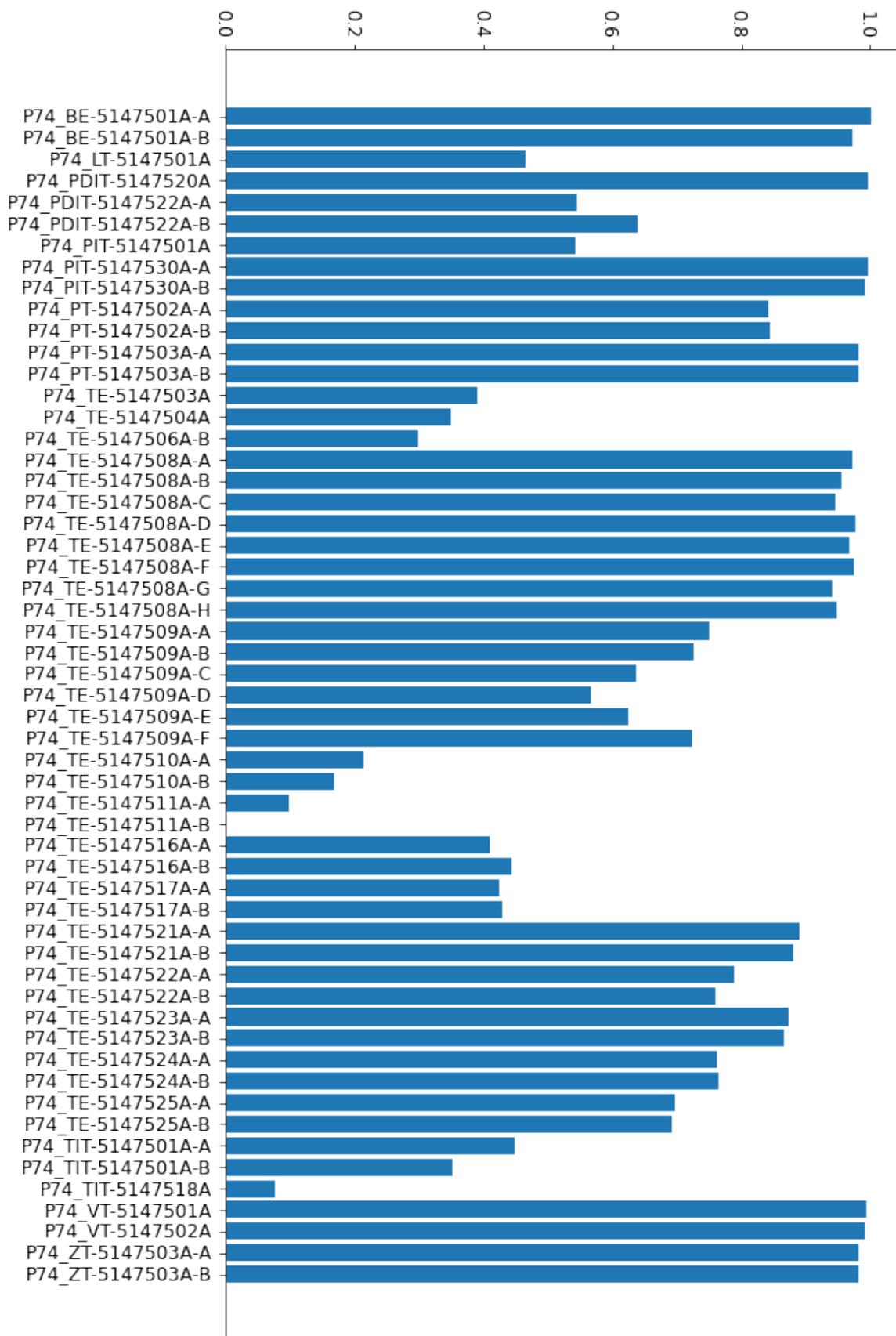
Conforme descrito no Capítulo 6, inicialmente houve um processo de limpeza de leituras inconsistentes nos dados dos sensores, nas quais as medições não foram possíveis. Após a etapa de limpeza, os dados foram normalizados e, em seguida, iniciou-se a etapa de seleção de atributos.

Primeiramente, realizou-se o cálculo da informação mútua entre os dados dos 55 sensores, e o vetor de ocorrência de falhas no turbogerador. Os resultados encontram-se resumidos no gráfico ilustrado na Figura 7.1

Estabeleceu-se um valor limite, ou *threshold* para as variáveis de maior relevância. Esse valor foi de 60%. Assim, aquelas variáveis que apresentassem informação mútua superior a 60% (ou 0.6 numa escala de 0 a 1) foram selecionadas, resultando em uma seleção de 32 sensores, conforme ilustrado na Figura 7.2.

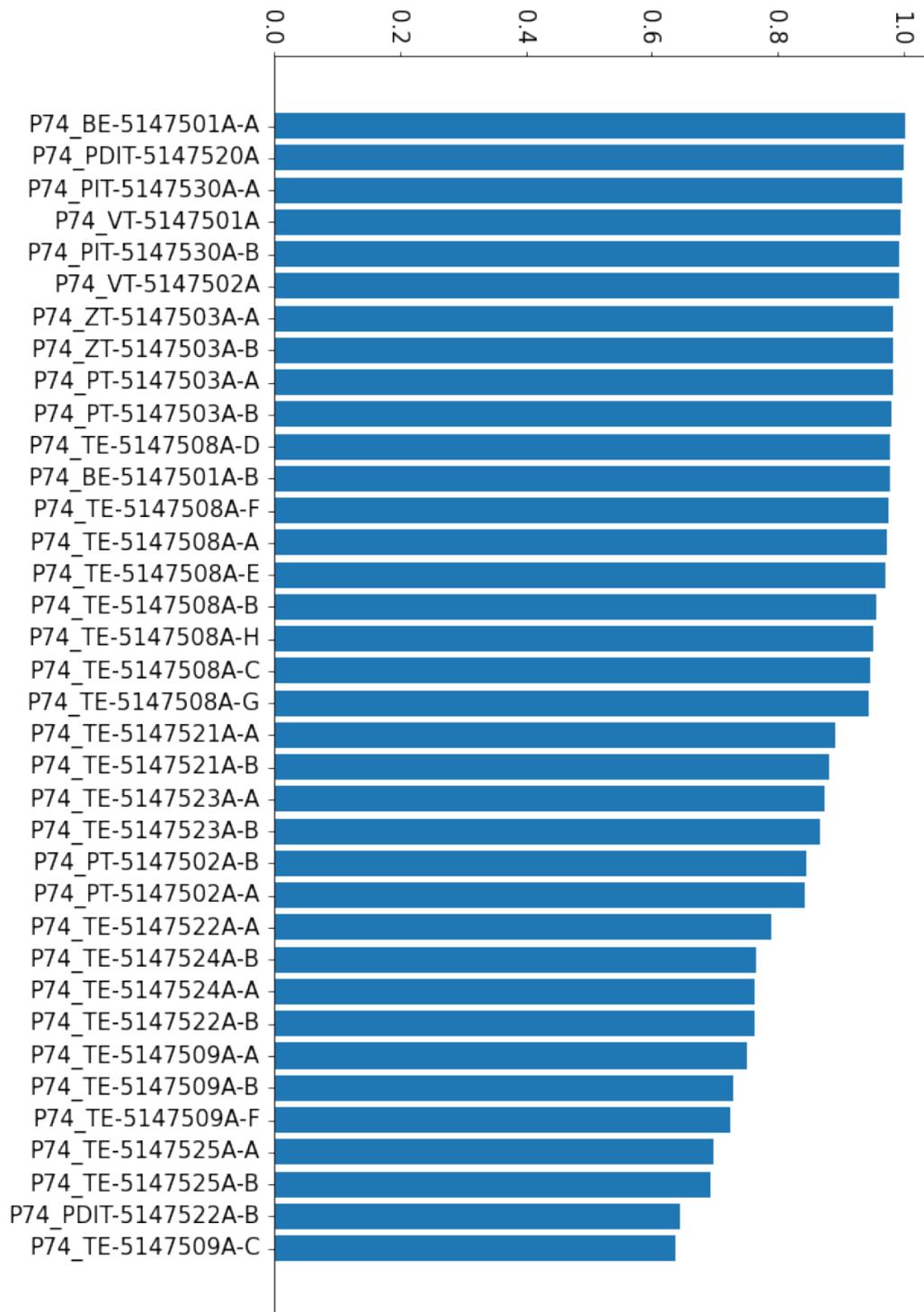
Em seguida, após o cálculo da informação mútua, iniciou-se o processo de análise da correlação linear entre os sensores selecionados, conforme mostrado na Seção 3.1.3 deste

Figura 7.1: Informação mútua entre os 55 sensores e o vetor de falhas.



Fonte: Autoria própria.

Figura 7.2: Seleção dos sensores com informação mútua acima do valor *threshold*



Fonte: Autoria própria.

documento. De modo a facilitar a análise, foi plotado o mapa de calor da correlação entre esses atributos, também chamada de matriz de correlação linear (Figura 7.3).

Analisando a Figura 7.3, percebe-se que muitos atributos apresentam correlação linear igual a 1, isto é, correlação linear máxima. Isso implica que esses atributos estão fortemente correlacionados e podem estar mensurando a mesma variável de monitoramento, resultando na captação de informações redundantes.

Sendo assim, para este estudo, selecionou-se os valores do triângulo superior da matriz de correlação e eliminou-se uma variável de cada par que apresentavam correlações lineares iguais a 1. Ao fim deste processo, foram selecionadas 16 "*features*" ou atributos. O resultado desta análise encontra-se na Figura 7.4.

Ao final do processo de seleção de atributos, os sensores escolhidos para compor o conjunto de dados de entrada do modelo de previsão estão descritos na Tabela 7.1.

Tabela 7.1: Sensores selecionados ao final no processo de seleção de atributos.

Sensor	Descrição
P74_BE-5147501A-A	Sensor de chama
P74_PDIT-5147520A	Diferencial de pressão de admissão de ar
P74_PDIT-5147522A-B	Diferencial de pressão do duto de exaustão
P74_PIT-5147530A-A	Pressão de óleo sintético
P74_PT-5147502A-A	Sensor de pressão de gás combustível a montante da válvula metering de gás
P74_PT-5147503A-A	Sensor de pressão de gás combustível a jusante da válvula metering de gás
P74_TE-5147508A-A	Temperatura dos gases de exaustão da GG - T48
P74_TE-5147509A-A	Temperatura dos gases de exaustão da PT - T8
P74_TE-5147509A-C	Temperatura dos gases de exaustão da PT - T8
P74_TE-5147521A-A	Temperatura do header de óleo mineral
P74_TE-5147522A-A	Temperatura de óleo sintético AGB
P74_TE-5147523A-A	Temperatura de óleo sintético do header de óleo sintético

P74_TE-5147524A-A	Temperatura de óleo sintético do header de óleo
P74_VT-5147501A	sintético
P74_VT-5147502A	Sensor de vibração do compressor da PT
P74_ZT-5147503A-A	Sensor de posição das VSV

Após a definição dos sensores a serem utilizados na análise de falhas, ocorreu-se o processo de construção do *DataFrame* com os valores a serem alimentados pelo modelo. A ideia do projeto é realizar uma análise das medições realizadas em passos de tempo de 1 minuto dos sensores em períodos anteriores a falha e em condições normais de operação.

Nesse sentido, foi construído um *dataset* com registros de 9 horas antes da falha no turbogerador numa granularidade de 1 minuto. Também foram obtidas sequências com intervalos de tempo de mesmo comprimento, mas para condições normais de operação.

No intuito de realizar uma comparação entre as leituras dos sensores em intervalos de tempo próximos à falha (F) e em condições normais (N), plotou-se a distribuição das leituras dos atributos em uma escala de 0 a 1 por meio do gráfico de violino para cada sensor (Figura 7.5).

A Figura 7.5 evidencia que os dados em situação de falha e em condições normais apresentam, de modo geral, distribuições distintas. Essa característica indica que os dados coletados são bons indicadores de falha em sistemas de classificação de séries temporais.

Os dados advindos dos sensores apresenta um desbalanceamento de classe, isto é, há muitos registros do turbogerador em condições normais de operação quando em comparação com os dados de falha. Diante desse contexto, foi desenvolvida uma base de dados artificial para condições anteriores a falta para lidar com esse desbalanceamento. Para isso, utilizou-se de um processo de previsão de séries temporais multivariadas por meio do modelo VAR descrito no Capítulo 4.

A Figura 7.6 ilustra uma instância da base de dados artificial, com valores correspondentes aos 16 sensores num período de 9 horas antes do momento da falha.

As previsões do modelo VAR foram avaliadas por meio do cálculo do MAPE (do inglês, *Mean Absolute Percentage Error*) entre as previsões e os valores reais de cada sensor. Essa métrica foi escolhida tendo em vista que as séries temporais apresentam escalas distintas e

Figura 7.3: Matriz de correlação linear entre os 32 sensores pré-selecionados.

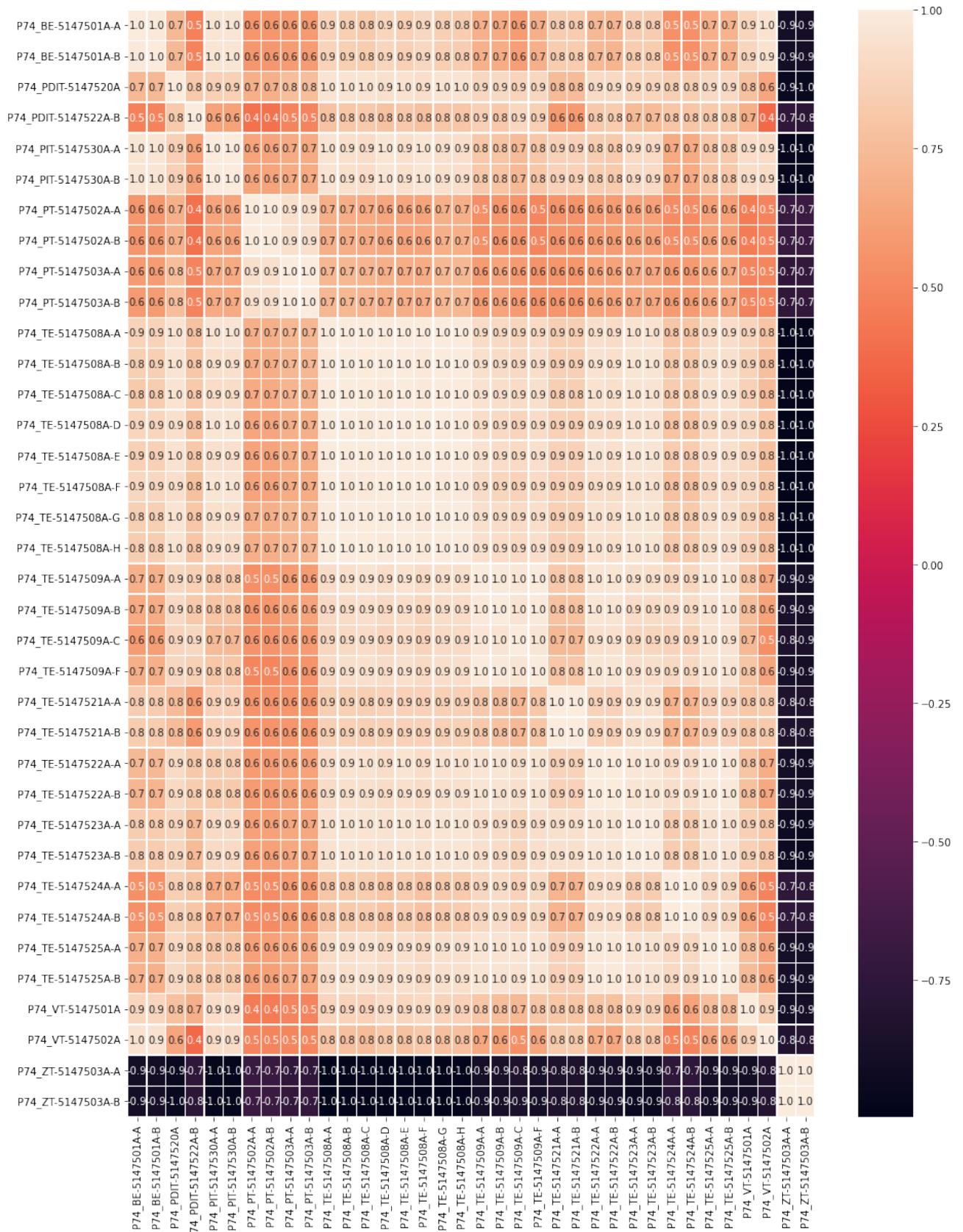
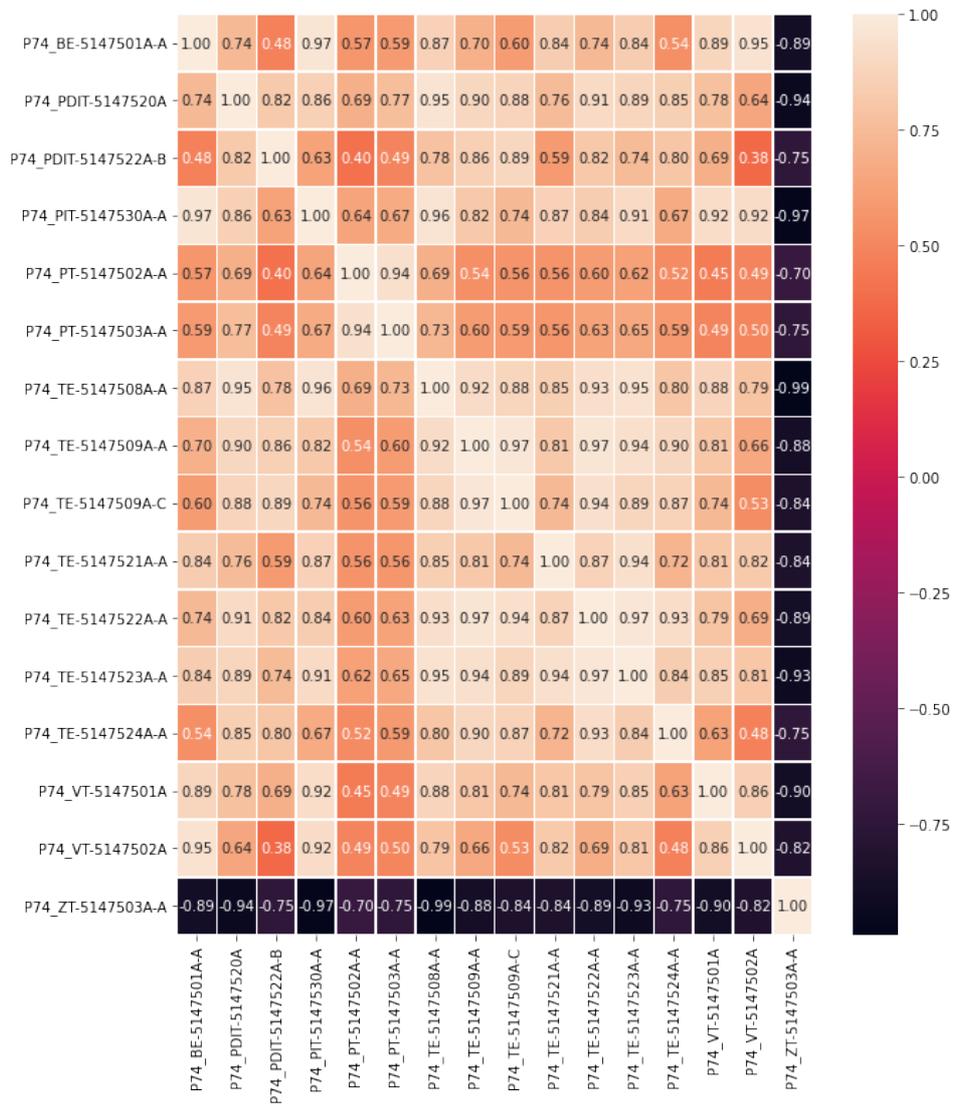


Figura 7.4: Matriz de correlação linear entre os sensores selecionados.



Fonte: Autoria própria.

Figura 7.5: Gráfico de violino para cada um dos sensores selecionados

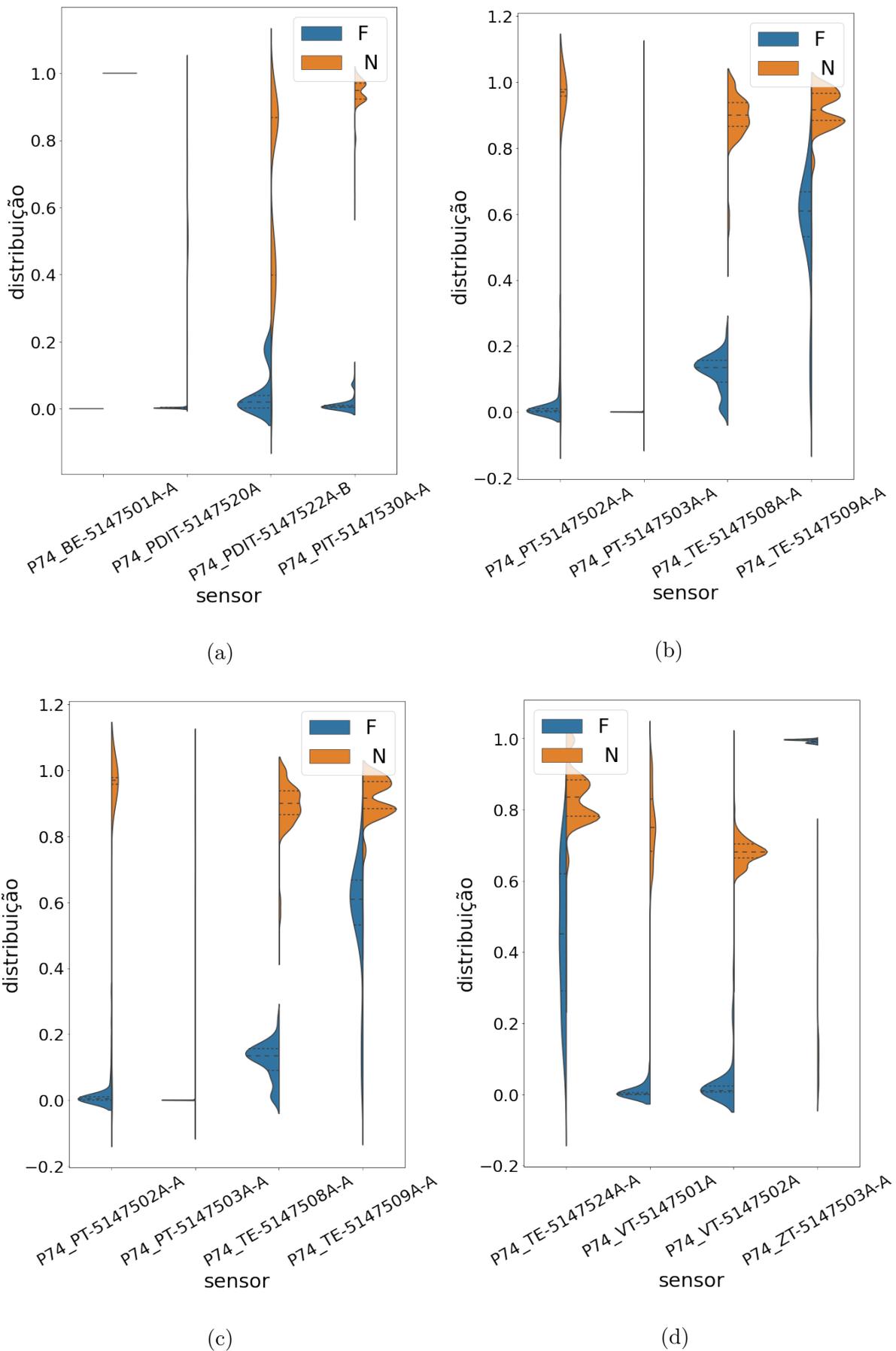
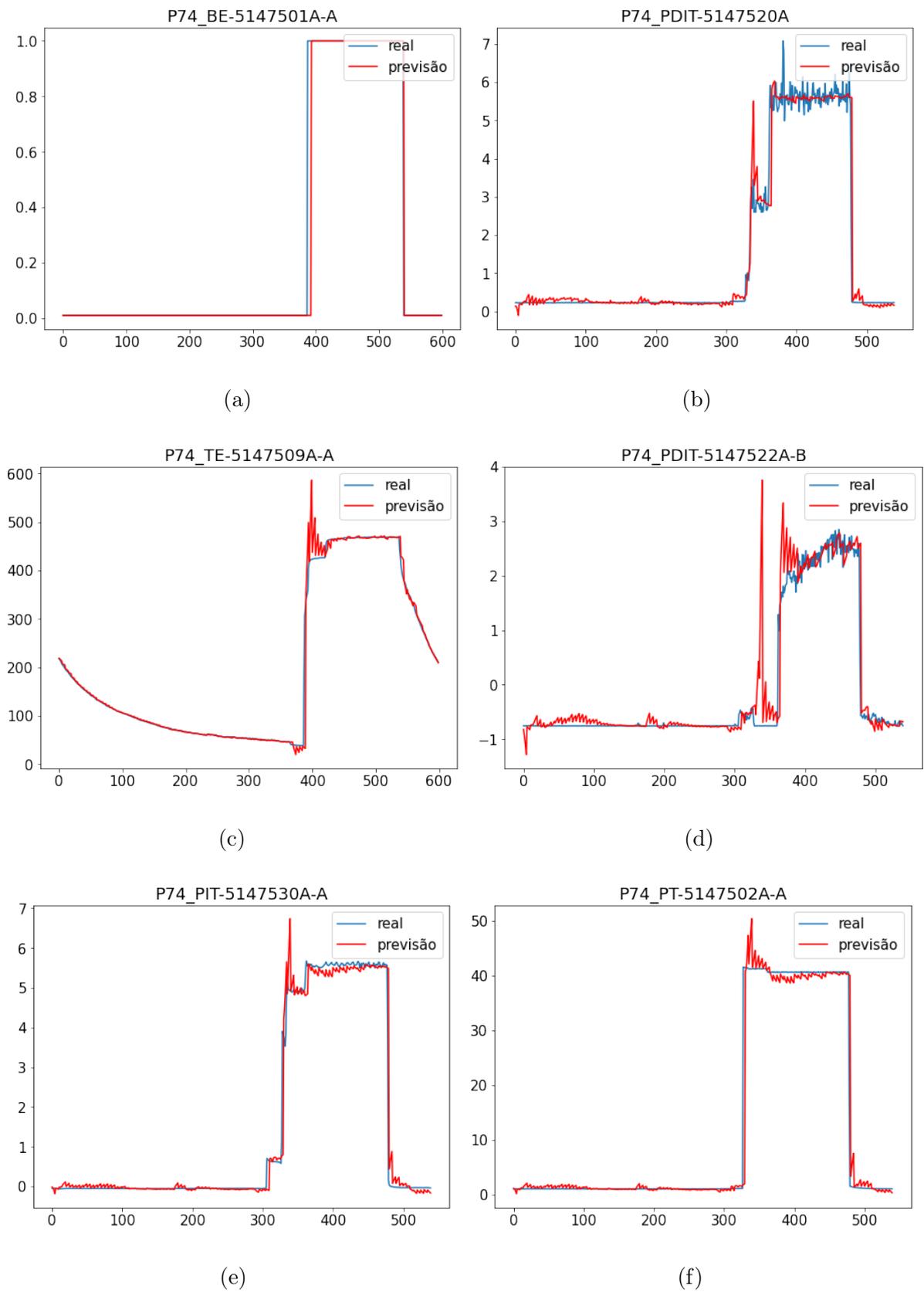
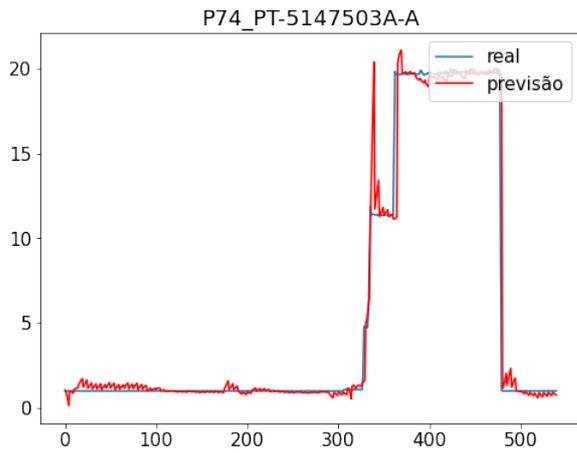
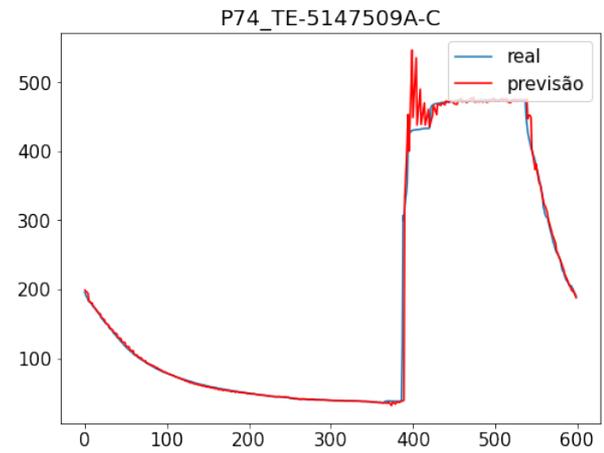


Figura 7.6: Leitura dos sensores para uma instância da base de dados artificial

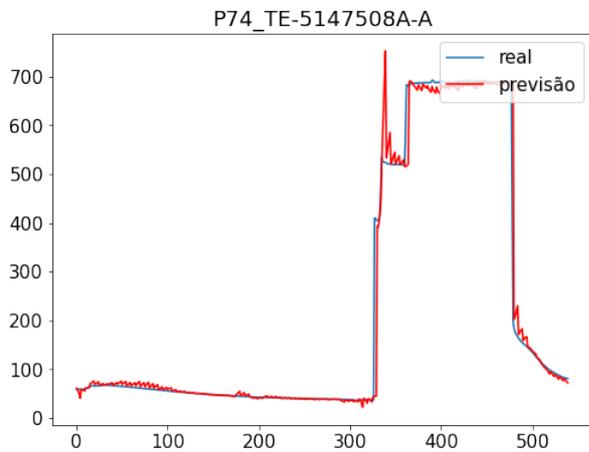




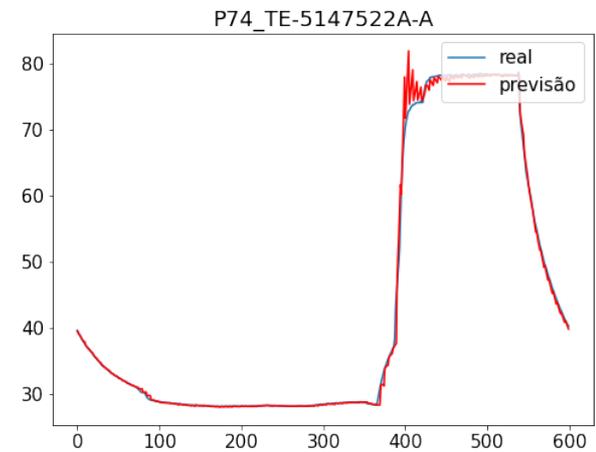
(g)



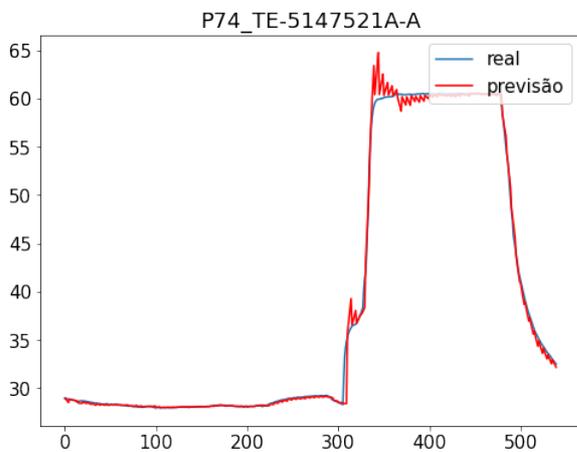
(h)



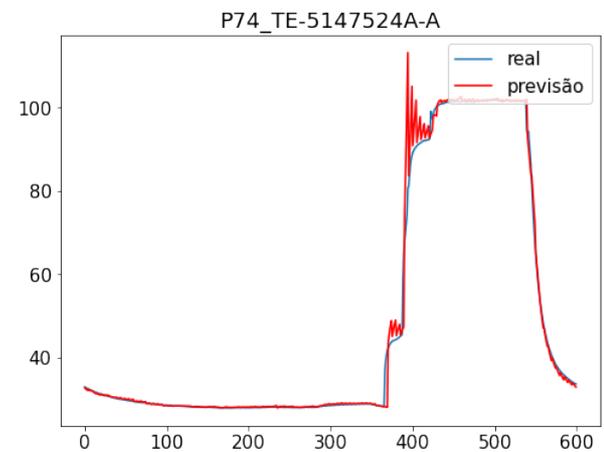
(i)



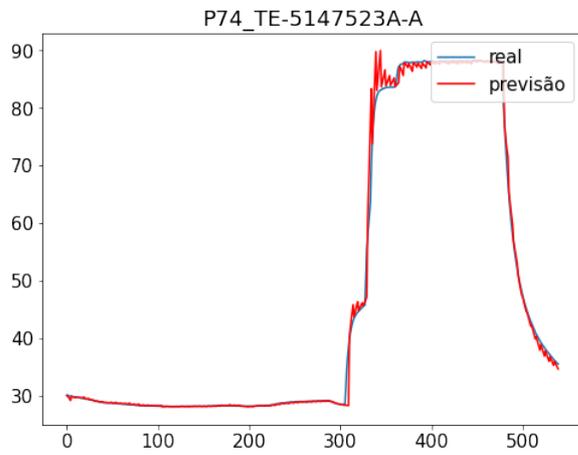
(j)



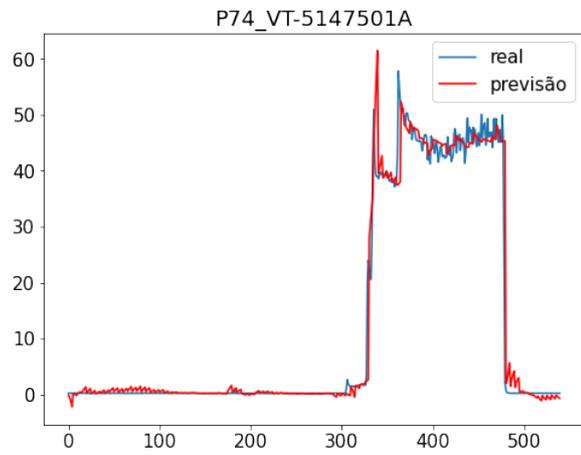
(k)



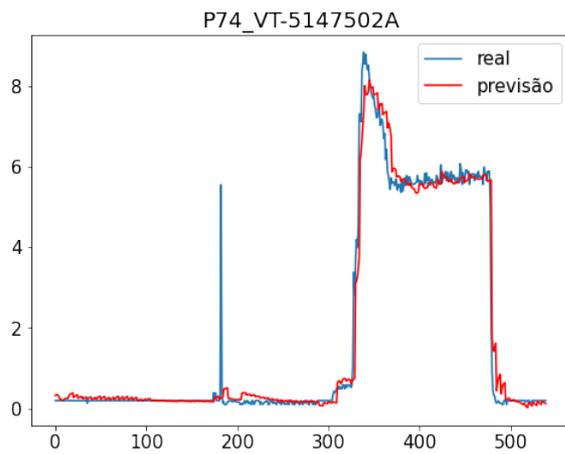
(l)



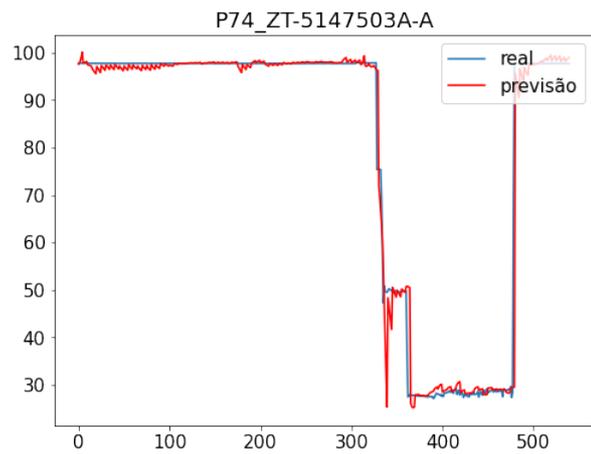
(m)



(n)



(o)



(p)

Fonte: Autoria própria.

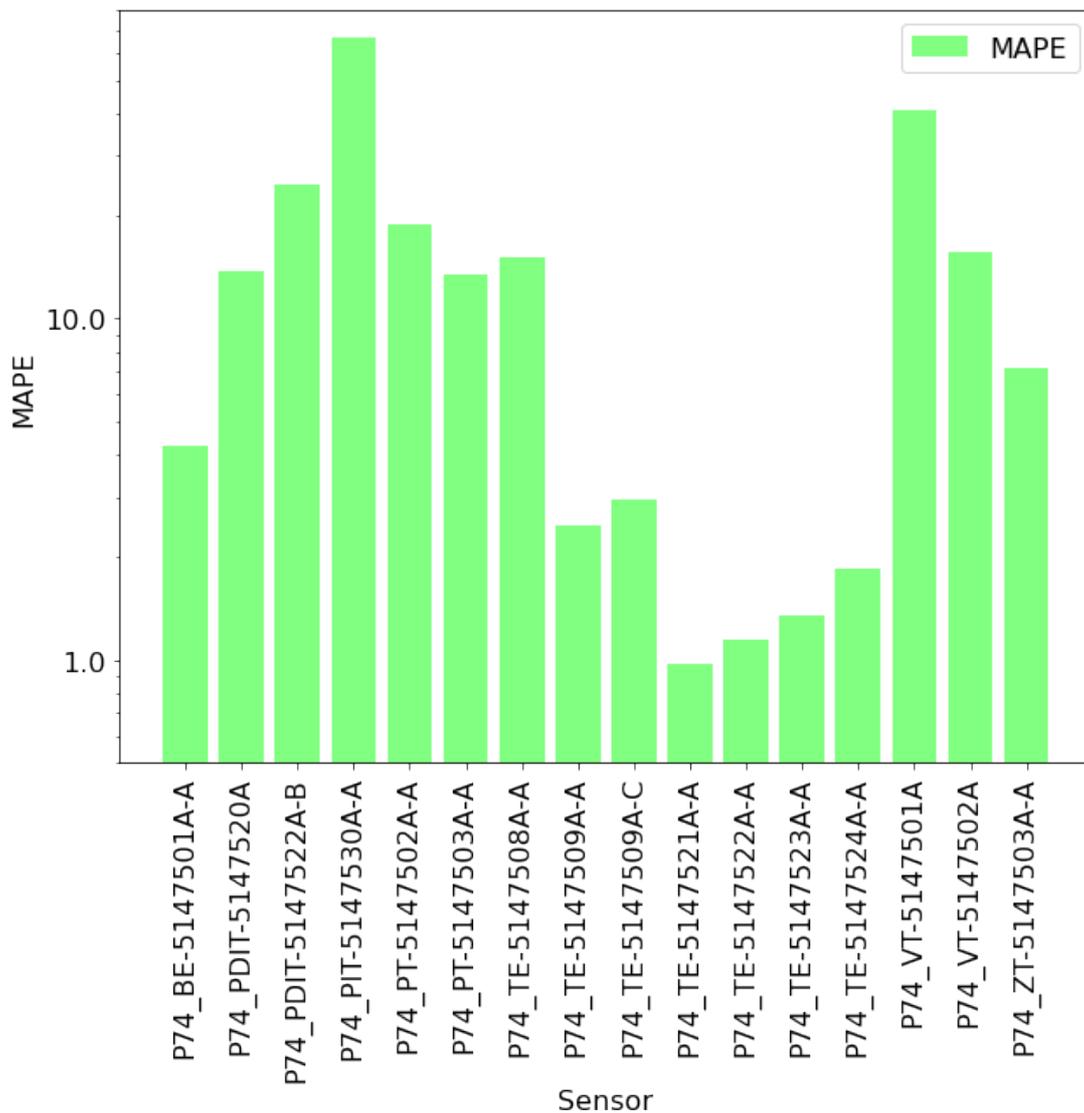
o MAPE não depende da escala. O cálculo do MAPE é mostrado conforme a Equação 7.1.

$$\text{MAPE} = \frac{1}{n} \sum_{t=1}^n \left| \frac{R_t - P_t}{R_t} \right| \cdot 100\%, \quad (7.1)$$

em que n é o número de observações, R_t é o valor real no tempo t e P_t é o valor previsto.

A Figura 7.7 ilustra o MAPE (%) médio obtido para cada sensor nas previsões pelo modelo VAR.

Figura 7.7: MAPE (%) referente a previsão do modelo VAR para cada sensor



Autoria Própria

7.2 Modelo de detecção de falhas

Para o modelo de detecção de falhas desenvolvido implementou-se um modelo de classificação baseado em *deep learning*, mas especificamente em LSTM, cuja explicação teórica está descrita no Capítulo 5.

De modo a agilizar o tempo de processamento, o modelo foi treinado com uma GPU tipo NVIDIA Tesla K80 disponibilizada na plataforma *colab* da google.

Os dados foram divididos em treinamento, validação e teste, de modo que 35% das sequências fossem utilizadas para teste, 10% para validação e 55% para o treinamento do modelo.

Os parâmetros da rede LSTM foram definidos de modo que houvessem três camadas de neurônios, com 64 neurônios totais nas camadas escondidas. O número de amostras que serão propagadas pela rede (também chamado de *batch size*) foi configurado para 16. Como critério de parada, foi estabelecido que o número de ciclos/épocas a serem repetidos (*epochs*) seria 200.

Os valores do MSE em cada época foram salvos para os dados de treinamento e validação. Esses valores podem ser vistos nas Figura 7.8 e Figura 7.9.

Após a etapa de treinamento, o modelo de detecção de falhas foi alimentado com os dados de teste. O modelo apresentou 100% de acerto para com esses dados, resultando em previsões bastante acuradas. A Tabela 7.2 descreve as métricas de avaliação do modelo LSTM implementado.

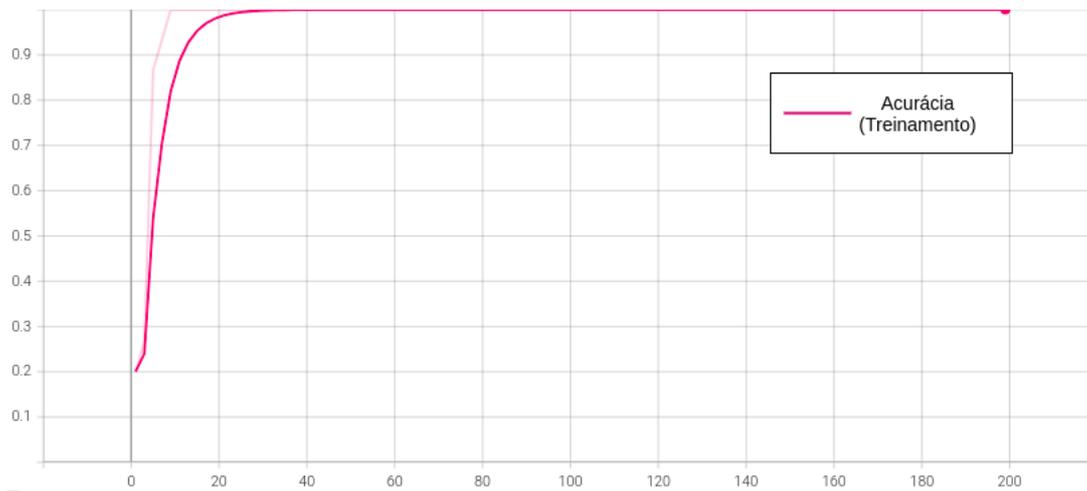
Tabela 7.2: Avaliação quantitativa do modelo de detecção de falhas

	<i>precision</i>	<i>recall</i>	<i>f1-score</i>	<i>support</i>
Normal	1.00	1.00	1.00	8
Falha	1.00	1.00	1.00	7
acurácia			1.00	15

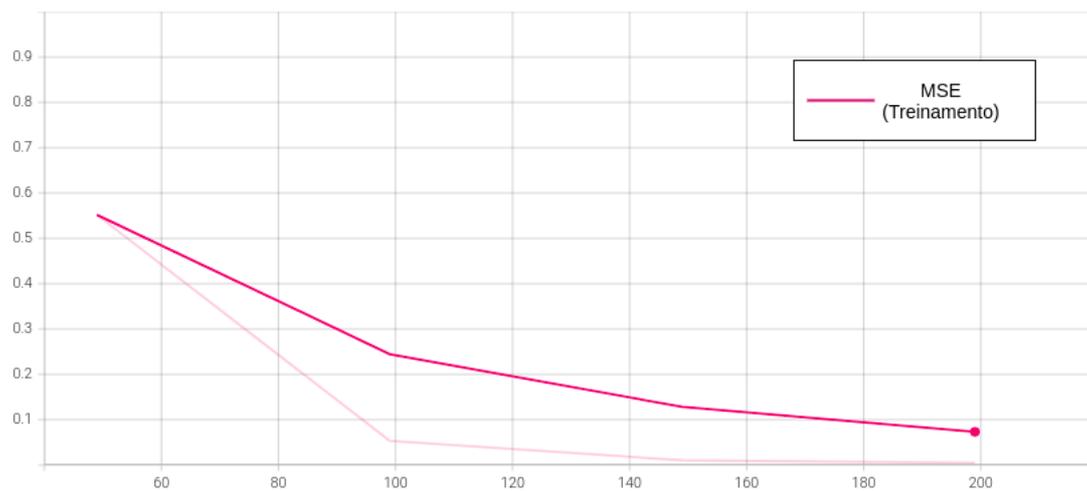
Por fim, no intuito de melhor visualizar a qualidade do modelo de previsão, foram plotadas a matriz de confusão e a curva ROC dos resultados

Conforme percebido na matriz de confusão o modelo classificou corretamente todas as séries temporais correspondentes tanto a períodos imediatamente anteriores a falhas, quanto em condições normais de operação.

Figura 7.8: Acurácia (a) e MSE (b) para os dados de treinamento ao longo das épocas



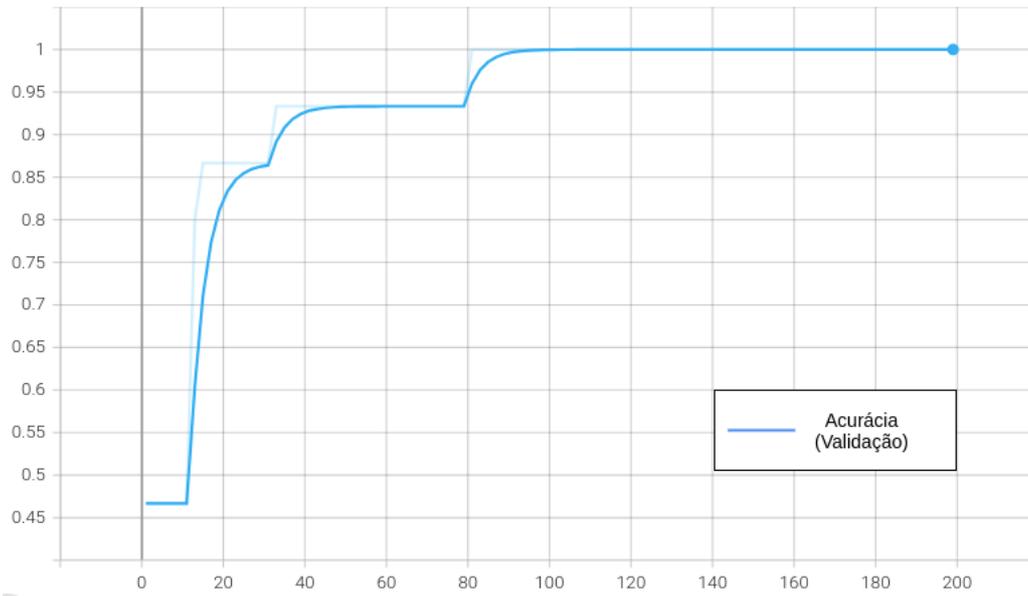
(a)



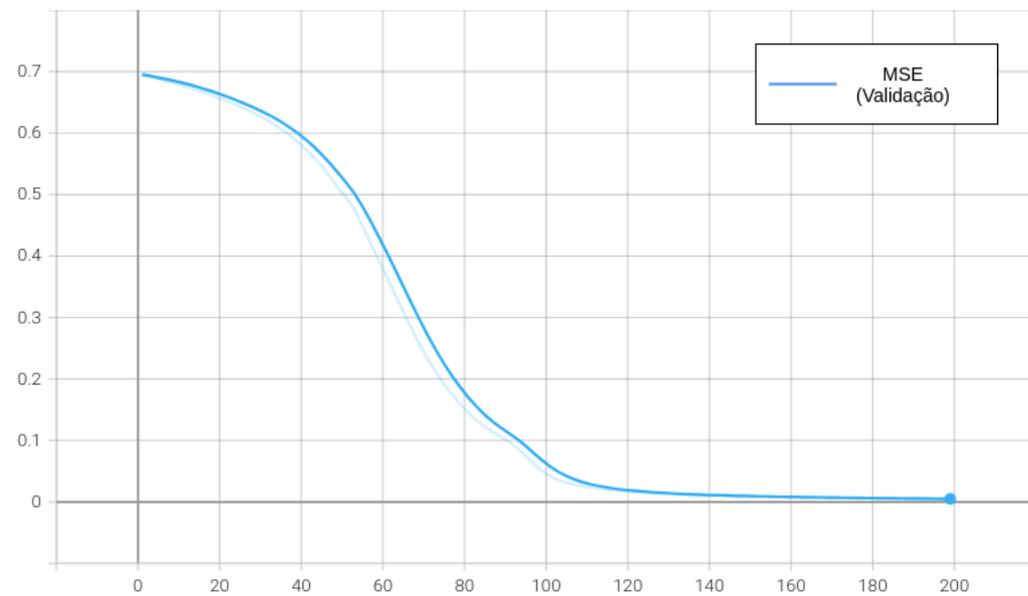
(b)

Autoria Própria

Figura 7.9: Acurácia (a) e MSE (b) para os dados de validação ao longo das épocas



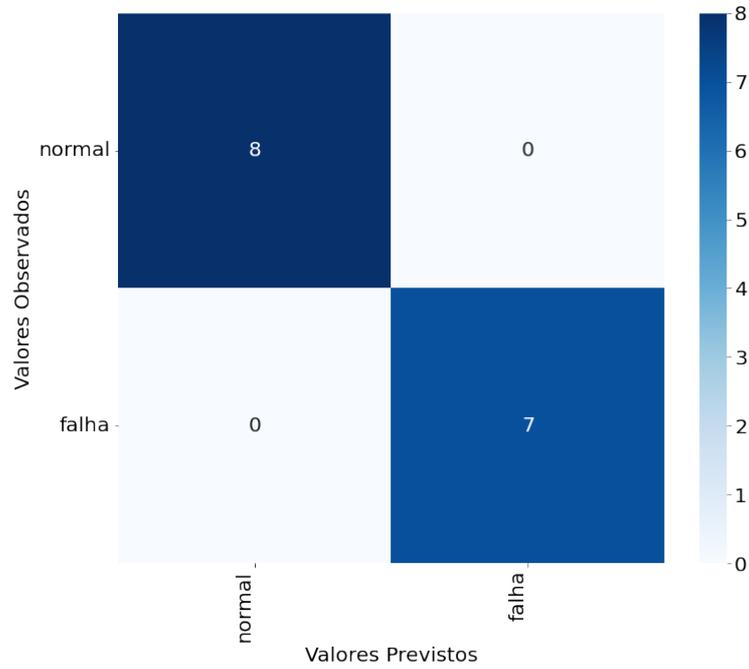
(a)



(b)

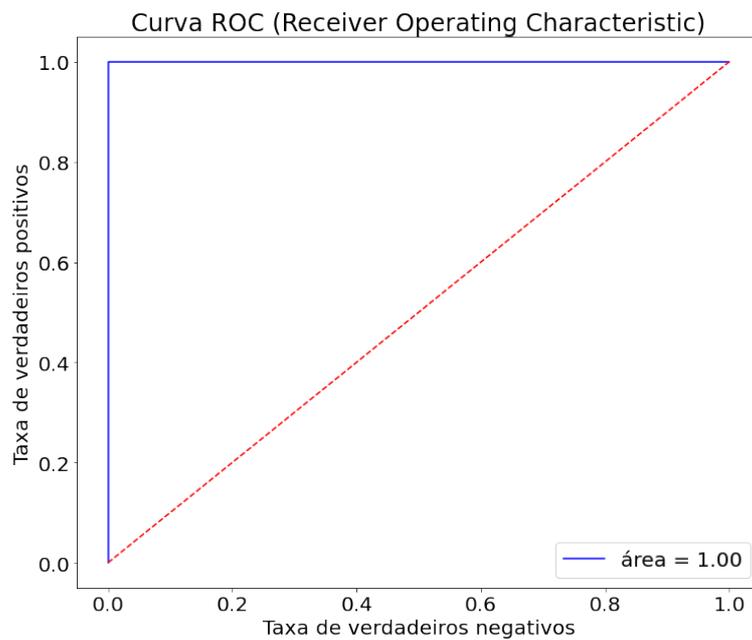
Autoria Própria

Figura 7.10: Matriz de confusão com os valores de teste do modelo de detecção de falhas.



Fonte: Autoria própria.

Figura 7.11: Curva ROC do modelo



Fonte: Autoria própria.

A curva ROC mostra o quão bom o modelo criado pode distinguir entre duas classes (0 ou 1, positivo e negativo, falha ou normal). Os modelos mais acurados conseguem distinguir com precisão o binômio.

Essa curva apresenta dois parâmetros: taxa de verdadeiro positivo (TPR, do inglês *True Positive Rate*) e a taxa de falso positivo (FPR, do inglês *False Positive Rate*).

$$\text{TPR} = \frac{TP}{TP + FN} \quad (7.2)$$

$$\text{FPR} = \frac{FP}{FP + TN} \quad (7.3)$$

Uma curva ROC mostra a relação entre “TPR vs. FPR” em diferentes limiares de classificação. A área impressa no gráfico da Figura 7.11 representa uma maneira de resumir a curva ROC em um único valor, agregando todos os limiares da ROC e calculando a “área sob a curva”. Quanto mais próxima de 1 for a área, mais acurado é o modelo.

7.3 Considerações finais

Neste Capítulo foram apresentados os resultados obtidos nas etapas de pré-processamento dos dados brutos e construção do modelo de detecção de falhas. Cabe destacar a importância da fase de pré-processamento para o alcance de modelos mais acurados. Por meio de ferramentas gráficas, como o gráfico de barras de informação mútua, pôde-se perceber as variáveis de maior importância para o desenvolvimento do modelo. A matriz de correlação linear entre as variáveis evidenciou a redundância de informações nos dados coletados pelos sensores, auxiliando no processo de seleção de atributos. Ademais, o gráfico de violino mostrou a distinção na distribuição dos dados entre as duas classes, antecipando que essas séries temporais poderiam ser usadas na alimentação de modelos de classificação acurados. Na etapa de desenvolvimento do modelo, foi implementada uma LSTM treinada com GPU que apresentou resultados acurados na detecção de falhas do turbogerador, apresentando 100% de acerto para os dados de teste.

Capítulo 8

Conclusão

Neste trabalho de conclusão de curso, foram discutidas as etapas de desenvolvimento de um sistema de detecção de falhas em turbogeradores de uma plataforma tipo FPSO baseado na análise de séries temporais multivariadas. Por meio dessa documentação é possível compreender as etapas constituintes do pré-processamento dos dados dos sensores de monitoramento, além do conhecimento teórico por trás de algoritmos de DL, mais especificamente LSTM em modelos de CST.

Plataformas tipo FPSO apresentam grandes vantagens econômicas e práticas diante da exploração de petróleo em águas cada vez mais profundas e em posições muito longe do continente. Nos campos de petróleo onde a instalação de infra-estrutura é bastante cara, a utilização de sistemas de FPSOs acaba sendo uma alternativa mais atraente e, muitas vezes, a única alternativa plausível. Outra vantagem está relacionado ao fato de que, em geral, a plataforma trabalha em conjunto com uma plataforma de petróleo/gás ou um modelo de submarino, processando e armazenando o óleo/gás para descarga sempre que for necessário.

A plataforma FPSO é totalmente dependente do seu sistema de geração de energia para a alimentação do *topside* e do casco, por isso a importância do monitoramento desse tipo de sistema e detecção de possíveis falhas. Ao longo da pesquisa, foi analisado o turbogerador que apresentou maior registro de falhas dentre os quatro turbogeradores que compõem o sistema de geração principal da plataforma.

Durante a exploração dos dados disponíveis, foram analisados dados operacionais e registros de sensores de monitoramento desse turbogerador. Na etapa de pré-processamento dos

dados brutos advindos dos sensores foram desenvolvidas atividades relacionadas a preparação, organização e estruturação dos dados. A etapa de pré-processamento é determinante para a qualidade final dos dados que irão alimentar o modelo de detecção de falhas.

A partir da análise da correlação linear entre os atributos (sensores) e da informação mútua entre os atributos (sensores) e o vetor de ocorrência de falhas, foi possível diminuir a dimensionalidade das *features* de 55 para 16. Dessa forma, selecionou-se as variáveis de maior impacto, simplificando os cálculos do modelo de detecção e eliminando informações redundantes e não relevantes.

Com a investigação dos dados, observou-se um desbalanceamento entre registros de falhas e não falhas. Sendo assim, foi construída uma base de dados artificial, introduzindo novos registros de falha por meio do modelo VAR para previsão de STs multivariadas.

Após o pré-processamento dos dados, foi possível o desenvolvimento do sistema de detecção de falhas por meio da implementação de rede LSTM em uma GPU, no intuito de acelerar o processamento. Essa aceleração no treinamento do modelo foi constatada ao longo dos testes realizados nos dois ambientes. Ao final do treinamento, o modelo foi testado com novas variáveis de entrada, apresentando 100% de acerto de classificação do estado do turbogerador (falha ou condição normal de operação). Nesse sentido, comprovou-se, na análise quantitativa, o bom desempenho do modelo de predição, que se mostrou bastante acurado, apresentando a área máxima na curva ROC.

8.1 Trabalhos futuros

Como sugestão de trabalhos futuros na linha de pesquisa, é possível citar:

- Desenvolvimento de modelos de similaridade ou regressão para estimar a vida útil do equipamento, identificando em quanto tempo o mesmo iria falhar;
- Aquisição de mais dados dos sensores para conseguir distinguir diferentes causas de falha no turbogerador por meio da associação entre modelos de previsão de séries temporais e sistemas especialistas.

Referências bibliográficas

- 1 CASTRO, Patrício Fernandes de et al. *Análise de confiabilidade do sistema elétrico de uma plataforma de produção de petróleo e gás*. Dissertação (Mestrado) — Universidade Federal da Paraíba, 2020.
- 2 BRASIL ENERGY INSIGH. *New Petrobras FPSO Contracts in October*. ., 2019. Disponível em: <<https://brazilenergyinsight.com/2019/09/23/new-petrobras-fpso-contracts-in-october/>>.
- 3 MEDEIROS, Juliana Pegado de. *Estudo e implementação de algoritmos inteligentes para detecção e classificação de falhas na medição de gás natural*. Dissertação (Mestrado) — Universidade Federal do Rio Grande do Norte, 2009.
- 4 HUANG, Hao; YOO, Shinjae. *Failure Analysis on Multivariate Time-series Data given Uncertain Labels*. [S.l.], 2019.
- 5 LIMA, Felipe Mazurek dos Santos Pereira. Mapeamento e avaliação preliminar da simulação de um sistema de manutenção para turbogeradores em uma fpso. *Trabalho de Conclusão de Curso*, 2019.
- 6 SILVA, ANIELLE MARTINS DA. Unidades flutuantes de produção, armazenagem e transferência de óleo : Conversão de navios em plataformas tipo fpso. *Trabalho de Conclusão de Curso*, 2011.
- 7 MEDEIROS, Victor Alves de. O estado da arte do fpso. *Projeto de graduação, Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ, Brasil*, 2015.
- 8 GROVE, Marcio de Abreu. *Sistema de posicionamento híbrido para FPSOs*. Tese (Doutorado) — Universidade Federal do Rio de Janeiro, 2005.
- 9 PEREIRA, José Antônio Maciel; JESUS, João Damasceno de. Caracterização dos sistemas de geração elétrica dos fpsos em operação no brasil. *Engevista*, v. 17, n. 3, p. 433–443, 2015.
- 10 REMBE. *Rupture discs for FPSO applications*. 2019. Acessado em: 2021-09-10. Disponível em: <<https://www.rembel.com/applications/pressure-relief-for/fpso/>>.
- 11 COÊLHO, Lucas Rodrigues. *Análise exploratória e correlação das variáveis de lingo-tamento contínuo de uma siderúrgica*. 2019.

- 12 RADEČÍČ, Dario. *Violin Plot — It's Time to Ditch the Box Plots*. 2019. Acessado em: 2021-09-14. Disponível em: <<https://towardsdatascience.com/violin-plot-its-time-to-ditch-the-box-plots-785629b0ff3a>>.
- 13 KOTSIANTIS, Sotiris B; KANELLOPOULOS, Dimitris; PINTELAS, Panagiotis E. Data preprocessing for supervised learning. *International journal of computer science*, Citeseer, v. 1, n. 2, p. 111–117, 2006.
- 14 GUYON, Isabelle; ELISSEEFF, André. An introduction to variable and feature selection. *Journal of machine learning research*, v. 3, n. Mar, p. 1157–1182, 2003.
- 15 ZIVOT, Eric; WANG, Jiahui. Vector autoregressive models for multivariate time series. *Modeling Financial Time Series with S-Plus®*, Springer, p. 385–429, 2006.
- 16 SANTOS, Alexandre José dos. *Modelos Vetoriais Auto-Regressivos com Transição Suave Estruturados por Árvores-STVAR-Tree*. Tese (Doutorado) — PUC-Rio, 2009.
- 17 YANG, Qiang; WU, Xindong. 10 challenging problems in data mining research. *International Journal of Information Technology & Decision Making*, World Scientific, v. 5, n. 04, p. 597–604, 2006.
- 18 FAWAZ, Hassan Ismail; FORESTIER, Germain; WEBER, Jonathan; IDOUMGHAR, Lhassane; MULLER, Pierre-Alain. Deep learning for time series classification: a review. *Data mining and knowledge discovery*, Springer, v. 33, n. 4, p. 917–963, 2019.
- 19 PRA, Marco Del. *Time Series Classification with Deep Learning*. 2020. Acessado em: 2021-09-10. Disponível em: <<https://towardsdatascience.com/time-series-classification-with-deep-learning-d238f0147d6f>>.
- 20 KAUSHIK, Shruti; CHOUDHURY, Abhinav; SHERON, Pankaj Kumar; DASGUPTA, Nataraj; NATARAJAN, Sayee; PICKETT, Larry A; DUTT, Varun. Ai in healthcare: time-series forecasting using statistical, neural, and ensemble architectures. *Frontiers in big data*, Frontiers, v. 3, p. 4, 2020.
- 21 GREFF, Klaus; SRIVASTAVA, Rupesh K; KOUTNÍK, Jan; STEUNEBRINK, Bas R; SCHMIDHUBER, Jürgen. Lstm: A search space odyssey. *IEEE transactions on neural networks and learning systems*, IEEE, v. 28, n. 10, p. 2222–2232, 2016.
- 22 PAI, Aravind. *CNN vs. RNN vs. ANN — Analyzing 3 Types of Neural Networks in Deep Learning*. 2020. Acessado em: 2021-09-10. Disponível em: <<https://www.analyticsvidhya.com/blog/2020/02/cnn-vs-rnn-vs-mlp-analyzing-3-types-of-neural-networks-in-deep-learning/>>.
- 23 YU, Yong; SI, Xiaosheng; HU, Changhua; ZHANG, Jianxun. A review of recurrent neural networks: Lstm cells and network architectures. *Neural computation*, MIT Press One Rogers Street, Cambridge, MA 02142-1209, USA journals-info . . . , v. 31, n. 7, p. 1235–1270, 2019.
- 24 HOCHREITER, Sepp; SCHMIDHUBER, Jürgen. Long short-term memory. *Neural computation*, MIT Press, v. 9, n. 8, p. 1735–1780, 1997.

- 25 SVITLA. *CPU, GPU, and TPU for fast computing in machine learning and neural networks*. 2020. Acessado em: 2021-09-12. Disponível em: <<https://svitla.com/blog/cpu-gpu-and-tpu-for-fast-computing-in-machine-learning-and-neural-networks>>.
- 26 CAO, Qingqing; BALASUBRAMANIAN, Niranjana; BALASUBRAMANIAN, Aruna. Mobirnn: Efficient recurrent neural network execution on mobile gpu. In: *Proceedings of the 1st International Workshop on Deep Learning for Mobile Systems and Applications*. [S.l.: s.n.], 2017. p. 1–6.