SENSORES VIRTUAIS BASEADOS EM APRENDIZADO DE MÁQUINA PARA POÇOS DE PETRÓLEO

ARTHUR VICTOR DOS SANTOS ALVES

PROJETO DE GRADUAÇÃO EM ENGENHARIA MECÂNICA
DEPARTAMENTO DE ENGENHARIA MECÂNICA

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA

UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA MECÂNICA

SENSORES VIRTUAIS BASEADOS EM APRENDIZADO DE MÁQUINA PARA POÇOS DE PETRÓLEO

ARTHUR VICTOR DOS SANTOS ALVES

Orientador: PROF. DR. ADRIANO TODOROVIC FABRO, ENM/UNB

PROJETO DE GRADUAÇÃO EM ENGENHARIA MECÂNICA

PUBLICAÇÃO ENM.PG - XXX/AAAA
BRASÍLIA-DF, 16 DE FEVEREIRO DE 2023.

UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA MECÂNICA

SENSORES VIRTUAIS BASEADOS EM APRENDIZADO DE MÁQUINA PARA POÇOS DE PETRÓLEO

ARTHUR VICTOR DOS SANTOS ALVES

PROJETO DE GRADUAÇÃO SUBMETIDO AO DEPARTAMENTO DE ENGENHARIA MECÂNICA DA FACULDADE DE TECNOLOGIA DA UNIVERSIDADE DE BRASÍLIA, COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE ENGENHEIRO MECÂNICO.

BANCA EXAMINADORA:

Prof. Dr. Adriano Todorovic Fabro, ENM/UnB
Orientador

Prof. Dr. Alberto Carlos Guimarães Castro Diniz, ENM/UnB
Examinador interno

Prof. Dr. Marcela Rodrigues Machado, ENM/UnB
Examinador interno

BRASÍLIA, 16 DE FEVEREIRO DE 2023.

FICHA CATALOGRÁFICA
ARTHUR VICTOR DOS SANTOS ALVES
SENSORES VIRTUAIS BASEADOS EM APRENDIZADO DE MÁQUINA PARA POÇOS DE PETRÓLEO
2023xv, 147p., 201x297 mm
(ENM/FT/UnB, Engenheiro Mecânico, Engenharia Mecânica, 2023)
Projeto de Graduação - Universidade de Brasília
Faculdade de Tecnologia - Departamento de Engenharia Mecânica

REFERÊNCIA BIBLIOGRÁFICA

ARTHUR VICTOR DOS SANTOS ALVES (2023) SENSORES VIRTUAIS BASEADOS EM APRENDIZADO DE MÁQUINA PARA POÇOS DE PETRÓLEO. Projeto de Graduação em Engenharia Mecânica, Publicação xxx/AAAA, Departamento de Engenharia Mecânica, Universidade de Brasília, Brasília, DF, 147p.

CESSÃO DE DIREITOS

AUTOR: Arthur Victor dos Santos Alves
TÍTULO: SENSORES VIRTUAIS BASEADOS EM APRENDIZADO DE MÁQUINA PARA POÇOS DE PETRÓLEO.
GRAU: Engenheiro Mecânico ANO: 2023

Arthur Victor dos Santos Alves
vsaarthur@gmail.com

Agradecimentos

Em primeiro lugar, agradeço à minha família, ao meu pai Raimundo, à minha mãe Maria do Socorro, à minha irmã Mayara e à minha irmã Amanda. Muito obrigado por todo suporte, amor e pela batalha diária que me dá liberdade e suporte para focar na faculdade e no trabalho. Vocês são essenciais para mim e serão eternamente minha fonte de inspiração. Em especial, aos meus pais, serei, para sempre, grato por toda força e por todas as dificuldades que vocês tiveram e têm que superar para dar as oportunidades que eu e minhas irmãs possuímos.

Em segundo lugar, agradeço à minha namorada Júlia pela paciência, pelo companheirismo e por tudo mais, que fazem eu acreditar e me dão energias para continuar.

Agradeço a cada um dos meus amigos mais próximos que de alguma maneira fizeram parte dessa caminhada e contribuíram de alguma forma. Em particular, gostaria de agradecer aos meus amigos Diogo e Juarez, companheiros desde a escola, que me acompanharam nessa jornada da Engenharia Mecânica e que deixaram ela mais fácil, mais leve e mais divertida. Parte do meu sucesso na faculdade eu devo a eles.

Quero agradecer ao meu Orientador Prof. Dr. Adriano Todorovic Fabro, primeiro, por oferecer um tema que é tão divertido de desenvolver, com um foco que é mais difícil de encontrar na Engenharia Mecânica, o qual se encaixa com o meu perfil. Segundo, por, de fato, realizar a atividade de orientar com qualidade. Por fim, pela paciência durante o desenvolvimento do projeto.

E à Deus.

Resumo

O presente trabalho tem como objetivo o desenvolvimento de um sensor virtual no contexto de um gêmeo digital para prognóstico de anomalias em poços de petróleo com base em algoritmos de Aprendizado de Máquina. Para isso, dividiu-se o problema em duas abordagens, a primeira de classificação e a segunda de previsão. Na abordagem de classificação, inicialmente são reproduzidos resultados obtidos por outros trabalhos para, então, avançar com teste de outras metodologias e com otimização dos resultados. Os modelos testados nessa abordagem foram Floresta Aleatória, o XGBoost, a Floresta de Isolamento, o *Local Outlier Factor*, o Envelope Elíptico e o *One Class* SVM com *kernels* linear, RBF, POLY e SIGMOID. Na abordagem de previsão, aplica-se um modelo LSTM, *Long Short-Term Memory*, para prever dados em instantes futuros. Ao fim, os modelos de detecção e de previsão são mesclados para produzir o prognóstico de eventos indesejados. Os resultados obtidos foram positivos para a detecção de anomalias, sendo o *Local Outlier Factor* e a Floresta Aleatória os modelos não supervisionado e supervisionado, respectivamente, que melhor performaram. Entretanto, a previsão de dados apresenta pontos positivos e negativos nos resultados. Nesse cenário, os resultados da combinação dos modelos para prognóstico das anomalias mostrou que a mesclagem do modelo não supervisionado não produziu resultados confiáveis. Por outro lado, a mesclagem com o algoritmo supervisionado mostrou que a abordagem performa bem separadamente para dados normais e para dados com anomalia. Por fim, são propostos dois modelos de sensores virtuais, um exclusivamente para detecção de anomalias e outro para prognóstico de anomalias.

Abstract

This work aims to develop a virtual sensor in the context of a digital twin for the prognosis of anomalies in oil wells based on Machine Learning algorithms. In order to do that the problem was splitted into two approaches, the first is the classification approach and the second is the forecasting approach. In the classification approach, initially the results obtained by other works are reproduced, then it proceeds to testing other methodologies and optimizing the results. The models tested in this approach were Random Forest, XGBoost, Isolation Forest, Local Outlier Factor, Elliptical Envelope and One Class SVM with linear, RBF, POLY and SIGMOID kernels. In the forecasting approach, an LSTM model, Long Short-Term Memory, is applied to forecast data at future timesteps. Finally, the detection and forecasting models are blended to make the prognosis of undesirable events. The results obtained were positive for the anomalie detection, Local Outlier Factor and Random Forest were the algorithms, unsupervised and supervised, respectively, that best performed. However, forecasting has positive and negative points in the results. In this scenario, the results of the combination of models for prognosis of anomalies showed that the merge of the unsupervised model did not produce reliable results. On the other hand, the merge with the supervised algorithm showed that the approach performs well separately for normal data and for anomalous data. Finally, two models of virtual sensors are proposed, one exclusively for detecting anomalies and the other for the prognosis of anomalies.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIAÇÕES

PDG	*Permanent Downhole Gauge1*
TPT	*Temperature/pressure transducer*
P-PDG	Pressão do fluido em PDG
P-TPT	Pressão do fluido em TPT
T-TPT	Temperatura do fluido em TPT
P-MON-CKP	Pressão do fluido montante à válvula *choke* de produção
T-JUS-CKP	Temperatura do fluido jusante à válvula *choke* de produção
P-JUS-CKGL	Pressão do fluido jusante à válvula *choke* de *Gas Lift*
T-JUS-CKGL	Temperatura do fluido jusante à válvula *choke* de *Gas Lift*
QGL	Vazão de *Gas Lift*
LOF	*Local Outlier Factor*
STM	Série temporal multivariada
LSTM	*Long Short Term Memory*
MLP	*Multilayer Perceprton*
SVM	*Support Vector Machine*

Capítulo 1

Introdução

A realidade trazida pela indústria 4.0 resultou em progresso para diversas áreas do conhecimento de forma bastante acelerada. Ferramentas como aprendizado de máquina, realidade virtual, internet das coisas, entre outras, permitiram que novas frentes de fenômenos e de processos já conhecidos fossem exploradas e desenvolvidas. Nesse contexto, surge o sensoriamento virtual e o gêmeo digital.

Segundo Mohr (2018), considera-se que um gêmeo digital é "uma tecnologia analítica de dados imersiva que fornece informações sobre interações homem-infraestrutura máquina para permitir que os executivos tomem decisões contextuais". Além disso, um dos principais benefícios e aplicações de um gêmeo digital é a detecção de falhas, monitoramento e predição da qualidade da estrutura ou do processo (Wagg et al. (2020)). Outro modo de interpretar um gêmeo digital é como um sensor virtual, o qual usa dados de sensores físico e algoritmos para gerar *outputs* importantes para o monitoramento seja de uma estrutura seja de um processo (Liu, Kuo e Zhou (2009)). Na literatura recente, as principais aplicações dessa ferramenta são em sistemas dinâmicos, nos quais a modelagem física é complexa, o que faz com que sejam feitas simplificações para a resolução do problema.

No coração da maioria dos gêmeos digitais, estão os algoritmos de aprendizado de máquinas. Eles são uma ferramenta que permite abordar uma gama problemas, por exemplo, problemas de classificação, de regressão, de linguagem natural, de computação visual, entre outros. Para isso, eles fazem uso intensivo de dados a fim de encontrar e aprender padrões, com isso, generalizar soluções para dados nunca antes vistos.

Nesse contexto, a exploração de petróleo engloba uma série de processos que carregam diversos desafios, entre eles estão o monitoramento dos eventos e a detecção de anomalias, isto é, eventos indesejados em poços de petróleos, os quais acarretam em perdas na produção. Uma das formas de abordar esse problema é com a utilização da estrutura de um gêmeo digital. Nesse sentido, é possível utilizar algoritmos de aprendizado de máquinas com três objetivos específicos, os quais são classificar, detectar e prever anomalias. Com isso, cria se um método sistemático de manutenção da vida útil e do funcionamento dos poços de

petróleo.

Com relação ao fenômeno e citado acima, o trabalho de Vargas et al. (2019) resultou em contribuições ricas para o desenvolvimento da metodologia referida anteriormente. O 3W *dataset* reúne dados de oito tipos de eventos indesejados em poços de petróleo, os quais são capturados com janelas de tempo diferentes e em períodos variados.

A partir dos dados do 3W *dataset*, alguns trabalhos foram desenvolvidos a fim de detectar e de classificar as irregularidades nos poços de petróleo. Entretanto, há uma lacuna na área de previsão das anomalias. Com isso, o presente trabalho busca reproduzir os resultados já obtidos das produções, que usam o 3W *dataset* e propor uma metodologia para fazer o prognóstico das anomalias. Vale destacar que anomalias, eventos indesejados e irregularidades nos poços de petróleo são tratados como sinônimos neste trabalho.

1.1 Referencial Teórico

1.1.1 Sensor Virtual e Gêmeo Digital

Ainda não há total consenso do que é um gêmeo digital, vários autores se empenharam em fazer uma revisão de literatura para concatenar as definições e características de um gêmeo digital. Nesse sentido, Wanasinghe et al. (2020) trazem definições de diversas fontes, por exemplo:

- "Modelo virtual e simulado ou uma réplica verdadeira de um ativo físico"(Poddar (2018))
- "Uma tecnologia analítica de dados imersiva que fornece informações sobre interações homem-infraestrutura máquina para permitir que os executivos tomem decisões contextuais"(Mohr (2018))
- "Um modelo baseado na física e nos dados de um sistema ou um ativo que modela todos os vários subsistemas, suas propriedades, a interação entre eles e as interações do sistema com o ambiente"(Saini et al. (2018))
- "Um modelo virtual de um ativo físico"(Sharma et al. (2018))

Ao fim, os autores concatenam as definições na seguinte ideia: "ativo físico, modelo virtual, troca de dados entre um ativo físico e modelo digital, análise e visualização de dados". Com relação a essa ideia, Barricelli, Casiraghi e Fogli (2019) reforçam que um gêmeo digital é diferente de modelos de desenho assistido por computador, de engenharia assistia por computador e de simulações. O que transforma esses produtos em gêmeos digital é a aplicação de inteligência artificial e a troca contínua ou, pelo menos, periódica de dados entre o ativo físico e o modelo virtual. Além disso, Wanasinghe et al. (2020) mostra que a área de

aplicação que mais se destaca é a de monitoramento e manutenção de ativos, que coincide com a do presente projeto.

Ademais, Grieves e Vickers (2016) pontuam dois tipos de gêmeo digital: o protótipo e a instância. O primeiro reúne os informações físicas, por exemplo, o modelo 3D e a lista de materiais, para descrever um protótipo virtual do dispositivo. Além disso, o protótipo não fica conectado ao produto, com isso, ele não evolui com o tempo, isto é, para cada estado do objeto físico, é necessário criar um novo gêmeo digital protótipo. Por outro lado, o gêmeo digital do tipo instância permanece conectado ao ativo durante todo o seu ciclo de vida, ou seja, ele evolui com o tempo. Ademais, cada representação do tipo instância pode conter dados diferentes e funções diferentes, por exemplo, a lista de componentes, o histórico de processos aplicados e os dados capturados de sensores reais, os quais são o foco do presente trabalho. Dessa forma, a junção de todas as instâncias formam o gêmeo digital agregado, o qual realizará operações nas informações agrupadas.

Nesse cenário, Jones et al. (2020) discretizam as definições acima em 13 características dos gêmeos digitais. Entre elas, destaca-se a entidade virtual e os processos virtuais. O gêmeo digital pode ser composto por mais de uma entidade virtual, cada uma com uma função específica, por exemplo, otimização de um processo e monitoramento da integridade do produto físico. Já os processos virtuais são as tarefas realizadas pela entidade virtual no ambiente virtual. A maioria desses processos presentes na literatura estão nas áreas de simulação, de modelagem, de otimização, de monitoramento, de diagnóstico e de previsão. Esses dois conceitos em conjunto, se aproximam da ideia de sensor virtual. Essa segregação das entidades e dos processos de gêmeo digital é mencionada por Sharma et al. (2017) na forma de camada de complexidade, isto é, modelos separados podem ser desenvolvidos para atividades diferentes, adicionando camadas de complexidade.

Sensores virtuais capturam as saídas dos sensores reais e calculam *outputs* mediante o uso modelos (Liu, Kuo e Zhou (2009)). A Fig. 1.1 apresenta a descrição anterior de forma gráfica. O sensoriamento virtual é usado em variadas indústrias e aplicações. Dentro dos conceito de Grieves e Vickers (2016) e de Jones et al. (2020), o sensor virtual é um gêmeo digital do tipo instância, uma entidade virtual, que irá desempenhar um processo virtual específico, o qual será agregado a um todo, isto é, o gêmeo digital agregado de um ativo.

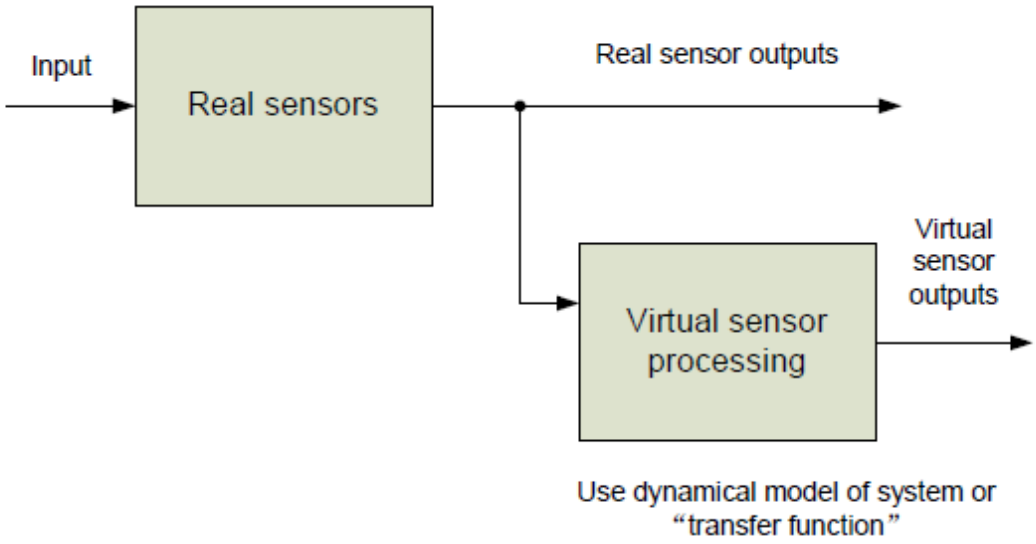

Figura 1.1: Representação de um sensor virtual (Liu, Kuo e Zhou (2009)).

Nesse contexto, Barricelli, Casiraghi e Fogli (2019) destacam outras características que permeiam um gêmeo digital, as quais são o tratamento de dados de alta dimensão e a aplicação de algoritmos supervisionados e não supervisionados de Aprendizado de Máquina. Essas características coincidem com um dos componentes que compõe um gêmeo digital segundo Parrott e Warshaw (2017), que é *Analytics*, isto é, um conjunto de técnicas analíticas usadas pelo gêmeo digital para gerar *insights*.

Min et al. (2019) propõem um *framework* de gêmeo digital baseado em Aprendizado de Máquinas para otimização da produção na indústria petroquímica. O *framework* é composto pelas etapas de coletar dados da linha de produção física em tempo real, utilizar dados históricos e em tempo real para treinar os modelos de Aprendizado de Máquina, validar o modelo, atualizar o modelo e, por fim, dar *feedbacks* para a produção, esse processo é feito de forma contínua.

Nesse sentido, Jove et al. (2019) propõe o desenvolvimento de um sensor virtual para detecção, isolamento e recuperação de dados no processo de uma máquina de mistura bicomponente. O sensor virtual prevê valores dos sensores reais. Caso os valores reais fiquem dentro de um intervalo pré-determinado, eles são válidos, caso contrário, os valores reais são substituídos pelos valores previstos pelo sensor virtual. Além disso, o sensor virtual trabalha para detectar se há falhas no processo.

Cadei et al. (2020) propõe o desenvolvimento de um gêmeo digital mediante utilização de métodos analíticos avançados para detecção de anomalias em séries temporais. O autor utilizar Regressão Ridge para prever valores futuros e comparam com os valores reais para estimar uma distribuição do erro da medida e, a partir disso, definir uma métrica probabilística que decidir se há anomalia ou não. Os dois últimos trabalhados citados, ((Jove et al. (2019)), (Cadei et al. (2020))), mostram de forma clara o contexto de aplicação do presente trabalho em outras áreas e objetos de pesquisa.

A partir dos conceitos mostrados, a metodologia apresentada por esse trabalho é um sensor virtual de eventos indesejados, ou anomalias, em poços de petróleo. Para isso, utiliza-se dados do 3W *dataset* de sensores reais e fabricados por especialistas. Por fim, dentro do contexto de gêmeo digital, o sensor virtual pode ser uma das entidades virtuais que irá compor um gêmeo digital agregado que representa o poço como um todo, isto é, o sensor virtual é uma camada de complexidade que auxiliará a equipe de monitoramento.

1.1.2 Poços de Petróleo

O petróleo pode ser extraído em meios terrestres e em meios marítimos. Em ambos, ela é feita através de poços de petróleo. Conforme é mencionado em (Vargas et al. (2019)), "um poço de petróleo se refere a um conjunto de sensores e de sistemas mecânicos, elétricos e hidráulicos". Os poços terrestres também são chamados de *onshore*, e os poços marítimos denominam-se também *offshore*. Há dois tipos de poços, o injetor e o produtor. Eles dois possuem a principal função de conectar o meio poroso, isto é, o reservatório, a uma instalação industrial, mas o fazem com intuitos diferentes. O poço injetor injeta uma substância no reservatório a fim de aumentar a pressão nele. Em contrapartida, o poço produtor transfere petróleo entre os meios citados (Vargas et al. (2019)). O presente trabalho foca em poços produtores *offshore* devido a natureza dos dados do 3W *dataset*.

O processo produtivo do petróleo engloba quatro etapas para extrai-lo do meio poroso e levá-lo até a plataforma (Andreolli (2016)). A Fig. 1.2 apresenta cada uma das etapas e onde elas se situam na cadeia de produção. A seguir, são apresentados os detalhes de cada estágio.

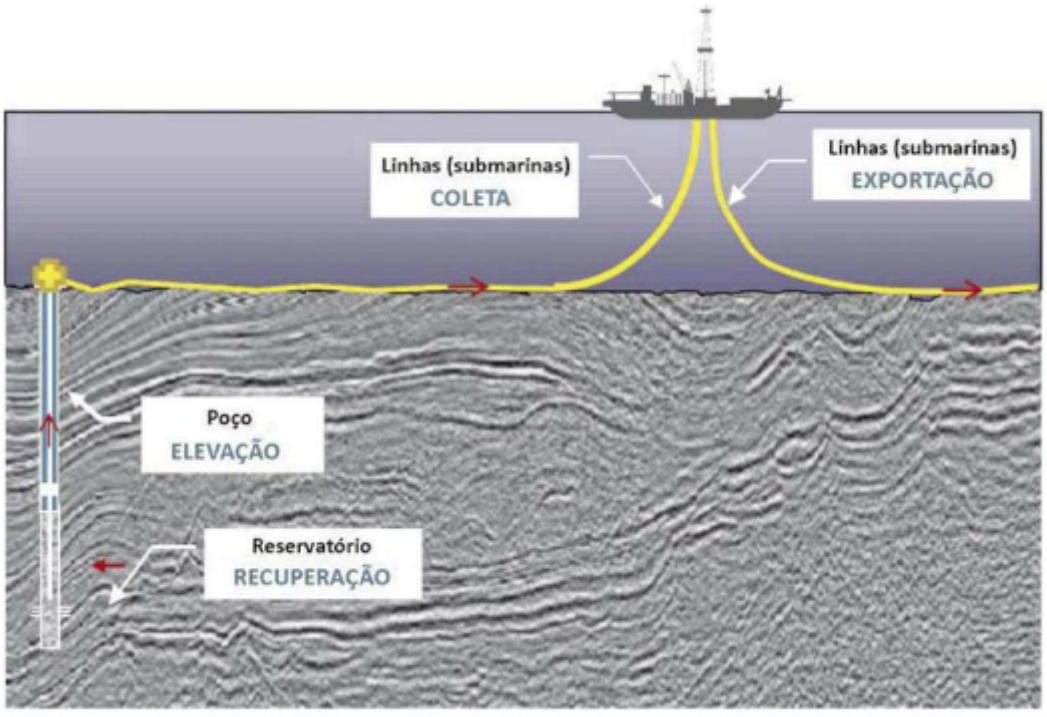

Figura 1.2: Etapas da produção de petróleo desde o meio poroso até a plataforma (Andreolli (2016)).

A produção se inicia na recuperação, na qual ocorre o escoamento no meio poroso, conhecido como reservatório. O escoamento corre através de canais estreitos e densos. Em segunda instância, acontece a elevação, que é caracterizada pelo fluxo vertical do fluido na coluna de produção dentro do poço. Com isso, a força gravitacional é a componente que mais influencia na capacidade de o fluido escoar. Nesse contexto, há dois tipos de elevação, a natural e a artificial, as quais podem ser divididas em subtipos (Andreolli (2016), Vargas et al. (2019)).

Quando a elevação do fluido através da coluna de produção ocorre sem adição de alguma forma de energia, denomina-se a elevação como natural, isto é, o fluido escoa mediante a energia fornecida pela pressão do reservatório. Esse tipo de poço também é chamado de surgente. Por outro lado, a elevação artificial acontece quando a pressão estática não supre a demanda de energia para elevar o fluido. Para complementar essa energia, é utilizado algum meio artificial, por exemplo, injeção de gás natural em alta pressão (elevação artificial com injeção de Gás ou *Gas Lift*), uso de bomba centrífuga (elevação artificial com bombeio centrífugo submerso) e transformação de movimento rotativo em movimento alternativo (elevação artificial com bombeio mecânico). Além disso, vale salientar que um poço pode operar em alguns momentos com elevação natural e em outros com elevação artificial (Andreolli (2016), Vargas et al. (2019)).

Ademais, o próximo elo da cadeia produtiva é a coleta. Ela é descrita como o escoamento do fundo do mar até a plataforma. Por fim, a exportação fecha o processo escoando os fluidos da plataforma até bases terrestres ou até outros navios coletores de óleo e gás. Cada etapa citada possui métodos, riscos, processos e anomalias associados específicos.

Nesse contexto, o 3W *dataset* foca em poços *offshore* de elevação natural de gás e de petróleo, isto é, esse é o objeto desse trabalho. Diante do exposto, ao aliar esses dados com o desenvolvimento de um sensor virtual e de um gêmeo digital para tratar anomalias nos poços, encaixa-se o presente trabalho na área de Acompanhamento de Produção, a qual é uma das três grandes áreas das atividades de elevação e de escoamento e busca, segundo Andreolli (2016), reduzir a ocorrência de problemas na produção. Além disso, o autor cita que, no projeto de tubulações, os aspectos operacionais devem ser contemplados, os quais estão relacionados à garantia do escoamento. Esses aspectos operacionais estão diretamente relacionados a alguns dos eventos indesejados, por exemplo, depósito de inorgânicos, e a previsão, prevenção e mitigação deles é a garantia do escoamento, o que também se relaciona diretamente com o presente trabalho.

1.1.3 Detecção e Prognóstico de Anomalias

Outra parte importante desse projeto é o tratamento das anomalias, as quais são dados com padrões diferentes do usual ou de instâncias normais (Liu, Ting e Zhou (2008)). Para essa tarefa, há três práticas, as quais precisam ser distinguidas: classificação de anomalias, detecção de anomalias e prognóstico de anomalias.

Baseado em (Vargas (2019)), as definições são as seguintes:

- Detecção: identifica que ocorreu alguma anomalia, sem especificá-la, ou seja, há dois rótulos: normalidade e anormalidade. Pode ser tanto *online* como *offline*;
  - Para a detecção de anomalia, são utilizadas múltiplos monitoramentos, que acionam alarmes menos ou mais importantes a depender de quais parâmetros foram violados.
- Classificação: identifica, uma vez ocorrida a anomalia, qual o tipo dela;
  - Na prática, um profissional especialista na área de Elevação e Escoamento de petróleo analisa se algum padrão característico das variáveis consideradas relevantes se formou após a detecção da anomalia.
- Prognóstico: identifica antecipadamente rótulos binários ou algum tipo específico de anomalia.

Dessa forma, o presente trabalho foca os esforços na classificação, na detecção e, principalmente, no prognóstico de anomalias.

1.1.4 Aplicações similares

O trabalho de Vargas (2019), além de construir a base de dados 3W *dataset*, testa algoritmos de classificação para detecção de anomalias na base de dados. O autor utiliza os algoritmos Floresta de Isolamento, *One Class Support Vector Machine* com *kernel* RBF, *One Class Support Vector Machine* com *kernel* poly, *One Class Support Vector Machine* com *kernel* sigmoid e *One Class Support Vector Machine* com *kernel* linear. Os resultados mostraram que o melhor algoritmo foi a Floresta de Isolamento.

Junior (2022) testa algoritmos na abordagem de classificação na base de dados 3W *dataset* para detecção de anomalia, os quais foram Floresta de Isolamento, *One Class Support Vector Machine* com *kernel* RBF, *One Class Support Vector Machine* com *kernel* poly, *One Class Support Vector Machine* com *kernel* sigmoid, *One Class Support Vector Machine* com *kernel*, *Local Outlier Factor*, Envelope Elíptico, Autoencoder e *Long Short-Term Memory*, Memória de Curto Prazo Longa (LSTM). Os resultados apresentaram o *Local Outlier Factor* e o Autoencoder como os algoritmos de melhor performance.

Marins et al. (2021) também utilizam os dados do 3W *dataset*, mas aplica uma abordagem supervisionada com o algoritmo Floresta Aleatória para detectar anomalias. O autor utiliza três abordagens diferentes. Na primeira, ele treina o modelo para detectar se há ou não anomalia. Na segunda, ele treina um modelo para detectar cada tipo de evento específico. Na última, ele treina um único modelo para detectar todos os eventos. O algoritmo produziu bons resultados, por exemplo, na terceira abordagem a acurácia foi de 94%.

Andrianov (2018) desenvolve um medidor virtual de escoamento (*virtual flow metering*) para prever taxas de escoamento multifásico em poços durante a produção de petróleo e de gás. Neste trabalho, o autor mostra que o modelo LSTM alcança uma boa performance tanto para prever os valores no instante de tempo atual e em instantes de tempo futuros.

1.2 Objetivos

O objetivo geral desse trabalho é utilizar o 3W *dataset* para desenvolver um sensor virtual com base em Aprendizado de Máquina para prognóstico de anomalias em poços de petróleo dentro do *framework* de um gêmeo digital. Mediante isso, a metodologia poderá ser empregada como uma ferramenta da engenharia de manutenção em múltiplos domínios, por exemplo, o monitoramento de falhas em estruturas marítmas.

1.3 Metodologia

- Reproduzir os resultados já obtidos com o 3W *dataset* com foco em classificação e detecção dos eventos indesejados;
- Expandir esses resultados para o desenvolvimento de um sensor virtual de detecção de anomalias;
- Utilizar metologias disponíveis na literatura para previsão de séries temporais;
- Aliar a previsão das séries temporais com a detecção de anomalias;
- Propor uma metodologia de prognóstico de anomalias para um sensor virtual dentro de um contexto de gêmeo digital.

1.4 Organização do Trabalho

O trabalho está organizado como segue. O capítulo 1 reúne a introdução, o referencial teórico e os objetivos. O capítulo 2 exibe os detalhes sobre o 3w *dataset*. O capítulo 3 apresenta as principais técnicas de aprendizado de máquina, focando nas que serão utilizadas no trabalho. O capítulo 4 apresenta como o trabalho foi desenvolvido para chegar nos resultados. O capítulo 5 mostra os resultados obtidos. Por fim, o capítulo 6 sumariza o trabalho e propõe possíveis próximos passos para a continuação deste trabalho.

Capítulo 2

Base de dados de um poço de petróleo: 3W *dataset*

O 3W *dataset*, principal insumo deste trabalho, é fruto do trabalho de Vargas et al. (2019). Ele é composto por séries temporais reais, simuladas e desenhadas, capturadas por 8 sensores diferentes. Essas séries podem representar o poço em normal funcionamento ou o poço com algum dos oito eventos indesejáveis acontecendo (Vargas et al. (2019)).

2.1 Estrutura Geral

O *dataset* é composto por m Séries Temporais Multivariadas (STM). Essa definição ajustada ao presente contexto é retirada de (Vargas et al. (2019)). Cada STM é nomeada de instância e é constituída por n séries temporais univariadas, isto é, as variáveis. Dessa forma, cada STM possui dez séries temporais univariadas ordenadas no tempo, oito de variáveis de processo, uma com o rótulo da observação e uma com o *timestamp* de captura do dado. Todas as séries temporais univariadas de uma determinada STM possuem a mesma quantidade de observações extraídas no tempo, mas nem todas as instâncias devem ter o mesmo número de observações.

Além disso, há instâncias de três tipos à depender da fonte: reais, desenhadas à mão e simuladas. As instâncias reais são as que realmente ocorreram durante a produção e não possuem nenhum pré-processamento. Por outro lado, as instâncias simuladas foram obtidas mediante o OLGA, um simulador dinâmico multifásico, utilizado pela Petrobras para simular cenários em poços de petróleo. Já as instâncias desenhadas à mão foram construídas por especialistas da área por intermédio de uma ferramenta composta por um modelo de gráfico e uma metologia de processamento de imagem (Vargas et al. (2019)). A Fig. 2.1 retirada de (Vargas et al. (2019)) mostra um exemplo de gráfico desenhado por um especialista para criar uma instância. É possível ver todos os campos que devem ser preenchidos durante esse processo. Dessa forma, o principal intuito de criar essas STM simuladas ou desenhadas é o

de lidar com um problema bastante presente na realidade industrial: o desbalanceamento do evento de interesse nos dados.

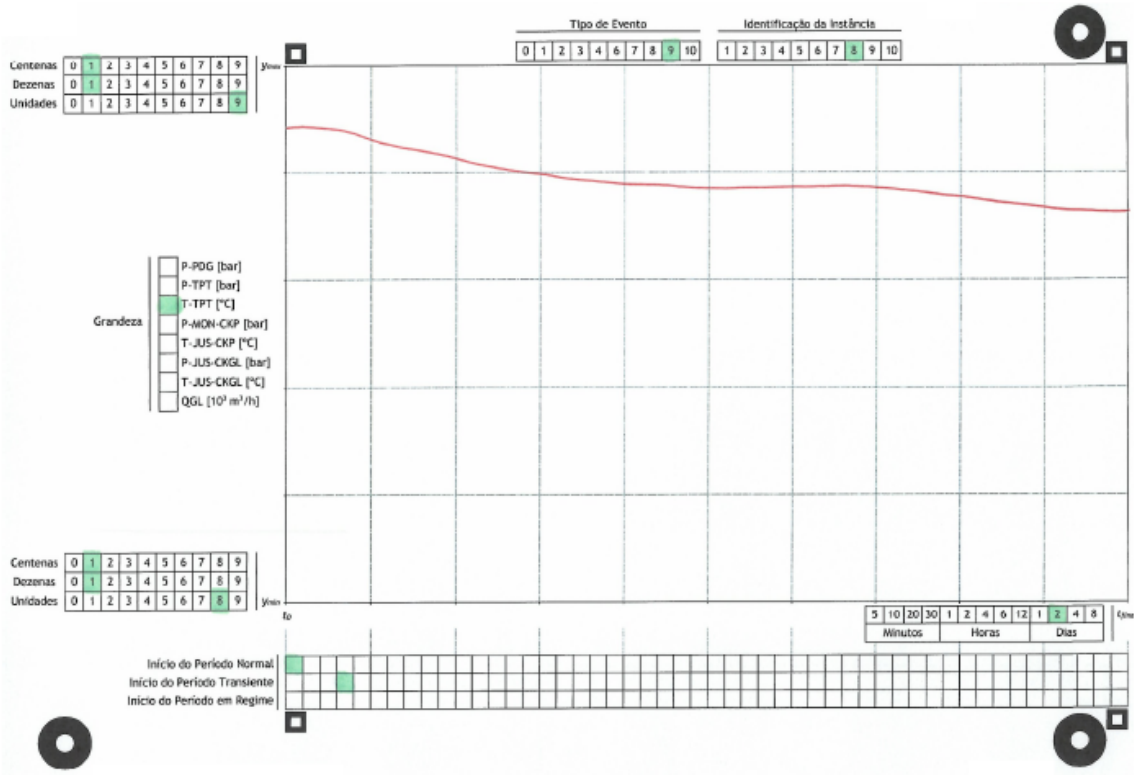

Figura 2.1: Exemplo de gráfico utilizado para criar instâncias desenhadas(Vargas et al. (2019)).

Devido ao fato de os dados das instâncias reais não passarem por pré-processamento, há três problemas em algumas instâncias: variáveis ausentes, variáveis congeladas e observações não rotuladas. As variáveis ausentes dizem respeito às séries temporais univariadas que não possuem nenhum valor em uma determinada instância devido à problemas nos sensores ou na rede de comunicação. A Fig. 2.3 apresenta um exemplo com cinco observações de uma variável ausente, T-JUS-CKGL. Por outro lado, as variáveis congeladas são as séries temporais univariadas que apresentam apenas um valor para todas as observações. Geralmente isso se dá em função de problemas nos sensores, nas configurações de sistema ou na rede de comunicação. Para o presente trabalho, as variáveis congeladas são um problema, uma vez que elas não indicam nenhum comportamento relacionado às anomalias. A Fig. 2.3 exibe uma amostra com cinco observações de duas variáveis congeladas, P-PDG e QGL. Por fim, observações não rotuladas não possuem nenhuma das três classificações: normal, transiente e estável de anomalia, ainda assim, a instância terá um rótulo.

Outro ponto importante de se destacar é a forma como é feita a rotulagem dos dados. Há duas classes de rótulos, o rótulo de instância e o rótulo de observação. O primeiro diz respeito ao evento que será atribuído a cada série temporal multivariada, limitando-se à apenas um rótulo. O segundo refere-se aos três períodos, sequência contínua de observações, em que uma observação se encontra: normal, transiente e estável de anomalia. Nos períodos

normais, hão há anomalias. Nos períodos transientes, as consequências dos eventos indesejados estão ocorrendo. Após isso, se inicia o período estável de anomalia. A partir desse modo que o *dataset* foi construído, o período transiente pode ser considerado como um período pré-anomalia, ou seja, pode ser usado para prever uma falha (Vargas et al. (2019)).

Posto isso, cada rotulagem tem um intuito específico. O objetivo do rótulo da instância é viabilizar classificações *offline*, isto é, classificação e detecção de anomalias. Em contrapartida, a finalidade do rótulo da observação é permitir classificações *online*, ou seja, prognóstico de falhas, principalmente utilizando o período transiente ou pré-anomalia para prever o período estável de anomalia como o evento indesejado em si.

Anteriormente foi citado que há oito variáveis de processo, definidas como de acordo com Vargas (2019):

1. P-PDG: Pressão do fluido em PDG;
    - "Sensor mais próximo do reservatório e que usualmente falha devido ao nível de hostilidade do ambiente. Além disso, como o PDG é um equipamento enroscado na própria coluna de produção, em caso de falha desse sensor, a sua substituição é considerada custosa e arriscada, pois demanda a substituição da própria coluna de produção".
2. P-TPT: Pressão do fluido em TPT;
    - "Sensor interno à Árvore de Natal Molhada que têm confiabilidade considerada boa pelos profissionais da área Elevação e Escoamento de Petróleo".
3. T-TPT: Temperatura do fluido em TPT;
    - "Sensor interno à Árvore de Natal Molhada que têm confiabilidade considerada boa pelos profissionais da área Elevação e Escoamento de Petróleo".
4. P-MON-CKP: Pressão do fluido montante à válvula *choke* de produção;
    - "Sensor também considerado de boa confiabilidade, quando presentes".
5. T-JUS-CKP: Temperatura fluido jusante à válvula *choke* de produção;
    - "Nessa posição podem existir fluidos de vários poços diferentes, mas esse sensor é considerado relevante porque em geral não há sensor de temperatura do fluido montante à Choke de Produção".
6. P-JUS-CKGL: Pressão do fluido jusante à válvula *choke* de *Gas Lift*;
    - Sensor relacionado à elevação artificial.
7. T-JUS-CKGL: Temperatura do fluido jusante à válvula *choke* de *Gas Lift*;

- Sensor relacionado à elevação artificial.

8. QGL: Vazão de *Gas Lift*.

    - Sensor relacionado à elevação artificial.

As cinco primeiras variáveis são capturadas por sensores geralmente disponíveis para poços de elevação natural posicionados nos equipamentos PDG, TPT e CKP. Devido ao poço estar em uma localização de difícil acesso, há maiores custos e impasses para instalar e calibrar esses sensores. Por isso, pode haver erros de medida. A Fig. 2.2 apresenta a posição de cada um dos sensores das cinco primeiras variáveis. Por outro lado, as variável seis, sete e oito se relacionam ao processo de elevação artificial, mas, por não haver histórico de quando o poço era operado de forma artificial ou não, elas também foram adicionadas ao *dataset*, uma vez que um poço de elevação natural também pode ser operado por meios artificiais em alguns momentos. Dessa forma, essas últimas variáveis auxiliam tratar anomalias também para os poços de elevação artificial. Os sensores que medem essas últimas variáveis são posicionados na linha de serviço (Vargas et al. (2019)).

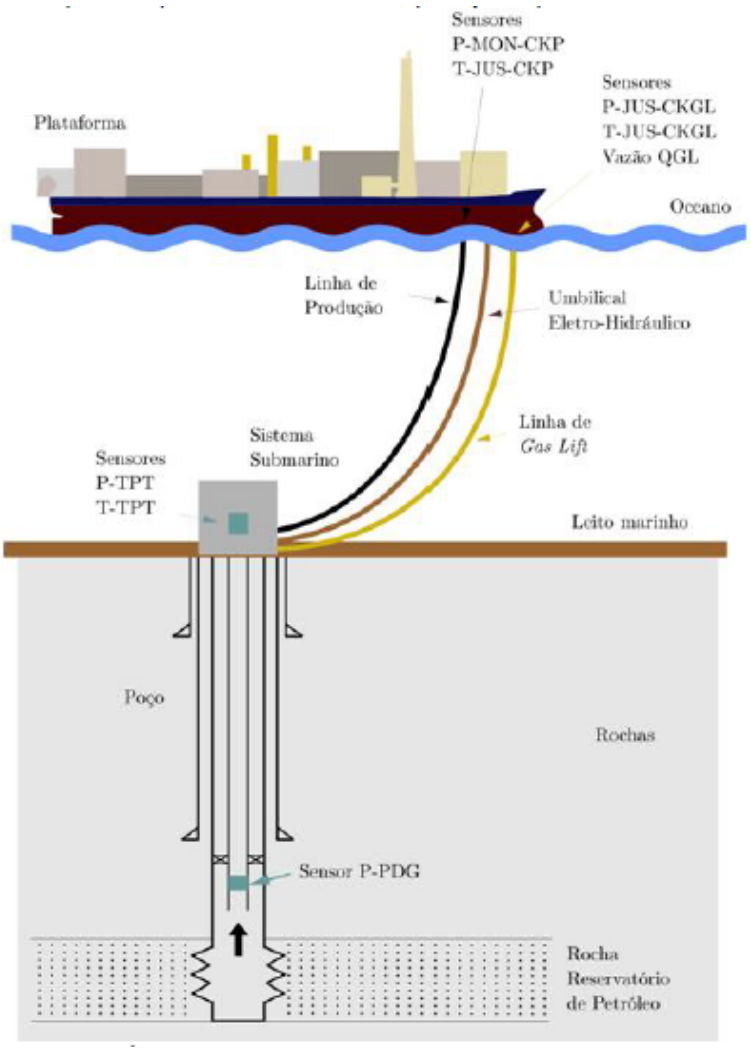

Figura 2.2: Representação de um poço marítimo surgente de petróleo, com a posição dos equipamentos relacionados às variáveis de processo (Junior (2022)).

Além disso, conforme é mencionado em (Vargas et al. (2019)), as unidades usadas foram Pascal [Pa], graus Celsius [ºC] e metros cúbicos por segundo padronizados [sm^3/s], e a taxa de amostragem das observações foi de 1 Hz. A Fig. 2.3 apresenta cinco observações de cada uma das dez séries temporais univariadas que foram uma STM classificada como operação normal. A partir dessa imagem, é possível ver que, a nível de observação, o rótulo também é de operação normal. A Fig. 2.4 apresenta os gráficos de cada uma das oito variáveis de processo associadas a mesma STM da Fig. 2.3, mas os gráficos possuem todas as observações, as quais são todas classificadas como normal.

timestamp	P-PDG	P-TPT	T-TPT	P-MON-CKP	T-JUS-CKP	P-JUS-CKGL	T-JUS-CKGL	QGL	class
2017-02-01 02:02:07.000000	0,0E+00	1,01E+13	1,19E+08	1,61E+12	8,46E+07	1,56E+12		0,0E+00	0
2017-02-01 02:02:08.000000	0,0E+00	1,01E+13	1,19E+08	1,62E+12	8,46E+07	1,56E+12		0,0E+00	0
2017-02-01 02:02:09.000000	0,0E+00	1,01E+13	1,19E+08	1,63E+12	8,46E+07	1,56E+12		0,0E+00	0
2017-02-01 02:02:10.000000	0,0E+00	1,01E+13	1,19E+08	1,64E+12	8,46E+07	1,56E+12		0,0E+00	0
2017-02-01 02:02:11.000000	0,0E+00	1,01E+13	1,19E+08	1,64E+12	8,46E+07	1,56E+12		0,0E+00	0

Figura 2.3: Exemplo de cinco observações das séries temporais univariadas de uma STM classificada como operação normal.

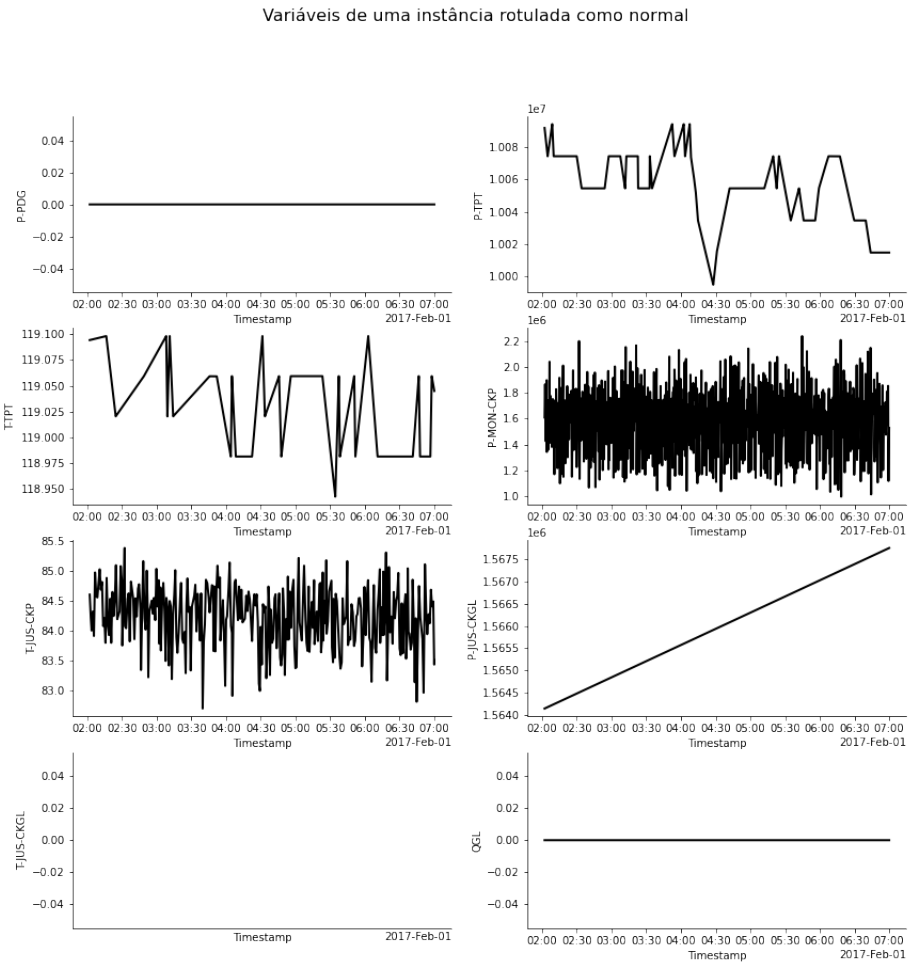

Figura 2.4: Cada uma das oito variáveis de processo de uma instância rotulada como normal.

2.2 Tipos de Anomalias

O 3W *dataset* cobre oito tipos de eventos indesejados que fazem parte do grupo de anomalias mais críticas e que ocorrem durante as três primeiras etapas do processo produtivo: recuperação, elevação e coleta. A seguir são enumeradas e explicadas as oito anomalias:

1. Aumento Abrupto de *Basic Sediment and Water*;
2. Fechamento espúrio da *Downhole Safety Valve*;
3. Intermitência Severa;
4. Instabilidade no Fluxo;
5. Perda rápida de produtividade;
6. Restrição Rápida em CKP;
7. Incrustação em CKP;
8. Hidrato em linha de produção;

É importante mencionar que os números associados a cada anomalia serão utilizados como rótulos a nível de instância e serão citados ao longo do trabalho. Além disso, a Fig. 2.5 apresenta estimativas de janelas temporais que os profissionais que realizam o monitoramento de poços na Petrobras utilizam para confirmar a ocorrência real de uma anomalia. Essas janelas podem ser usadas para otimizar o desempenho dos algoritmos (Vargas (2019)).

TIPO DE ANOMALIA	TAMANHO DE JANELA
1 – Aumento Abrupto de BSW	12 h
2 – Fechamento Espúrio de DHSV	5 min – 20 min
3 – Intermitência Severa	5 h
4 – Instabilidade de Fluxo	15 min
5 – Perda Rápida de Produtividade	12 h
6 – Restrição Rápida em CKP	15 min
7 – Incrustação em CKP	72 h
8 – Hidrato em Linha de Produção	30 min – 5 h

Figura 2.5: Tamanhos das janelas temporais para confirmar ocorrências de anomalias (Vargas (2019)).

2.2.1 Aumento Abrupto de *Basic Sediment and Water*

Segundo Andreolli (2016), "o *Basic Sediment and Water* ou BSW é definido como a razão entre a vazão ou volume de água e sedimentos produzidos e a vazão ou volume de líquido produzido, ambos medidos na condição padrão.". Ainda nesse sentido, a quantidade

de sedimentos produzidos é bastante reduzida. Dessa forma, utiliza-se, na prática, o BSW como a fração de água produzida do total de líquidos. Por exemplo, um BSW de 50% quer dizer, na prática, que 50 % da fração volumétrica líquida produzida por esse poço é água.

O aumento de BSW durante a ciclo de vida do poço é esperado devido à maior produção de água, mas não um aumento brusco. Nesse cenário, Andreolli (2016) menciona que o aumento abrupto de BSW provoca problemas relacionadas a diversos processos, por exemplo, à elevação de petróleo, à produção menor de óleo e à garantia de escoamento. Em geral, com esse fenômeno, a temperatura dos equipamentos aumenta, e a pressão, em áreas mais profundas, aumenta e, em pontos mais perto da superfície, diminui (Vargas (2019)). A Fig.2.6 apresenta um gráfico para cada variável de processo para um exemplo de instância que passa pelo aumento abrupto de *Basic Sediment and Water*.

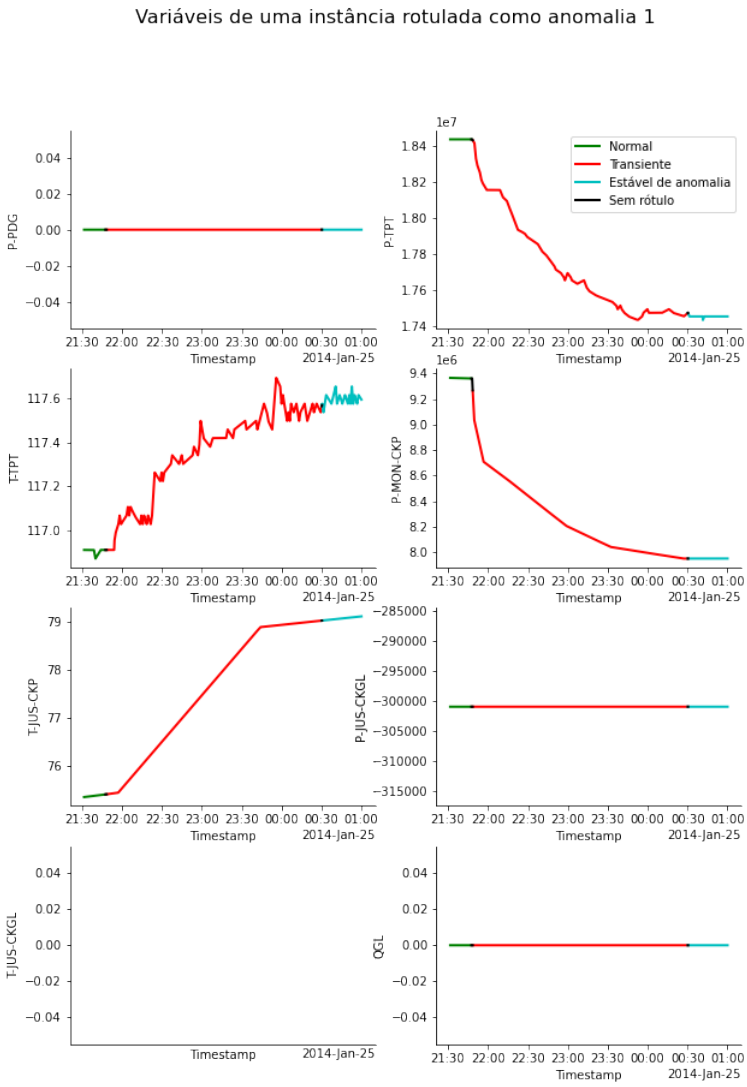

Figura 2.6: Variáveis de processo para uma instância rotulada com o evento aumento Abrupto de *Basic Sediment and Water*.

2.2.2 Fechamento Espúrio da *Downhole Safety Valve*

Downhole Safety Valve é uma válvula de segurança, que permanece aberta por um atuador hidráulico e fecha em caso de desconexão entre o poço e a unidade de produção. A anomalia aqui tratada diz respeito ao fechamento de forma errônea da válvula. Dessa forma, prever esse evento permite uma atuação mais ágil para reabrir a válvula, evitando perdas (Vargas (2019)). A Fig. 2.7 apresenta um gráfico para cada variável de processo para um exemplo de instância que passa pelo fechamento espúrio da *Downhole Safety Valve* (DHSV).

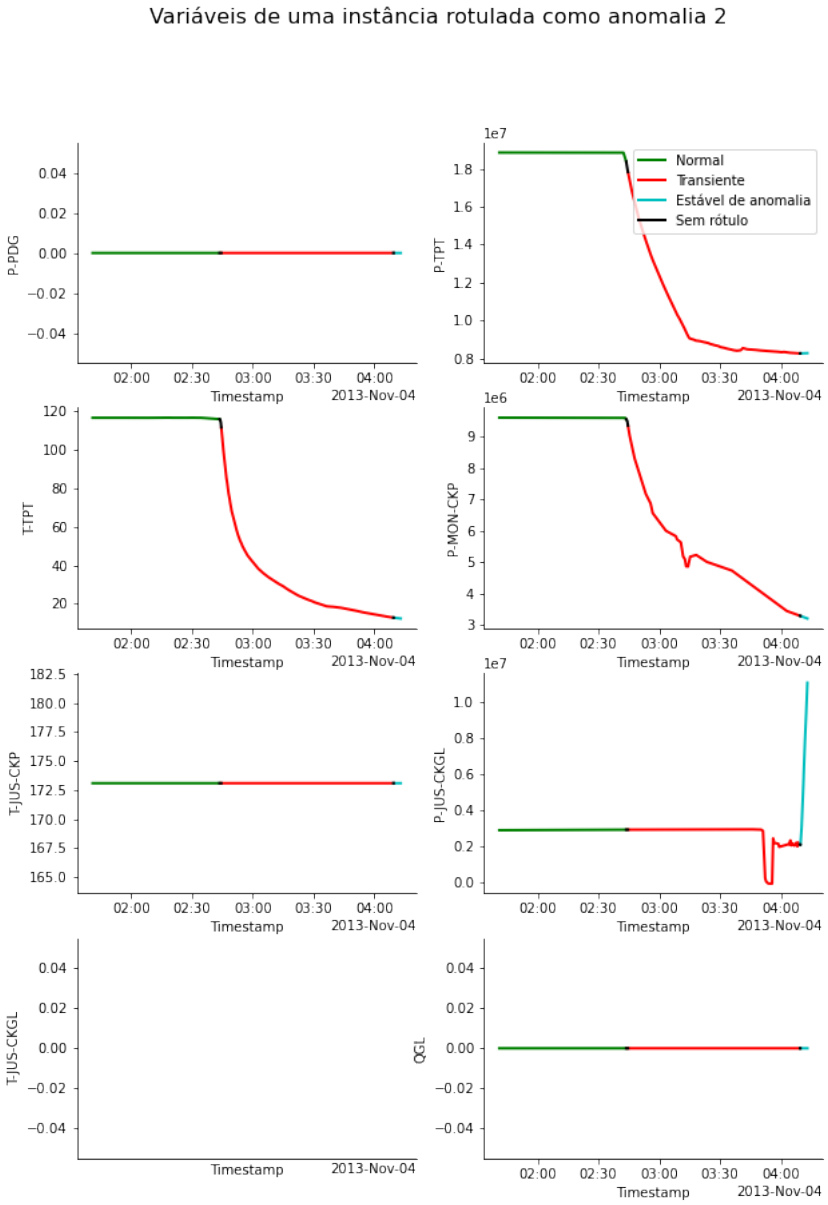

Figura 2.7: Variáveis de processo para uma instância rotulada com o evento fechamento espúrio da *Downhole Safety Valve*.

2.2.3 Intermitência Severa

Esse evento é um regime de fluxo intermitente bifásico ou trifásico com uma distribuição não homogênea das fases gasosa e líquida, isto é, bolhas de gás fluem através da tubulação separadas por poções de líquido. Isso faz com que, na saída, baixa produção com altos picos periódicos de produção, o que é prejudicial para o processo de separação (Meglio et al. (2012)).

Além disso, esse fenômeno carateriza-se pela periodicidade bem definida e pela alta intensidade, possibilitando a captura por sensores ao longo de todo a instalação (Vargas (2019)). Essa anomalia causa queda na produção e pode acarretar em danos aos equipamentos. A Fig. 2.8 apresenta um gráfico para cada variável de processo para um exemplo de instância que passa pela intermitência severa. Vale ressaltar que há apenas períodos estáveis de anomalia nessa instância específica.

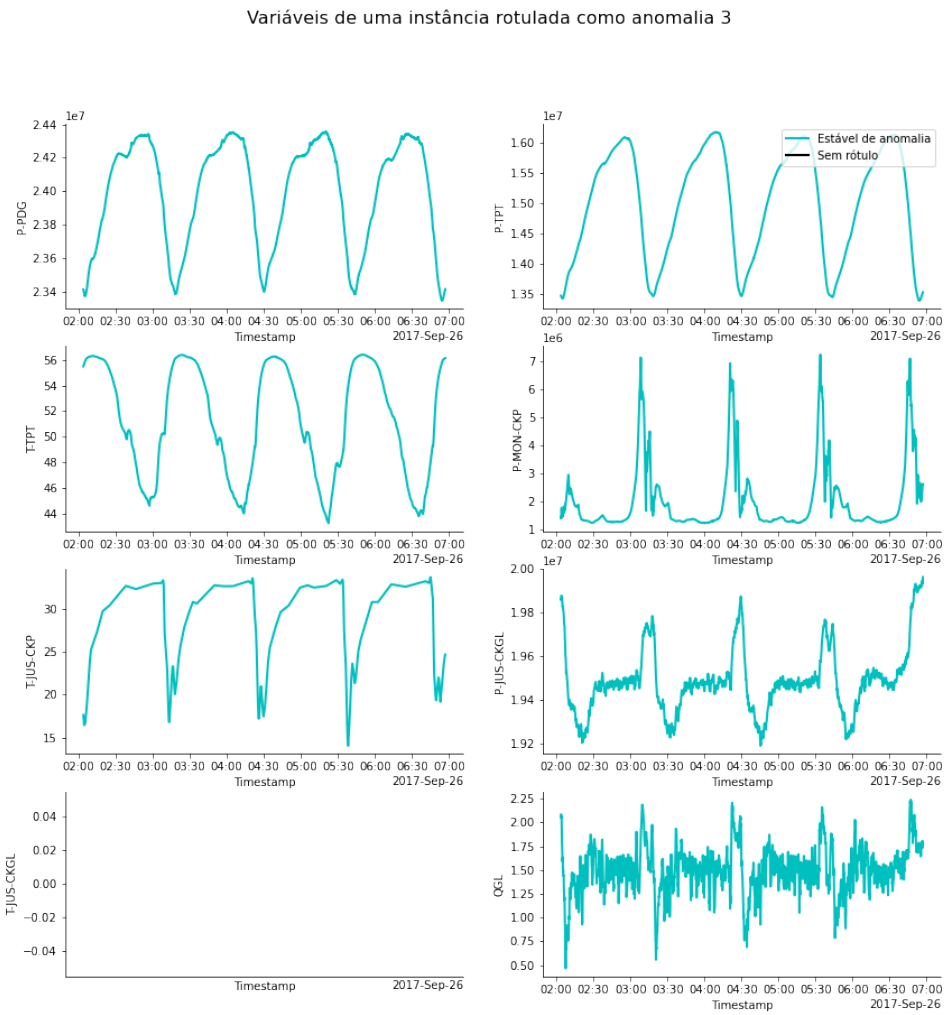

Figura 2.8: Variáveis de processo para uma instância rotulada com o evento intermitência Severa.

2.2.4 Instabilidade no Fluxo

Na instabilidade de fluxo, algumas variáveis apresentam modificações de amplitude relevantes. Esse fenômeno se assemelha a intermitência severa, mas difere-se pela falta de periodicidade entre os picos. Com isso, ele pode anteceder o evento da intermitência severa. Portanto, prever essa anomalia evita diversos problemas futuros (Vargas (2019)). A Fig.2.9 apresenta um gráfico para cada variável de processo para um exemplo de instância que passa pela intermitência severa. Nessa instância, também há apenas períodos estáveis de anomalia nessa instância específica.

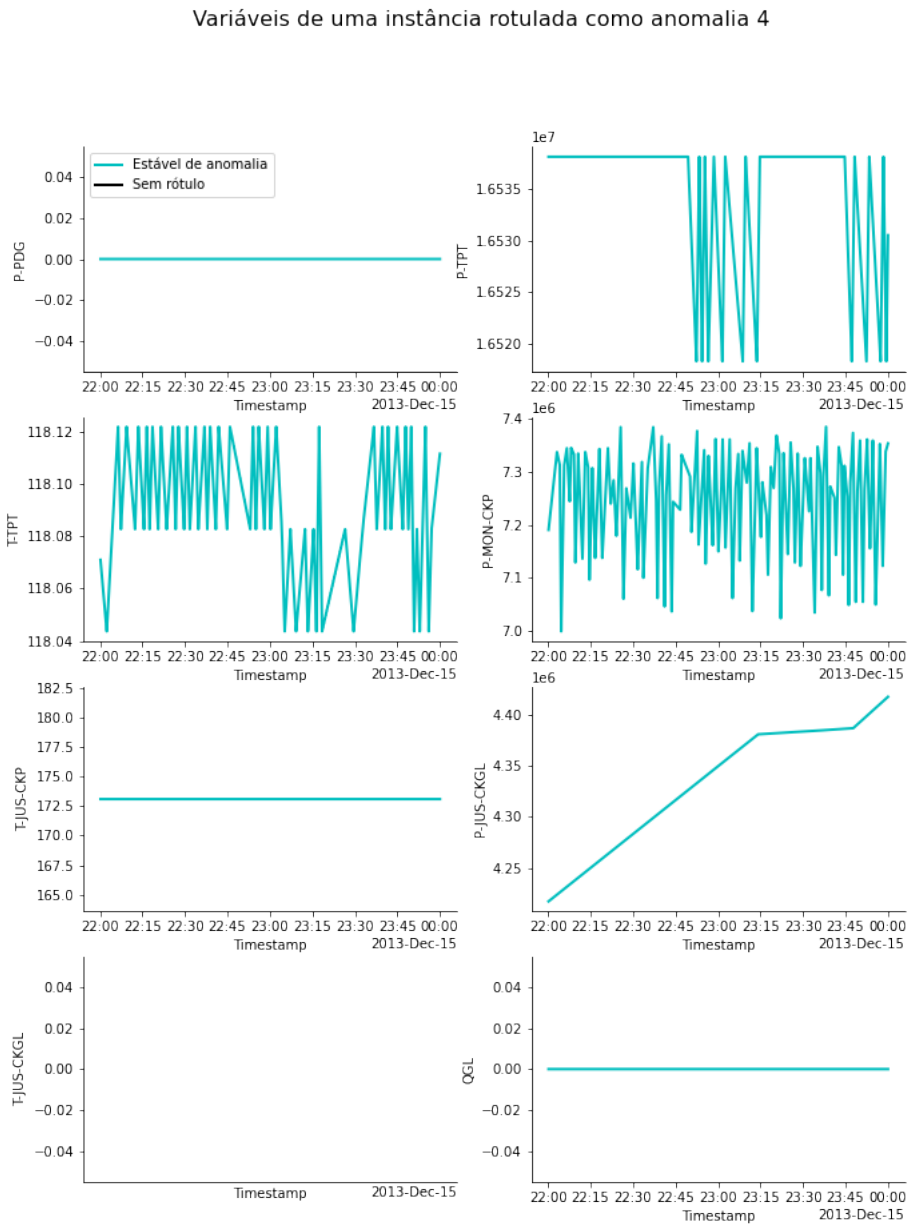

Figura 2.9: Variáveis de processo para uma instância rotulada com o evento instabilidade no Fluxo.

2.2.5 Perda Rápida de Produtividade

A perda rápida de produtividade ocorre quando as propriedades e condições necessárias para produção, por exemplo, a pressão estática e a viscosidade do fluido, mudam ao ponto de a energia do sistema não ser mais suficientes para as forças contrárias à elevação do fluido, isto é, ele não chega à superfície. O caso mais extremo desse fenômeno pode chegar ao fim da surgência. Dessa forma, prever esse fenômeno permite a alteração da operação para que o poço não perca produtividade (Vargas (2019)). A Fig. 2.10 apresenta um gráfico para cada variável de processo para um exemplo de instância que passa pela perda rápida de produtividade.

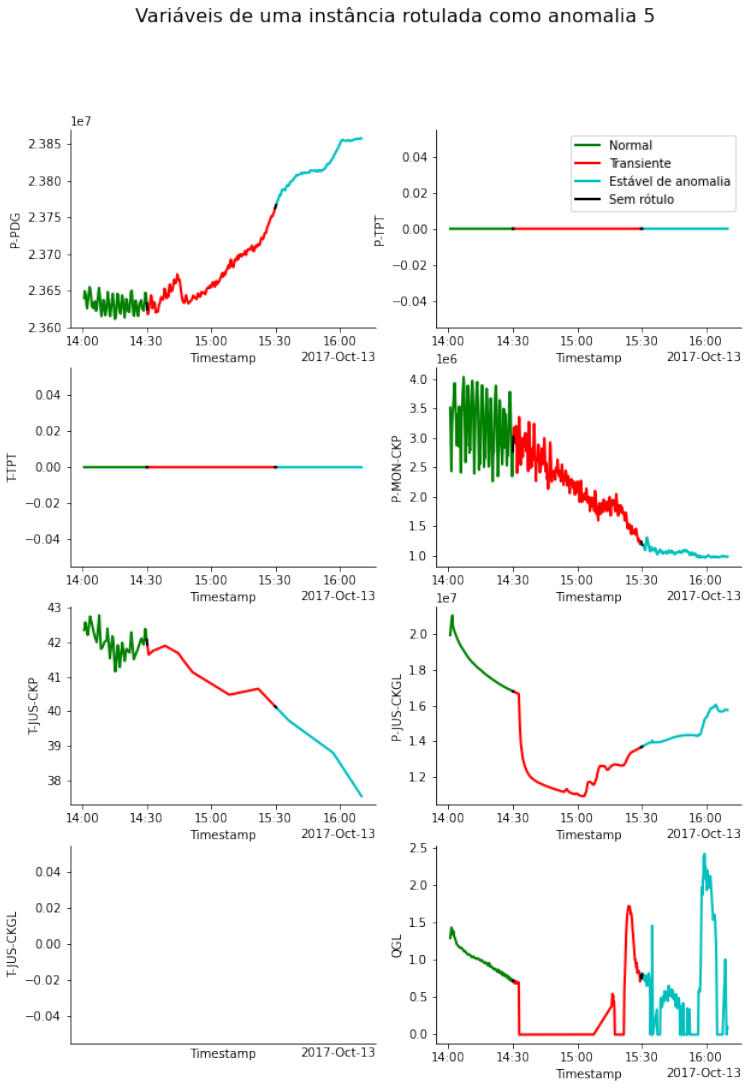

Figura 2.10: Variáveis de processo para uma instância rotulada com o evento perda rápida de produtividade.

2.2.6 Restrição Rápida em CKP

CKP é uma válvula geralmente manual instalada no início da unidade de produção, cuja função é controlar o poço na superfície. Ainda não há uma definição clara para essa anomalia na literatura mas, para a Petrobras, trata-se de uma restrição com amplitude acima de uma referência e durante um período restrito, a qual pode ocorrer devido à problemas operacionais (Vargas (2019)). A Fig. 2.11 apresenta um gráfico para cada variável de processo para um exemplo de instância que passa pela restrição Rápida em CKP.

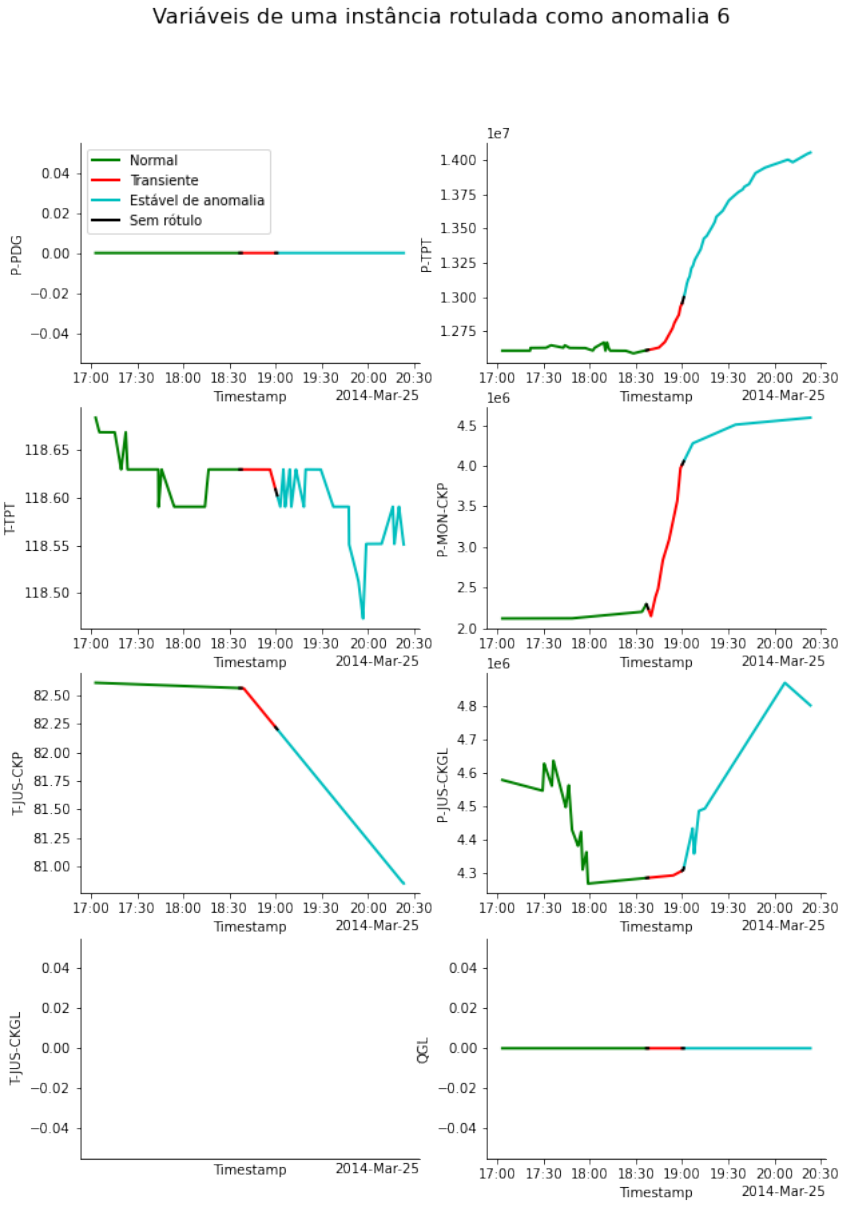

Figura 2.11: Variáveis de processo para uma instância rotulada com o evento restrição Rápida em CKP.

2.2.7 Incrustação em CKP

Ainda com relação à válvula CKP, pode haver depósitos inorgânicos que a obstruem, acarretando no fenômeno da incrustação em CKP, por conseguinte,na redução da produção. Há ações que podem ser tomadas caso esse evento seja previsto (Vargas (2019)). A Fig. 2.12 apresenta um gráfico para cada variável de processo para um exemplo de instância que passa pela restrição Rápida em CKP.

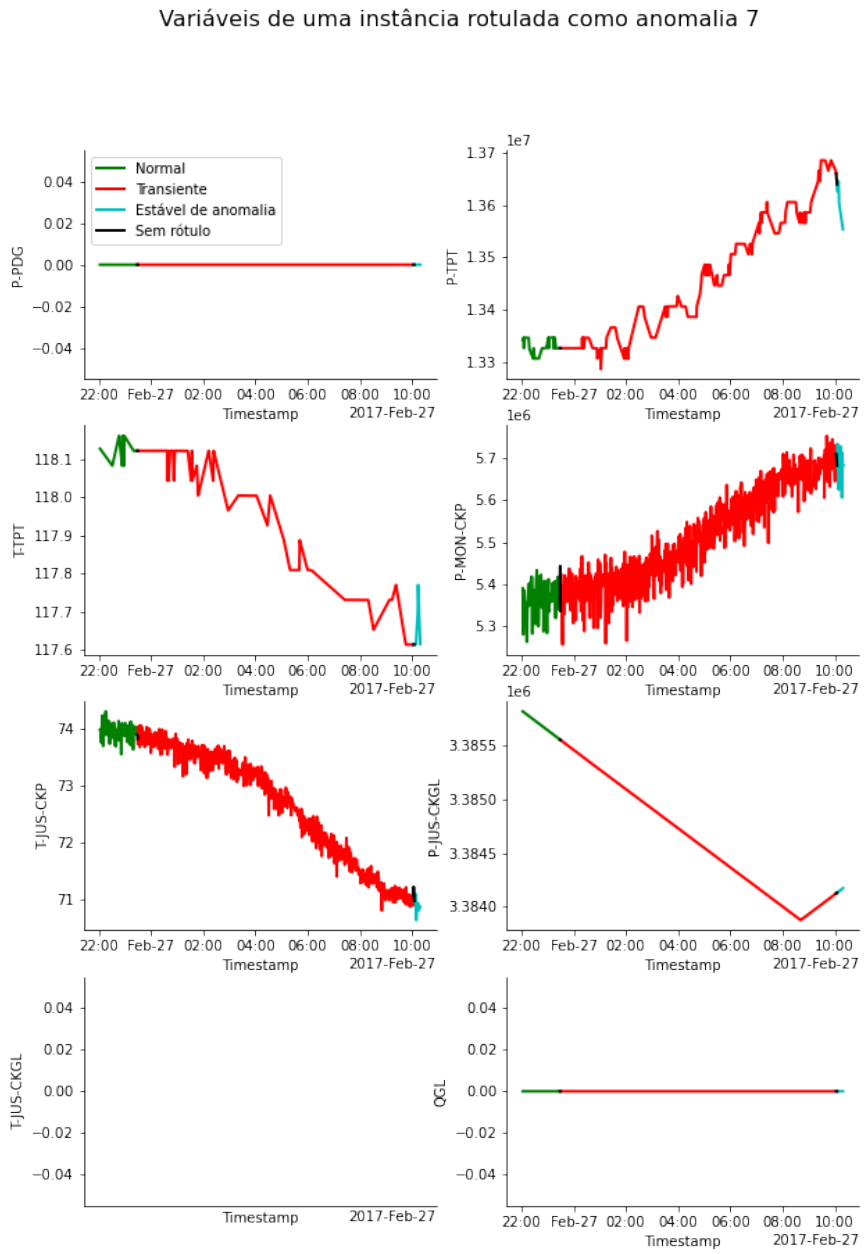

Figura 2.12: Variáveis de processo para uma instância rotulada com o evento incrustação em CKP.

2.2.8 Hidrato em Linha de Produção

O hidrato é um dos grandes problemas da indústria de petróleo. São necessários água e gás para que exista a possibilidade de formação de hidratos, por isso, os oleodutos que escoam óleo morto não possuem esse risco. Por outro lado, em gasodutos há essa possibilidade. A combinação de água, de gás natural, de baixas temperaturas e de altas pressões viabiliza a formação de compostos cristais, os hidratos, os quais acarretam na obstrução do escoamento, isto é, perda de produção (Andreolli (2016)). Além dos gasodutos e dos poços produtores de gás, essa anomalia também ocorre em poços produtores de petróleo, e evita-lá se traduz na prevenção de perdas de produção de dias ou até semanas (Vargas (2019)). Ademais, a Fig. 2.13 um exemplo de formação de hidrato em um poço da Petrobras, na qual a tubulação foi totalmente obstruída (Andreolli (2016)). A Fig. 2.14 apresenta um gráfico para cada variável de processo para um exemplo de instância que passa pelo hidrato em linha de produção.

Figura 2.13: Hidrato em um poço da plataforma P-34 da Petrobras (banco de imagem da Petrobras) (Andreolli (2016)).

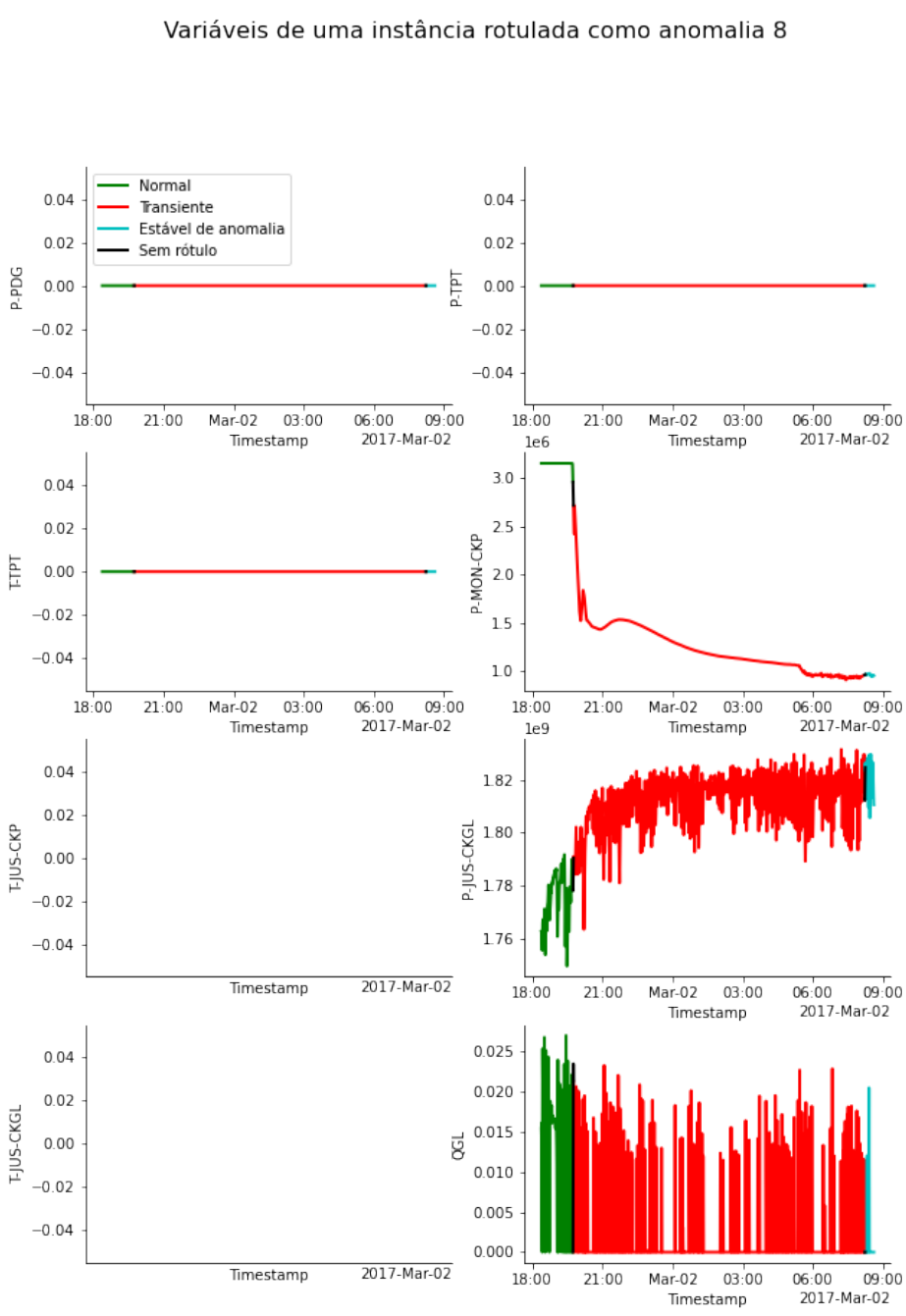

Figura 2.14: Variáveis de processo para uma instância rotulada com o evento hidrato em linha de produção.

2.3 Quantitativos dos Dados

Diante do exposto, é importante destacar alguns quantitativos dos dados presentes no 3W *dataset*, os quais são apresentados a seguir nas tabelas 2.1 e 2.2, e nas Figs. 2.15, 2.16 e 2.17.

A tabela 2.1 apresenta a quantidade de instâncias reais, desenhadas e simuladas de cada tipo de evento. Já a Fig. 2.15 apresenta a razão em porcentagem entre a quantidade de observações sem valor e o total de observações de cada evento. A última coluna dessa tabela apresenta a quantidade total de observações por evento. Além disso, a Fig. 2.16 exibe a razão em porcentagem entre a quantidade de instâncias com a determinada variável congelada em relação ao total de instâncias de cada evento. Por exemplo, o elemento $(1, 1)$ diz que 82,57 % das instâncias possuem a variável P-PDG congelada. Vale lembra que se, para uma determinada STM, todas as observações da variável são nulas, a variável de processo também é considerada congelada. A partir dessas tabelas citadas e das Figs. 2.6, 2.7, 2.8, 2.9, 2.10, 2.11, 2.12 e 2.14, é possível perceber que a variável T-JUS-CKGL apresenta valores ausentes para todas as observações de todas as instâncias, o que possivelmente não contribui em nada com o desenvolvimento do trabalho.

Ainda nesse contexto, a tabela 2.2 expõe, para cada tipo de anomalia, a porcentagem de observações de cada um dos três tipos em relação ao total de observações. Constata-se que algumas anomalias possuem todas as observações concentradas em apenas um tipo de período. Ademais, a Fig. 2.17 mostra a distribuição dos eventos reais do 3W *dataset* no tempo.

Evento	# Instâncias reais	# Instâncias desenhadas	# Instâncias simuladas	Total
0	597	0	0	597
1	5	10	114	129
2	22	0	16	38
3	32	0	74	106
4	344	0	0	344
5	12	0	439	451
6	6	0	215	221
7	4	10	0	14
8	3	0	81	84
Total	1025	20	939	1934

Tabela 2.1: Quantitativo de instâncias de cada tipo de evento e de cada fonte.

Evento	P-PDG	P-TPT	T-TPT	P-MON-CKP	T-JUS-CKP	P-JUS-CKGL	T-JUS-CKGL	QGL	Total
0	0,04%	0,04%	0,04%	10,13%	14,79%	25,63%	100,00%	25,48%	9956791
1	0,00%	0,00%	0,00%	0,00%	0,00%	98,68%	100,00%	98,68%	8988607
2	0,10%	0,10%	0,10%	17,89%	22,20%	97,44%	100,00%	96,58%	619464
3	0,01%	0,01%	0,01%	0,01%	0,01%	88,24%	100,00%	88,24%	4834079
4	0,03%	0,04%	0,04%	0,03%	0,05%	50,51%	100,00%	23,54%	2462076
5	0,00%	0,00%	0,00%	0,00%	0,00%	97,26%	100,00%	97,26%	13224267
6	0,00%	0,00%	99,07%	0,00%	0,00%	99,18%	100,00%	99,18%	5859002
7	0,01%	0,01%	0,01%	0,01%	0,01%	89,92%	100,00%	89,92%	2690918
8	0,00%	0,00%	0,00%	0,00%	4,00%	96,00%	100,00%	96,00%	2278011

Figura 2.15: Porcentagem de observações com valor ausente em relação ao total de observações de cada tipo de evento.

Evento	P-PDG	P-TPT	T-TPT	P-MON-CKP	T-JUS-CKP	P-JUS-CKGL	T-JUS-CKGL	QGL	Total
0	82,6%	2,2%	0,8%	9,5%	15,7%	36,7%	100,0%	99,0%	597
1	3,1%	0,0%	0,0%	0,0%	0,8%	96,9%	100,0%	100,0%	129
2	13,2%	0,0%	0,0%	36,8%	52,6%	94,7%	100,0%	97,4%	38
3	0,9%	0,0%	0,0%	0,0%	0,0%	69,8%	100,0%	70,8%	106
4	64,2%	0,9%	2,0%	0,0%	32,8%	50,6%	100,0%	69,8%	344
5	1,3%	0,4%	0,2%	0,0%	0,0%	97,3%	100,0%	98,7%	451
6	2,7%	0,0%	97,3%	0,0%	0,0%	98,6%	100,0%	100,0%	221
7	21,4%	0,0%	0,0%	0,0%	0,0%	71,4%	100,0%	92,9%	14
8	1,2%	1,2%	1,2%	0,0%	3,6%	96,4%	100,0%	96,4%	84

Figura 2.16: Porcentagem de instâncias com uma determinada variável congelada do total de instâncias.

Evento	% Normal	% Estável de anomalia	% Transiente	% Sem rótulo	# Observações
0	100,0	0,0	0,0	0,0	9956791
1	8,94	32,33	58,72	0,01	8988607
2	17,8	58,47	23,57	0,17	619464
3	0,0	100,0	0,0	0,0	4834079
4	0,0	100,0	0,0	0,0	2462076
5	1,9	79,78	18,31	0,01	13224267
6	7,19	66,2	26,6	0,01	5859002
7	10,41	4,04	85,53	0,02	2690918
8	6,78	26,6	66,6	0,02	2278011

Tabela 2.2: Porcentagem de observações de período em relação ao total de observações de cada tipo de anomalia.

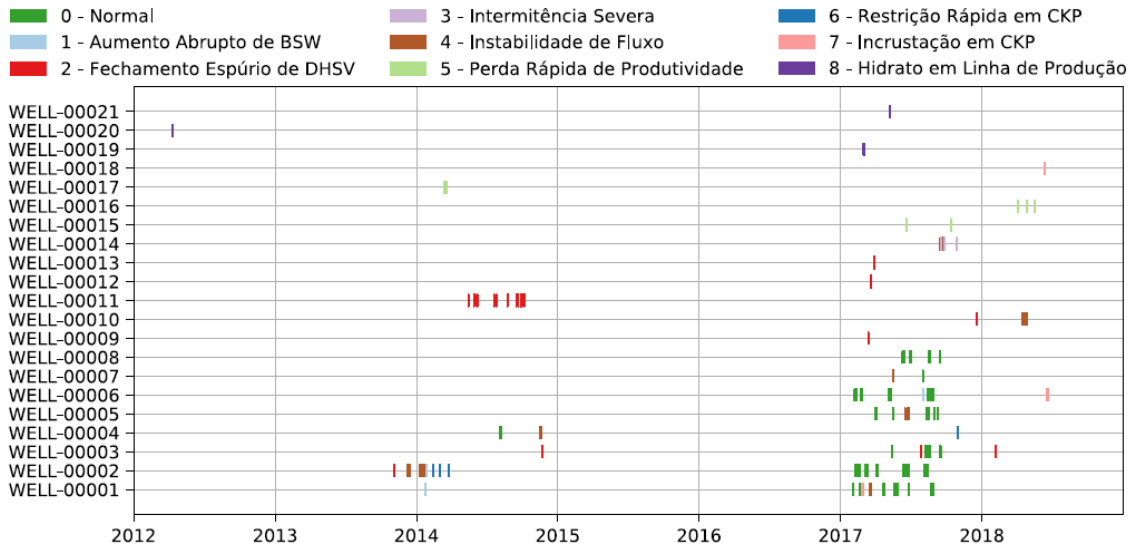

Figura 2.17: Mapa da dispersão das instâncias reais históricas do 3W *dataset* (Vargas et al. (2019)).

Com relação ao 3W *dataset*, Vargas (2019) apresenta um *benchmark* para testar o impacto de usar instâncias desenhadas e simuladas. Nesse sentido, foram usados algoritmos de aprendizado de máquina e testes estatísticos em sete cenários diferentes para o conjunto de treinamento, os quais são:

1. Apenas instâncias reais;
2. Apenas instâncias simuladas;
3. Apenas instâncias desenhadas à mão;
4. Apenas instâncias reais e simuladas;
5. Apenas instâncias reais e desenhadas à mão;
6. Apenas instâncias simuladas e desenhadas à mão;
7. Apenas instâncias reais, simuladas e desenhadas à mão.

A validação e os resultados foram colhidos com base em instâncias reais. Os resultados obtidos mostraram que apenas os cenários quatro e cinco foram capazes de mostrar, com grande probabilidade, melhores performances quando comparados ao cenário um, isto é, combinar instâncias reais com instâncias simuladas ou instâncias reais com instâncias desenhadas tem um impacto positivo na detecção de anomalias em instâncias reais.

2.4 Organização dos Dados

Por fim, é importante exibir a organização dos arquivos. Há nove pastas com os rótulos a nível de instância começando por 0, instâncias normais, depois 1, instâncias do evento 1 e assim por diante. Dentro de cada pasta, o arquivo é nomeado com o tipo de instância, real, simulada ou desenhada, com um código numérico e com a data relacionados ao poço e à aquisição dos dados quando a instância é real, e com um código numérico sequencial quando a instância é desenhada ou simulada. Além disso, o rótulo a nível de observação é feita da seguinte forma:

- Rótulo "0": período normal;
- Rotulo "10x": período transiente, sendo x um número de 1 à 8 de acordo com o rótulo da instância;
- Rótulo "x": período estável de anomalia, sendo x um número de 1 à 8 de acordo com o rótulo da instância.

Capítulo 3

Aprendizado de Máquina

O Aprendizado de Máquina, assim como a ciência de dados e a estatística, é uma área que descreve como aprender e como fazer previsões sobre os dados. Nesse sentido, com o avanço da tecnologia e com a revolução do *big data*, mais dados estão disponíveis e há mais capacidade computacional, esses dois fenômenos combinados resultou em um cenário próspero para o uso do aprendizado de máquinas. Dessa forma, hoje diversas empresas o utilizam em diversas áreas desde biotecnologia até a engenharia de carros autônomos (Mehta et al. (2019)).

As técnicas de Aprendizado de Máquina geralmente focam mais em previsão do que em estimação. Além disso, elas são utilizadas em problemas de alta dimensão mais complexos do que os encontrados na estatística clássica. Um problema típico de Aprendizado de Máquina segue a seguinte estrutura: escolhe-se uma variável x do sistema estudado, que se relaciona com alguns parâmetros θ de um modelo $p\left(x|\theta\right)$, o qual descreve a probabilidade de observar x dado θ. Dessa forma, reúne-se os dados necessários em um *dataset* X para estimar o modelo, isto é, encontrar $\hat{\theta}$ que fornecem a melhor explicação para os dados. Ao se deparar com dados novos, os parâmetros estimados são utilizados para prever novos valores (Mehta et al. (2019)). Dessa forma, o objetivo principal é estimar um modelo que irá generalizar para novas instâncias, isto é, aproximar o máximo a generalização do valor real (Géron (2019)). Por exemplo, neste trabalho, a variável de observação é a ocorrência da anomalia, e o *dataset* X é composto pelas variáveis de processo. Assim, ajusta-se um modelo $p\left(x|\theta\right)$ às variáveis. Com esse resultado, é possível classificar, detectar e prever anomalias. Dessa forma, utiliza-se o modelo estimado em dados não vistos anteriormente, chamados de dados de teste, então, pode-se comparar o *output* do modelo com o dado real a fim de avaliar a acurácia e outras métricas, isto é, quão próximo do real, as previsões chegam. Nesse sentido, quanto mais próximo os dados de treinamento estiverem da situação real em que o modelo será aplicado, melhor ele generalizará.

Há algumas categorias em que se classificam os sistemas de aprendizado de máquinas (Géron (2019)), a saber:

- Se o treinamento é feito com ou sem supervisão humana:
  - Aprendizado supervisionado;
  - Aprendizado não supervisionado;
  - Aprendizado semi-supervisionado;
  - Aprendizado por reforço.
- Se o aprendizado é ou não incremental:
  - Aprendizado *online*;
  - Aprendizado em *batch*.

As categorias acima não são as únicas, mas estão entre as principais para o presente trabalho. Em seguida, elas serão explicadas em maiores detalhes.

3.1 Aprendizado Supervisionado e Não Supervisionado

No aprendizado supervisionado, os dados de treino, os quais são usados para ajustar o algoritmo, são rotulados. Por exemplo, um conjunto de treinamento para classificar um determinado tipo de anomalia deve ter um rótulo que indica a qual evento indesejado os dados se referem. Dessa forma, o algoritmo aprende que a classe do rótulo segue determinado padrão. Ao generalizar para novos dados, o sistema terá que encontrar um rótulo igual aos do conjunto de treinamento. A Fig. 3.1 apresenta um exemplo típico de um conjunto de treinamento rotulado para a classificação supervisionada de *e-mails* como *spam* ou não *spam*, a instância nova mostrada é um *e-mail* que deve ser classificado dentro dos rótulos apresentados (Géron (2019)).

Figura 3.1: Exemplo de um conjunto de treinamento de aprendizado supervisionado para classificar um *e-mail* como *spam* (Géron (2019)).

As tarefas mais comuns de aprendizado supervisionado são classificação e regressão. A classificação é a previsão de classes, por exemplo, fraudador e não fraudador, *smap* e não

spam, classificar um tipo de flor, entre outros. Por outro lado, a regressão é a previsão de valores, isto é, o alvo é um valor numérico, por exemplo, prever preço de ações, preço de imóveis, entre outros. Os principais algoritmos de aprendizado supervisionado são (Géron (2019)):

- *K-Nearest Neighbours* (KNN);
- Regressão linear;
- Regressão Logística;
- Máquinas de Vetores de Suporte (do inglês, SVM - *Support Vector Machines*);
- Árvores de decisão;
- Florestas Aleatórias;
- Redes neurais.

No aprendizado não supervisionado, os dados de treinamento não têm rótulos, isto é, o sistema tenta aprender padrões dos dados sem um professor (Géron (2019)). A Fig. 3.2 apresenta o resultado de um algoritmo de agrupamento, aprendizado não supervisionado. O algoritmo em questão separou as instâncias em quatro grupos em função das características um e dois. Além disso, os algoritmos de detecção de anomalias aplicados nesse trabalho são não supervisionados.

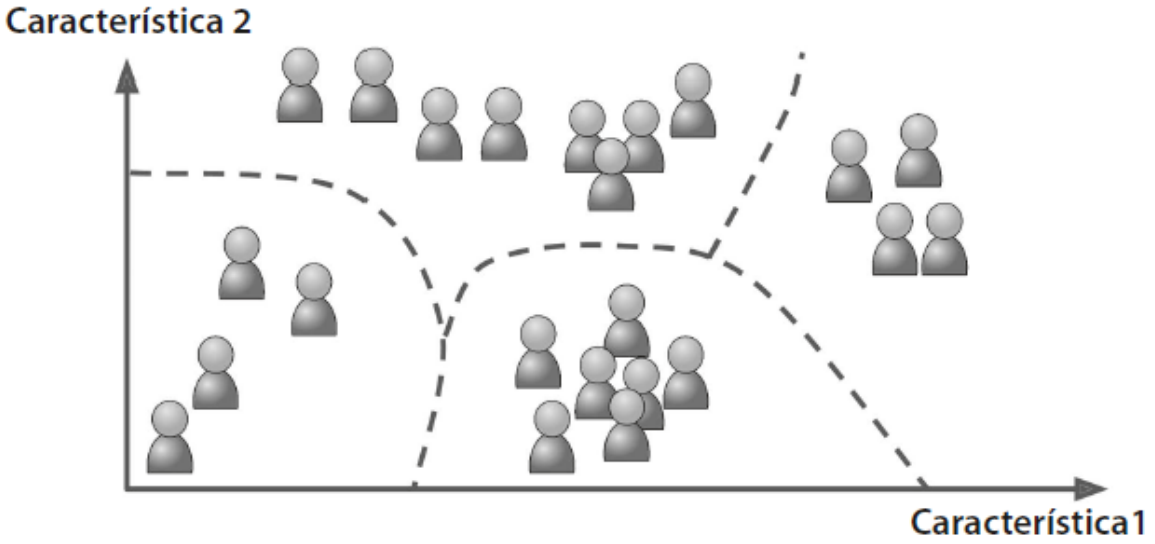

Figura 3.2: Exemplo de uma clusterização, um algoritmo de aprendizado não supervisionado (Géron (2019)).

As tarefas mais comuns de aprendizado não supervisionados são o agrupamento, a detecção de anomalias, a visualização e a redução de dimensionalidade. Os principais algoritmos de aprendizado não supervisionados são (Géron (2019)):

- Agrupamento;
  - K-means;
  - *Clustering* Hierárquico (HCA, do inlês);
  - Maximização da Expectativa.
- Detecção de anomalia;
  - Floresta de isolamento;
  - Uma-classe - SVM.
- Visualização e redução de dimensionalidade;
  - Análise de Componentes Principais (PCA, do inglês);
  - *Locally-Linear Embedding* (LLE);
  - *t-distributed Stochastic Neighbor Embedding* (t-SNE).

Ainda há o aprendizado semi-supervisionado e o aprendizado por reforço, mas não serão dados mais detalhes sobre eles, uma vez que o trabalho não foca nesse tipo de abordagem. Além disso, quando os dados possuem a estrutura de séries temporais, assim como ocorre no 3W *dataset*, uma tarefa bastante comum é fazer previsões, ou seja, usar os dados passados para prever dados futuros, essa tarefa pode ser considerada um aprendizado supervisionado apesar de não haver essa categorização no campo de previsão de séries temporais.

3.2 Aprendizado *Online* ou em *Batch*

A principal diferença entre o aprendizado *online* e *batch* é a capacidade de o algoritmo aprender incrementalmente. No aprendizado *batch*, o treinamento do modelo é feito com todos os dados disponíveis, pois o sistema não consegue aprender incrementalmente. Assim, caso deseje-se adicionar dados novos ao modelo, todo o sistema deverá ser treinado novamente do zero. Por outro lado, no aprendizado *online*, o sistema é treinado incrementalmente enquanto também é utilizado para fazer novas previsões (Géron (2019)).

Nesse sentido, devido ao contexto em que este trabalho está inserido, será empregada a categoria de aprendizado *batch*, pois, para se confirmar a ocorrência de uma anomalia e classifica-lá a fim de viabilizar o uso para treinar o modelo, é necessário uma análise de um especialista. Dessa forma, o sensor virtual será treinado com os dados disponíveis e não haverá treinamento incremental.

3.3 *Pipeline* de um Modelo

O desenvolvimento de um modelo de aprendizado de máquina geralmente segue uma sequência de passos, os quais, segundo Géron (2019), são:

1. Olhar para o contexto geral do problema;
2. Obter dados;
3. Descobrir e visualizar os dados para obter informações;
4. Preparar os dados para os algoritmos de Aprendizado de Máquinas;
5. Selecionar e treinar um modelo;
6. Apresentar a solução;
7. Lançar, monitorar e manter seu sistema.

A seguir, serão detalhados os pontos quatro e cinco em maiores detalhes, os quais carregam maior carga de especificidades.

3.3.1 Preparação dos Dados

Dentro dessa etapa, há três tarefas principais:

- Limpeza e tratamento dos dados;
- Extração de características;
- Seleção de características.

O estágio de limpeza e tratamento dos dados diz respeito à atividade de remover possíveis sujeiras dos dados, por exemplo, valores extremos ou valores ausentes. Em seguida, faz-se a extração de características, isto é, constrói-se as variáveis que pretende-se utilizar nos modelos. No caso deste trabalho, é usada uma biblioteca do *Python* para capturar a média e a mediana de uma quantidade sequencial de observações. Além disso, a etapa de seleção de características é a mais complexa entre as três e merece atenção especial. O processo de seleção de variáveis (*feature selection* do inglês) consiste na seleção de características previamente ao treinamento do modelo a fim de que ele performe melhor. De acordo com Banerjee (2020), os principais benefícios de realizar este processo, são:

- Melhor acurácia;
- Modelos mais simples de interpretar;

- Menores tempos de treinamento;
- Melhor generalização pela redução de sobreajuste;
- Maior facilidade para implementar;
- Menor risco de erro nos dados ao usar o modelo;
- Retirada da redundância das variáveis;
- Melhor comportamento de treinamento pela redução da dimensionalidade do problema.

Além disso, há três métodos de seleção com varias técnicas cada um: os métodos de filtro, os métodos *wrapper* e os métodos *embedded*.

Os métodos de filtro geralmente são usadas como uma ferramenta de limpeza e tratamento dos dados. Além disso, eles são independente do algoritmo que o modelo irá usar, isto é, as variáveis são selecionadas com base em scores de diferentes testes estatísticos para a correlação dela com a variável alvo (Banerjee (2020)). As principais técnicas dos métodos de filtro são:

- Métodos básicos;
  - Remoção de variáveis constantes ou quase constantes;
- Seleção univariada de variáveis;
- Ganho de informação;
- ANOVA F-valor para seleção de variáveis;
- Matriz de correlação.

Nesse contexto, os métodos *wrapper* podem ser entendidos como problemas de procura, isto é, um subconjunto das variáveis é usado para treinar o modelo. Com base nos resultados desse modelo, adiciona-se ou retira-se variáveis desse subconjunto. Além disso, são métodos geralmente computacionalmente caros (Banerjee (2020)). As principais técnicas desse procedimento são:

- *Forward selection*;
- *Backward elimination*;
- Seleção exaustiva de características;
- Eliminação recursiva de variáveis.

Por fim, os métodos *embedded* extraem quais filtros contribuem mais com o treinamento em uma determinada iteração. Ao final, chega-se à relevância das variáveis. Na primeira técnica citada abaixo, isso se traduz em pesos para as características, ou seja, as que não agregam em nada receberão peso zero. Por outro lado, os algoritmos de Árvore de Decisão computam quanto cada variável contribui para a redução de uma medida de impuridade, por exemplo, a impuridade de Gini, ou para o ganho de informação. No desfecho, é dado como *output* uma métrica numérica chamada de importância da variável (Banerjee (2020)).

1. Regressão LASSO e RIDGE;
2. Importância das variáveis com algoritmos de Árvore de Decisão.

É importante destacar que os processos não são excludentes, e sim complementares. Geralmente, utiliza-se diversos métodos de filtro, pelo menor um método *wrapper* e pelo menos um método *embedded*. Além disso, esses processos são aplicados nos dados de treinamento. Ao final, chega-se a um subconjunto final das variáveis que serão levadas para o processo de modelagem.

3.3.2 Selecionar e Treinar um Modelo

Nesse estágio, há três pontos que devem ser observados cuidadosamente:

- Seleção do modelo;
- Otimização dos hiperparâmetros;
- Métricas de performance.

Para selecionar o modelo, primeiro decide-se qual tipo de algoritmo, supervisionado ou não supervisionado, melhor abordará o problema. Com isso, lista-se os modelos disponíveis, então, escolhe-se um subconjunto de variáveis para treinar esses modelos sem nenhum ajuste de hiperparâmetro. Com base em uma métrica de performance e no tempo de treinamento do modelo, escolhe-se o melhor modelo para o contexto. Geralmente, busca-se as melhores métricas com o menor tempo de treinamento.

Em seguida, é feita a otimização de hiperparâmetros. Um hiperparâmetro é um parâmetro de um algoritmo de aprendizado. Como tal, ele não é afetado pelo próprio algoritmo de aprendizado, isto é, deve ser definido antes do treinamento e permanece constante durante ele (Géron (2019)). Assim, otimizar os hiperparâmetros é escolher a melhor combinação deles para um modelo em função da performance, da estabilidade e da redução do sobreajuste. Um exemplo de hiperparâmetro é o *max_depth* dos algoritmos de árvore, que é a profundidade máxima que as árvores podem crescer. Caso ele não seja definido, as árvores

vão crescer o máximo possível, o que pode causar o sobreajuste. Há duas principais maneiras para fazer a otimização: *random search* e *grid search*. No *grid search*, uma série de hiperparâmetros é passada, então, é feita uma combinação de todos os valores possíveis para serem testados. No fim, o *output* é a melhor combinação dos hiperparâmetros passados. De outra maneira, no *random search*, valores aleatórios para cada hiperparâmetro são passados, então, combinações aleatórias. Ao final, o *output* é o mesmo (Géron (2019)).

Nesse contexto, as métricas de performance são essenciais em varias etapas do processo de modelagem, por exemplo, na seleção do tipo de modelo, na avaliação do modelo final, no monitoramento, entre outras. Dessa forma, é necessário escolher a métrica adequada para o tipo de modelo e para o contexto. As principais métricas são:

- Regressão;
  - Erro absoluto médio (MAE) - equação 3.1;
  - Erro quadrático médio (MSE) - equação 3.2;
  - Raiz do erro quadrático médio (RMSE) - equação 3.3;
  - R^2;
  - Erro percentual.
    - Calculado como a diferença percentual absoluta entre o valor previsto e o valor real.
- Classificação;
  - Acurácia (*accuracy*) - equação 3.4;
  - Precisão (*precision*) - equação 3.5;
  - Revocação (*Recall*) - equação 3.6;
  - F1 *score* - equação 3.7;

Para avaliar entender as métricas, será convencionado que y é o *output* real e $\hat{y}$ é o *output* previsto.

A métrica MAE apresenta a distância absoluta média entre os valores reais e os valores previstos e é calculada mediante a equação 3.1. O MSE é a distância quadrática média entre os valores previstos e os valores reais e é calculada por intermédio da equação 3.2. O RMSE é a raiz quadrada do MSE, a vantagem dele é que o resultado é nas mesmas unidades de medida dos valores reais e previstos, o que facilita a interpretação. Ele é calculado pela 3.3. Além disso, o R^2 é um valor que varia de 0 à 1 e ele exibe o quanto o modelo explica a variável resposta.

$$MAE = \frac{1}{N}\sum |y - \hat{y}| \tag{3.1}$$

$$MSE = \frac{1}{N}\sum (y - \hat{y})^2 \tag{3.2}$$

$$RMSE = \sqrt{MSE} \tag{3.3}$$

A fim de entender as métricas de classificação, é necessário compreender a matriz de confusão. Para computá-la, é necessário ter os valores reais e os valores previstos, então, calcula-se quatro valores: a quantidade de verdadeiros positivos (TP), a quantidade de falsos positivos (FP), a quantidade de verdadeiros negativos (TN) e a quantidade de de falsos negativos (FN). Assim, os verdadeiros positivos são os casos em que a previsão e a classe real são o evento de interesse, por exemplo, ao classificar uma operação em fraude ou não fraude, a fraude seria o evento de interesse. Além disso, os falsos positivos ocorrem quando a classe prevista indica para o evento de interesse e a classe real não é o evento de interesse. Por outro lado, os verdadeiros negativos são a coincidência entre as classes previstas e as classes reais para a classe que não é o evento de interesse. Por fim, os falsos negativos ocorrem quando as classes previstas não são da classe do evento de interesse, enquanto as classes reais são do evento de interesse. A Fig. 3.3 apresenta um exemplo de matriz de confusão para um algoritmo de classificação de dígitos manuscritos em 5 ou não 5. Nesse caso, a classe negativa é classificar o digito como outro que não o cinco, e a classe positiva é classificar o digito como cinco.

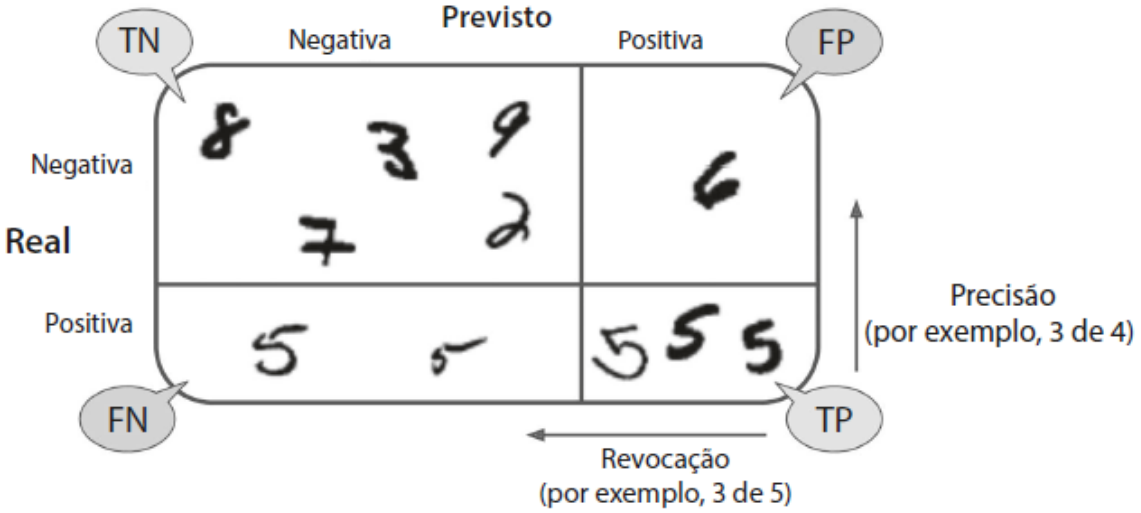

Figura 3.3: Exemplo de uma matriz de confusão para um algoritmo que tenta classificar se um digito manuscrito é o número 5 (Géron (2019)).

A partir dos conceitos apresentados e das equações 3.4, 3.5, 3.6 e 3.7, é possível calcular as quatro métricas. As quatro são usadas concomitantemente na avaliação da performance de um modelo de classificação. Além disso, há um *tradeoff* entre a precisão e o *recall*. Dessa forma, escolhe-se o melhor para avaliar o modelo de acordo com o contexto do problema. Com isso, o F1 *score* surge como uma maneira de ponderar esse *tradeoff*, por conseguinte,

se faz uma métrica única para comparar classificadores.

O *scikit-learn* (Pedregosa et al. (2011)), biblioteca *Python* de aprendizado de máquina, apresenta algumas formas de computar as quatro métricas a partir do parâmetro *average* da função *precision_recall_fscore_support* disponível na biblioteca. Os valores do parâmetro utilizados foram *micro*, *macro* e *none*. Com o valor *micro*, computa-se as métricas contando o verdadeiros positivos, os falsos positivos, os verdadeiros negativos e os falsos negativos de forma global sem distinguir por classe, por exemplo, anomalia e não anomalia. Por outro lado, o valor *macro* calcula as métricas para todas as classes, então, faz uma média aritmética entre os valores das classes. Por fim, o valor *none* apenas calcula as métricas para cada classe sem fazer nenhuma ponderação.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.4}$$

$$Precision = \frac{TP}{TP + FP} \tag{3.5}$$

$$Recall = \frac{TP}{TP + FN} \tag{3.6}$$

$$F1 = 2 \times \left(\frac{Precision \times Recall}{Precision + Recall} \right) \tag{3.7}$$

Ademais, em (Allwright (2022)), é apresentada uma regra de ouro para a comparação do F1 *score*:

F1 *score*	Interpretação
> 0,9	Muito bom
0,8 - 0,9	Boom
0,5 - 0,8	Ok
< 0,5	Ruim

Tabela 3.1: Regra de ouro para o F1 *score* apresentada por (Allwright (2022)).

Vale lembrar que existe uma gama maior de métricas do que as apresentadas. Entretanto, essas são as principais e que mais aparecerão neste trabalho.

3.4 Algoritmos Aplicados

3.4.1 Aprendizado *Ensemble*

Antes de detalhar cada técnica, é necessário entender o conceito dos métodos *ensemble*, os quais são usados em mais de um dos algoritmos que serão apresentados. Os métodos

ensemble combinam predições de diversos modelos geralmente fracos a fim de melhorar a capacidade preditiva (Mehta et al. (2019)). Devido à performance superior, os métodos *ensemble* são uma das mais poderosas e utilizadas ideias no Aprendizado de Máquina. Há quatro tipos principais de métodos *ensemble*, o*bagging*, *o pasting*, o *boosting* e o *stacking*. O primeiro e o último serão detalhados, pois serão os utilizados neste trabalho.

No *bagging*, os classificadores são treinados com amostras aleatórias com reposição dos dados. Ao final, é feita uma contagem de previsões de cada classificador, e a classe com maior quantidade de votos será a previsão do modelo agregado, isto é, para problemas de classificação. Para um problema de regressão, é feita a média do valor previsto por cada regressor. Um exemplo de algoritmo que utiliza o *bagging* é a Floresta Aleatória. Por outro lado, a ideia do *boosting* é treinar vários previsores fracos sequencialmente, cada um tentando corrigir seu antecessor. Os algoritmos mais populares de *boosting* são o *AdaBoost* e o *Gradient Boosting* (Géron (2019)).

3.4.2 Supervisionados

3.4.2.1 Floresta Aleatória

A Floresta Aleatória é constituída por um conjunto de Árvores de Decisão aleatórias, classificadores *tree-based*. Uma Árvore de Decisão usa uma séria de questões hierárquicas para separar os dados criando ramos e folhas. Em cada ramo, o dado é separado em subconjuntos menores (Mehta et al. (2019)). Dessa forma, a Floresta Aleatória utiliza o *bagging*, amostragem aleatória com reposição, tanto das observações como das variáveis para randomizar a solução ainda mais, reduzindo o sobreajuste. Vargas (2019) utiliza a Floresta Aleatória no *benchmark* para avaliar o impacto da utilização de instâncias desenhadas e de instâncias simuladas. A Fig. 3.4 mostra um exemplo de uma Árvore de Decisão com três variáveis explicativas e com uma variável alvo com quatro possíveis classes.

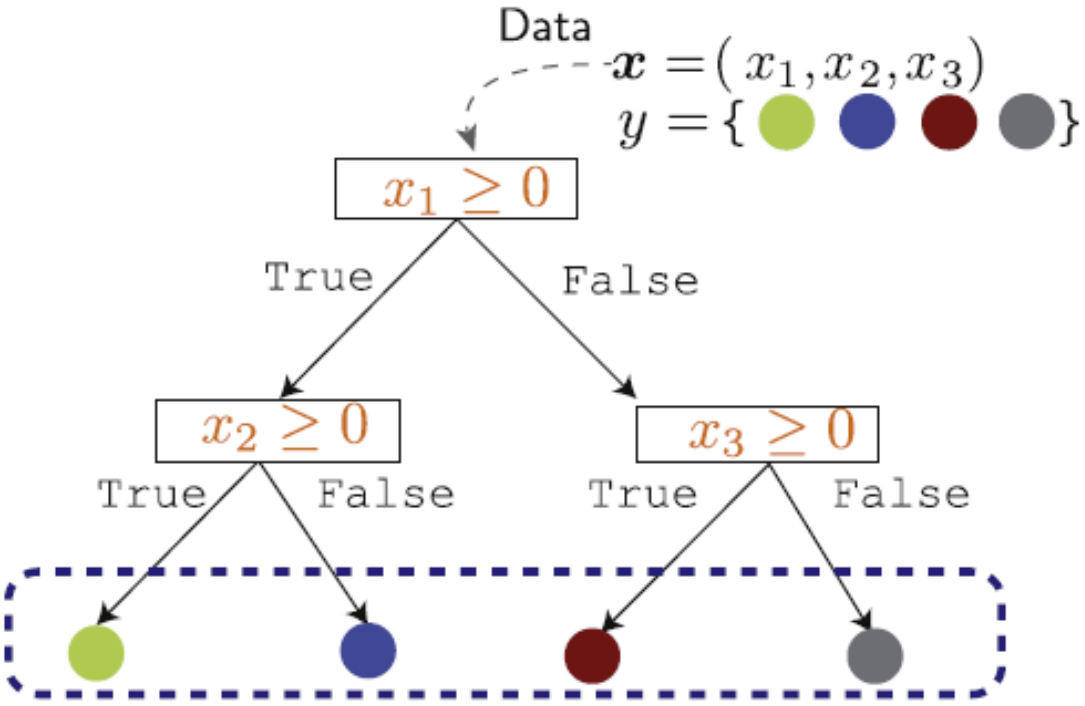

Figura 3.4: Exemplo de uma Árvore de Decisão (Mehta et al. (2019)).

3.4.2.2 XGBoost

O XGBoost é uma biblioteca que implementa o gradiente *boosting* de forma distribuída, eficiente, flexível e portátil (Chen e Guestrin (2016)). Geralmente, o próprio algoritmo é chamado de XGBoost, e é assim que será feito neste trabalho. Esse modelo também faz uso de Árvores de Decisão, mas é um algoritmo de *boosting*. Nele, o primeiro classificador é treinado para prever o alvo, em seguida, um novo classificador é treinado com os erros residuais feitos pelo anterior e assim por diante, ou seja, cada classificador tenta ajustar os erros do antecessor. No fim, o valor previsto é igual a soma ponderada por pesos da previsão de cada classificador. A Fig. 3.5 exibe um exemplo de ajuste de um XGBoost aos dados. Na coluna da esquerda, são apresentadas as previsões de três Árvores de Decisão, e, na direita, são mostradas as previsões do *ensemble*. Na primeira linha, a árvore é treinada para prever os valores alvos. Na segunda linha, a árvore é treinada para prever o erro residual da primeira árvore. Por fim, na terceira linha, a árvore é treinada para prever o erro residual da segunda árvore (Géron (2019)). Além disso, Vargas (2019) também faz uso do XGBoost no *benchmark* do impacto das instâncias fabricadas.

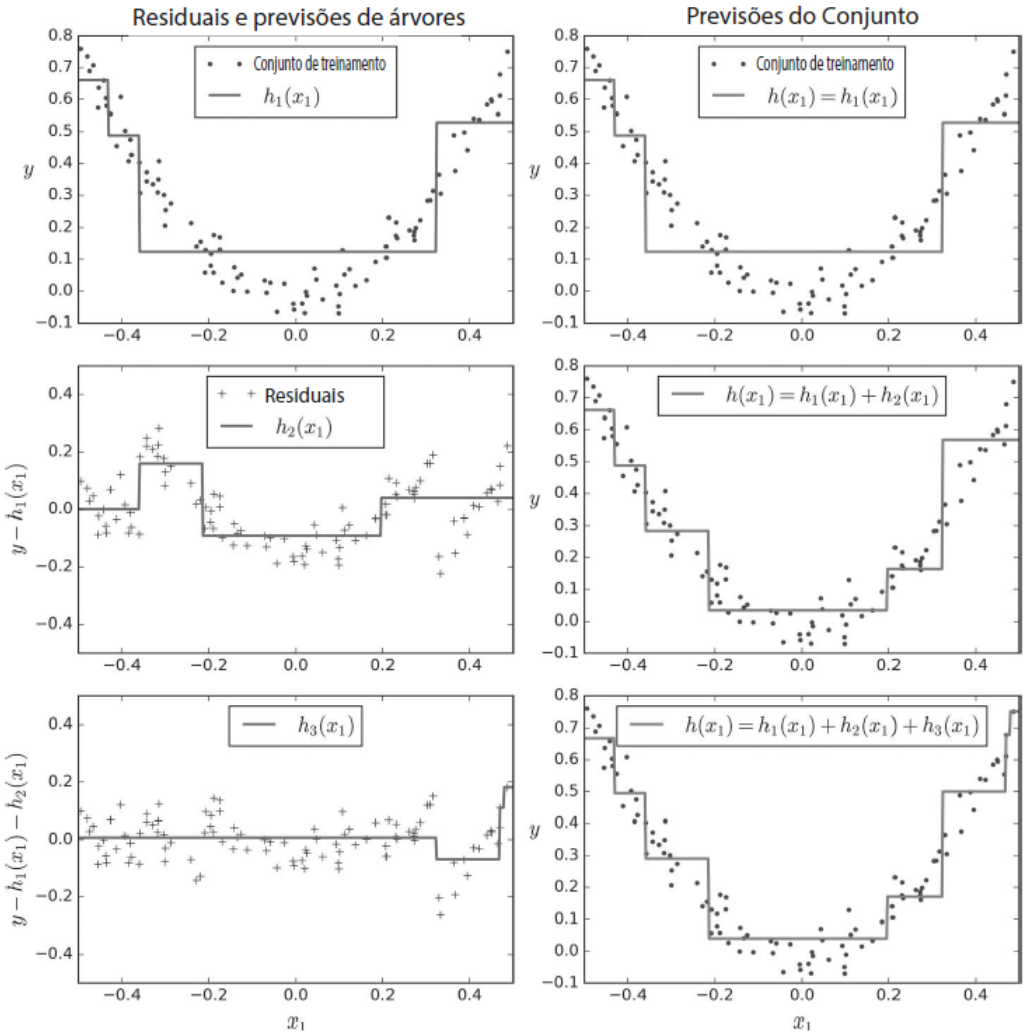

Figura 3.5: Exemplo das etapas de treinamento de um XGBoost (Géron (2019)).

3.4.2.3 Redes Neurais

Mehta et al. (2019) traz uma definição muito clara de Rede Neural: "As redes neurais são modelos não lineares de inspiração neural para aprendizado supervisionado. Eles são extensões mais poderosas de métodos de aprendizado supervisionado, como regressão linear e logística". A arquitetura mais simples de uma Rede Neural é a *Perceptron* ou neurônio. Conforme mostrado na Fig. 3.6, a *perceptron* é composta por entradas, pesos, um viés, uma função e saídas. O processo para gerar a saída é somar o produto dos pesos pelas entradas com o viés e aplicar a função de transformação na soma. Uma Rede Neural consiste na junção dos neurônios em camadas. Os dados de saída de uma camada servem como dados de entradas para outras camadas. Assim, a primeira camada é chamada de camada de entrada, e a última camada é chamada de camada de saída. Além disso, é destaca-se que as redes neurais podem ser usadas tanto para tarefas de classificação como para regressão. A Fig. 3.7 exibe um exemplo de rede ((Géron (2019)), (Mehta et al. (2019))).

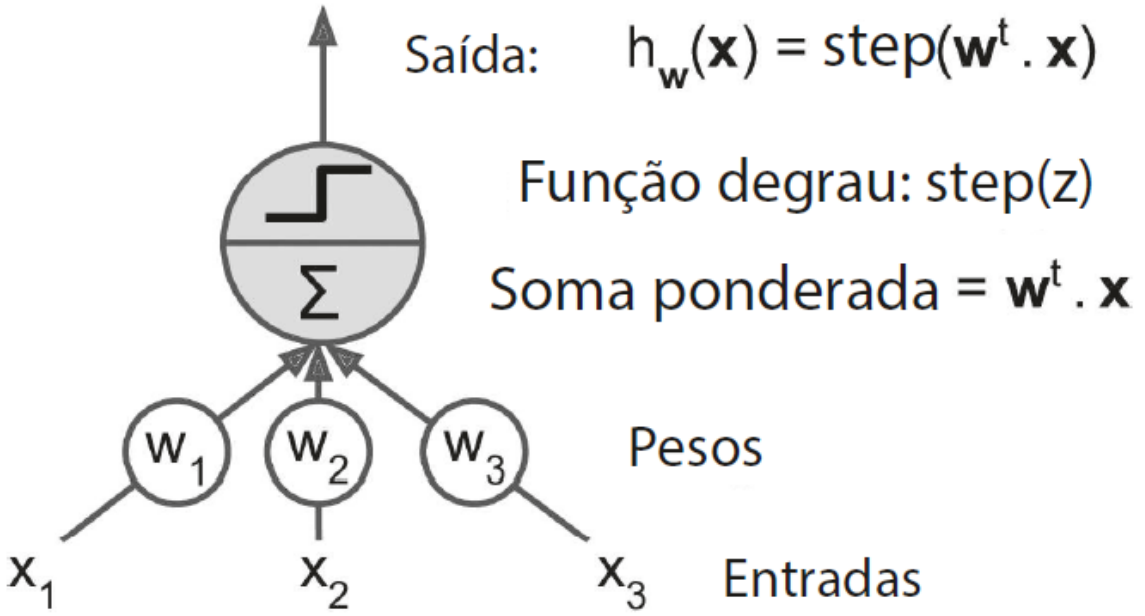

Figura 3.6: *Perceptron*, a unidade mais básica de uma Rede Neural artificial (Géron (2019)).

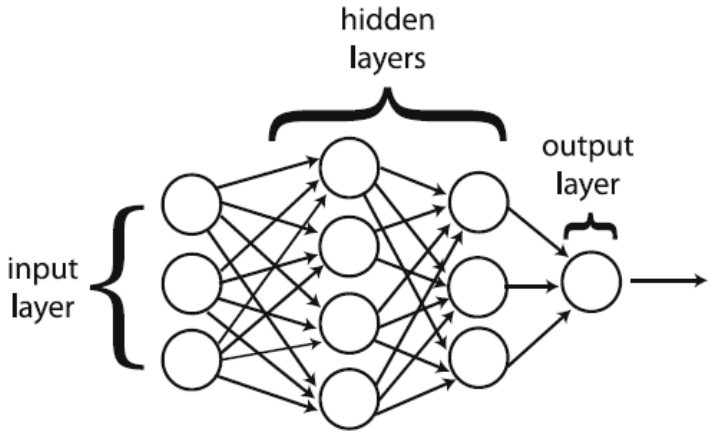

Figura 3.7: Exemplo de Rede Neural (Mehta et al. (2019)).

Assim, há diversos parâmetros que influenciam na performance e na convergência de uma Rede Neural. Um parâmetro importante é a função de ativação para transformar a soma de pesos, a qual pode mudar de acordo com o tipo de saída desejado. A Fig. 3.8 apresenta alguns exemplos de função de ativação, inclusive a que é usada no presente trabalho, ReLU. Além disso, outros parâmetros importantes são a quantidade de neurônios em cada camada, a quantidade de camadas ocultas, isto é, nem de saída e nem de entrada, a quantidade de épocas, o tamanho de batch e a taxa de aprendizado. Quando o modelo é ajustado, escolhe-se a quantidade de amostras que serão passadas de uma vez para ele, isto é chamado de

tamanho de batch. Já a quantidade de épocas é a quantidade de vezes que o modelo passará completamente pelos dados de treino para serem ajustado. Por fim, a taxa de aprendizado é a velocidade com que o algoritmo tentará convergir os parâmetros treinados para a solução ótima.

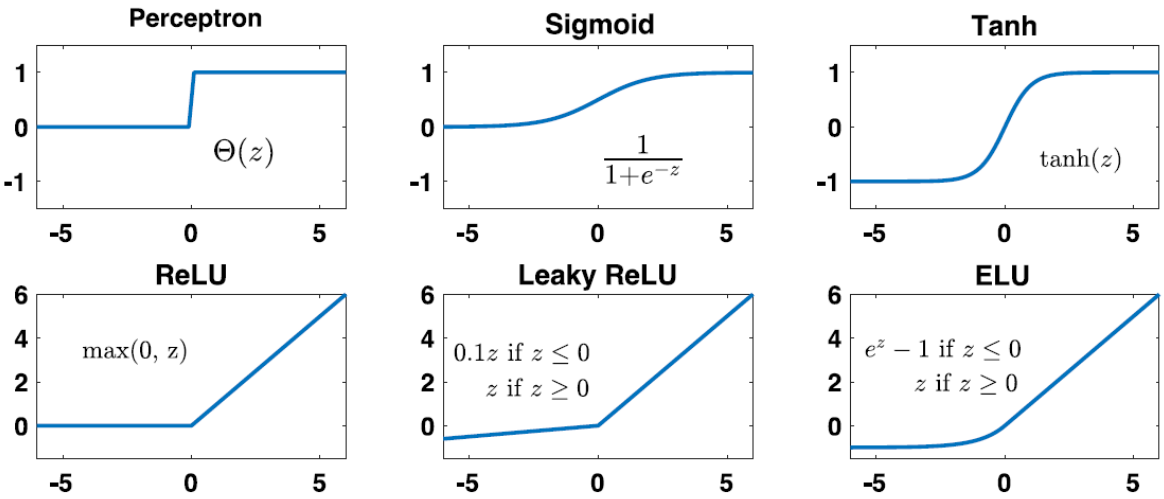

Figura 3.8: Exemplos de funções de ativação comumente utilizadas em redes neurais (Mehta et al. (2019)).

Diante desse padrão de redes neurais, chamada de camada densa ou de *Multilayer Perceptron*, MLP, surgem outras arquiteturas de redes para tratar problemas específicos, por exemplo, Redes Neurais Convolucionais para lidar principalmente com soluções de visão computacional e de imagens e Redes Neurais Recorrentes para encarregar-se de dados em sequências, por exemplo, séries temporais e textos.

No contexto do problema de previsão de dados futuros, o presente trabalho faz uso de Redes Neurais Recorrentes além da Rede Neural convencional. A principal característica de uma Rede Neural Recorrente é que a *perceptron* recebe os dados de entrada e os seus próprios dados de saída do instante anterior. Dessa forma, o neurônio passa levar em consideração dados passados no cálculo dos *outputs*, ou seja, o neurônio passa a ter um conjunto de pesos para os dados de entrada e um conjunto de pesos para os dados de saída dos instantes anteriores (Géron (2019)). A Fig. 3.9 exibe um neurônio de uma Rede Neural Recorrente desenrolado através do tempo.

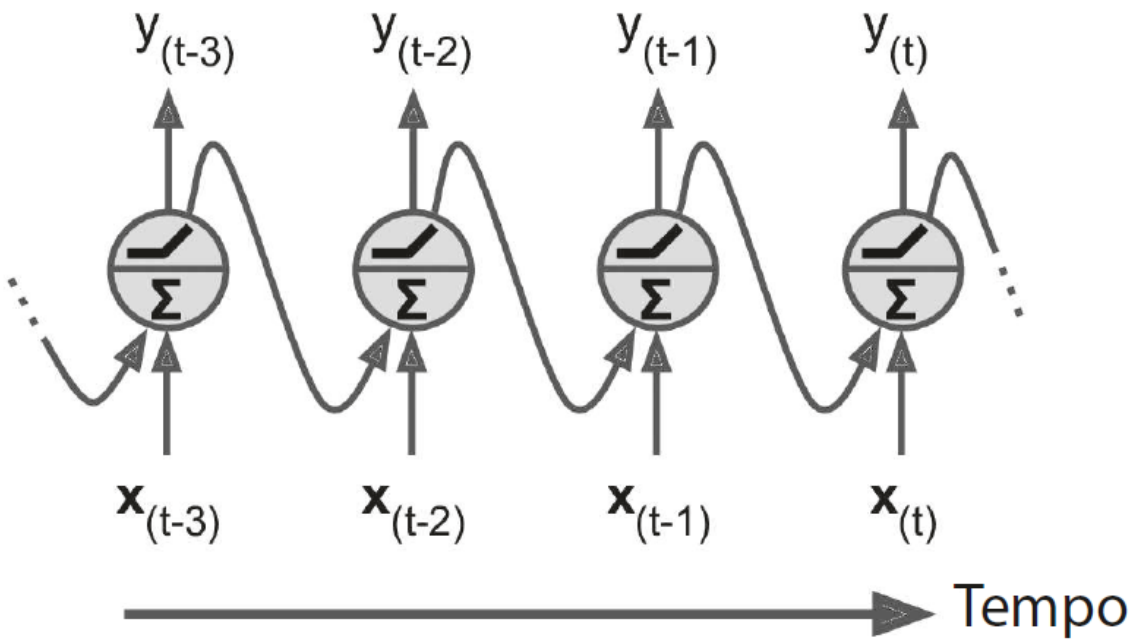

Figura 3.9: Rede Neural Recorrente desenrolada através do tempo (Géron (2019)).

Entretanto, a Rede Neural Recorrente simples apresenta problemas de memória para dados de longo prazo e de treinamento. Assim, surge outras arquiteturas de Rede Neural Recorrente para sanar essas dificuldades. Uma delas, que é utilizada nesse trabalho, é a célula *Long Short-Term Memory*, LSTM, a qual é composta por alguns portões, pelos dados de enrtada e por dois estados, o de curto prazo e o de longo prazo, os quais podem ser observados na Fig. 3.10. Conforme é possível ver na Fig. 3.10, há mais de uma camada em cada célula. O *Forget Gate*, porta esquecida, controla quais partes do estado de longo prazo devem ser deletados. o *Input Gate*, porta de entrada, controla quais parte das entradas atuais e dos estados anteriores de curto prazo devem ser adicionados ao estado de longo prazo. O *Output Gate*, porta de saída, controla quais partes do estado de longo prazo devem ser lidas e geradas neste intervalo de tempo, tanto para o estado de curto prazo quanto para a saída. A camada restante é responsável por analisar as entradas atuais e os estados anteriores de curto prazo. Assim, a célula LSTM é eficaz em identificar padrões de longo prazo em dados sequenciais, pois ela reconhece entradas importantes, armazena esse dado no estado de longo prazo, preserva a informação pelo tempo preciso e utiliza o dado nos instantes necessários (Géron (2019)).

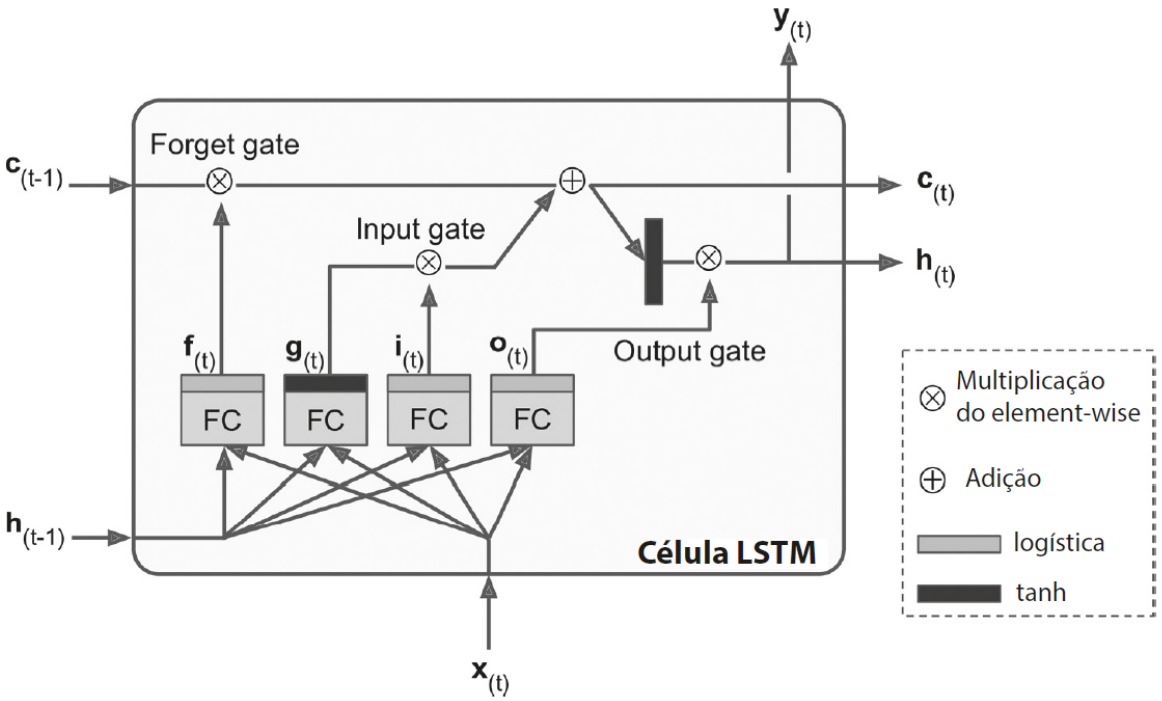

Figura 3.10: Célula LSTM (Géron (2019)).

3.4.3 Não Supervisionados

3.4.3.1 Floresta de Isolamento

A Floresta de Isolamento é um algoritmo que foi desenvolvido por Liu, Ting e Zhou (2008) com o objetivo de abordar a detecção de anomalias. Ele se baseia no princípio de isolar as anomalias ao invés de criar um perfil para instâncias normais. Nesse sentido, esse também é um modelo *ensemble*, agregando diversas Árvores de Isolamento. Devido à maior suscetibilidade ao isolamento, anomalias são isoladas mais próximas à raiz da árvore, ou seja, elas têm um um caminho médio curto. Dessa forma, ao criar uma floresta de árvores aleatórias que produzem caminhos mais curtos para um determinado ponto, então, há grandes chances de ele ser uma anomalia. A Fig. 3.11 apresenta um exemplo de isolamento de duas instâncias, sendo a da esquerda, x_i, uma instância normal, e a da direita, x_0, uma instância anomalia. Para isolar, a instância normal são necessárias doze partições aleatórias, enquanto que, para a instância anomalia, são necessárias apenas quatro partições. As partições são geradas mediante a seleção aleatória de uma variável e um valor para a ramificação. Ou seja, particionamento recursivo é pode ser representado por uma estrutura de árvore, e a quantidade de partições necessárias para isolar um ponto corresponde ao caminho entre o nó da raiz e o nó terminal. Ao final, é calculada a média do comprimento dos caminhos em todas as árvores. Junior (2022) e Vargas (2019) empregam Floresta de Isolamento para detectar anomalias nos dados do 3W *dataset*.

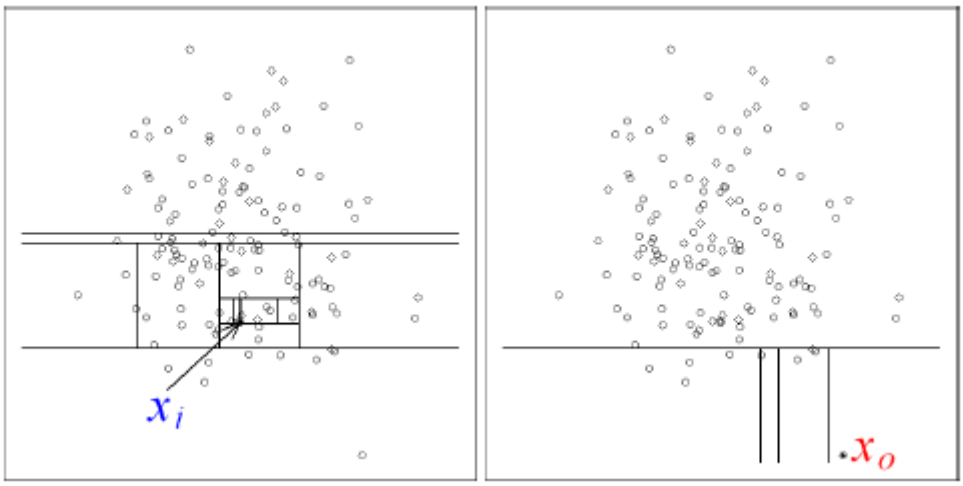

Figura 3.11: Exemplo de isolamento de dois pontos x_i (normal) e x_0 (anomalia) (Liu, Ting e Zhou (2008)).

3.4.3.2 One-class SVM

A Máquina de Vetores de Suporte (SVM) é um algoritmo que separa as classes por intermédio de um hiperplano no espaço, otimizando as margens entre as classes, isto é, a distância entre o hiperplano e o ponto mais próximo de cada classe. A Fig. 3.12 exibe como o SVM atua, os elementos principais são a linha sólida do meio, o hiperplano, o quadrado e o círculo circunscritos e as linhas pontilhadas. O quadrado e o círculo com circunscritos são os vetores de suporte, eles irão determinar o tamanho da margem, isto é, a distância entre a linha pontilhada e o hiperplano. A partir dessa margem do hiperplano, são classificados os dados. Além disso, para dados mais não lineares, o SVM utiliza *kernels*, que são funções lineares que mapeiam o espaço de variáveis em outro de maior dimensionalidade no qual a separabilidade entre as classes pode ser maior. A funções *kernel* disponíveis são o polinomial (POLY), o RBF Gaussiano (RBF) e o Sigmoid ((Géron (2019)), (Vargas (2019))).

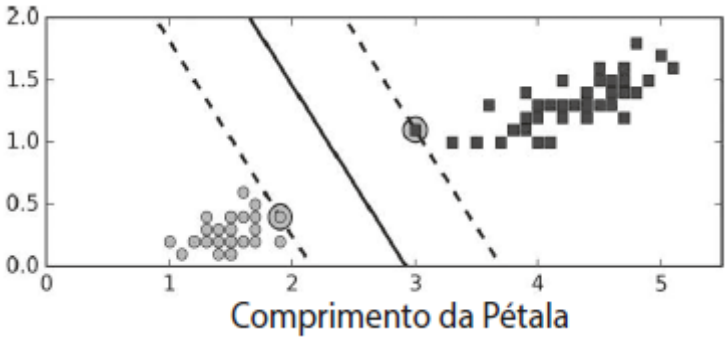

Figura 3.12: Exemplo da atuação de um modelo de SVM para classificação de flores em Iris-Versicolour (quadrados) e em Iris-Setosa (círculos). No eixo y, é a variável Largura da Pétala (Géron (2019)).

A partir desse algoritmo base, desenvolve-se a Máquina de Vetores de Suporte de uma

classe (*One Class* SVM). Nesta nova versão, o algoritmo deixa de ser supervisionado e passa a ser não supervisionado. Dessa forma, o *One Class* SVM tenta separar as instâncias da origem, maximizando da distância entre um hiperplano e a origem. O resultado é uma função que captura as regiões onde está a densidade de probabilidade dos dados de entrada. Assim, se os dados ficarem nessas regiões, serão instâncias normais, caso contrário, serão anomalias (Vlasveld (2013)). Tanto Junior (2022) como Vargas (2019) empregam *One Class* SVM com diferentes *kernels* para detectar anomalias nos dados do 3W *dataset*.

3.4.3.3 *Local Outlier Factor*

Esse algoritmo é baseado em densidade e calcula o *Local Outlier Factor* (LOF), o qual quantifica o grau com que um objeto é uma anomalia mediante a captura do grau de isolamento desse objeto em relação a vizinhança (Breunig et al. (2000)). A pontuação LOF é computada como a razão entre a densidade média dos k vizinhos mais próximos de uma instância e a densidade local dela. Quando as a densidade é semelhante a dos vizinhos, geralmente é uma instância normal. Por outro lado, anomalias geralmente são mais isoladas dos pontos mais próximos, ou seja, a densidade será menor, e maior será o LOF (Junior (2022)). Neste trabalho, esse algoritmo é usado com o hiperparâmetro *novelty* = *True*, o que viabiliza a utilização do modelo em dados novos. Além disso, Junior (2022) usa esse algoritmo para detectar anomalias no 3W *dataset*.

3.4.3.4 Envelope Elíptico

Esse algoritmo utiliza a técnica *minimum covariance determinant* para separar as instâncias normais das anomalias. Ele assume que as instâncias normais são geradas a partir de uma distribuição Gaussiana. Dessa forma, estima-se os parâmetros da distribuição, os quais dão o formato de uma elipse que envolve os objetos normais, isto é, um envelope elíptico. Assim, os dados que ficam de fora dessa elipse são considerados anomalias (Géron (2019)). Por fim, Junior (2022) também implementa esse algoritmo para detectar eventos indesejados no 3W *dataset*.

Capítulo 4

Metodologia

4.1 Reprodução de Resultados

Para reproduzir os resultados dos trabalhos, utilizou-se e adaptou-se os códigos em Python disponibilizados por Vargas (2019) no *Github* <https://github.com/ricardovvargas/3w_dataset>. Nesse código, o autor utiliza os algoritmos de Floresta de Isolamento e de *One Class* SVM com os quatro tipos de *kernel*. A principal adaptação feita foi adicionar os algoritmos Envelope Elíptico e *Local Outlier Factor* usados por Junior (2022). Assim, serão destacados os principais pontos da metodologia utilizada para chegar ao resultados.

4.1.1 Tratamento dos Dados

Inicialmente, são determinados alguns parâmetros que para serem seguidos:

- Apenas instâncias reais com algum tipo de evento e que possuam períodos normais e outro tipo de período foram usadas;
- As instâncias devem ter no mínimo 1200 observações no período normal;
- Amostragem por janela deslizante de 180 observações;
- Quantidade máxima de amostras por período é quinze;
- Percentual de valores ausentes de no máximo 10%;
- Desvio padrão mínimo da variável deve ser de 0,01;
- Não haverá seleção de hiperparâmetros.

A Fig. 4.1 ilustra o estratégia de amostragem de janela deslizante citada acima. Dessa forma, a partir dos pontos acima determinados, foram extraídas as amostras. Em seguida,

a biblioteca *tsfresh* foi utilizada para extrair características das observações dentro de cada amostra, as quais foram a mediana, a média, o desvio padrão, a variância, o RMS, o máximo e o mínimo. Assim, para cada variável de processo, essas métricas foram calculadas. Além disso, os primeiros 60% das amostras de períodos normais foram usadas como dados de treino, enquanto os 40% seguintes e as amostras dos períodos transiente e estável de anomalia foram usadas como dados e teste. Por fim, antes de passar para a etapa de treinamento do modelo, os dados foram normalizados mediante a utilização da média e do desvio padrão dos dados de treinamento conforme a equação 4.1. Em posse desses dois valores, os dados de teste também foram normalizados.

$$z = \frac{x - \mu}{\sigma} \tag{4.1}$$

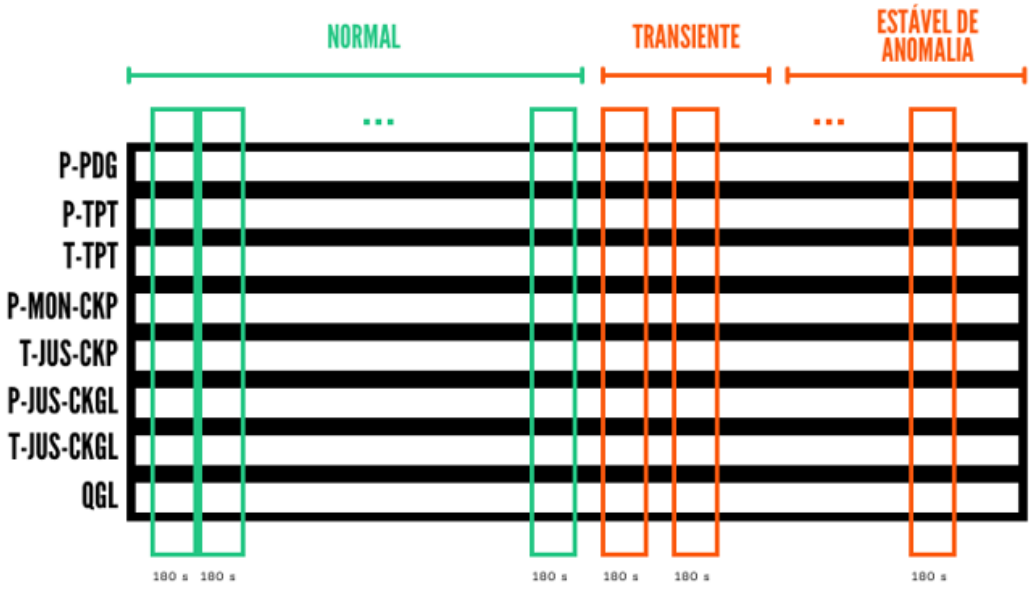

Figura 4.1: Esquema de amostragem com janelas deslizantes de 180 observações. Adaptada de (Junior (2022)).

4.1.2 Treinamento dos Modelos e Avaliação dos Resultados

O treinamento e a avaliação dos resultados em cadeia com a amostragem e com a extração de características, isto é, dentro de um *loop* eram seguidos os seguintes passos para cada instância:

1. Extrair amostras;
2. Filtrar variáveis boas de acordo com os parâmetros determinados;
3. Normalizar os dados;
4. Extrair características;

5. Treinar o modelo;

6. Calcular as métricas de performance.

Ao final desse *loop*, há as três métricas (precisão, *recall* e F1 *score*), estimadas com o parâmetro *micro*, para cada instância e modelo utilizados. Com isso, calcula-se uma média das métricas para chegar-se a um valor final. Dessa forma, os modelos aplicados foram todos de aprendizado não supervisionado, a saber:

- *One Class - SVM* com os 4 *kernels*;
- Floresta de Isolamento;
- *Local Outlier Factor*;
- Envelope Elíptico.

Essa abordagem é proposta pelos autores puramente como *Benchmark* para soluções posteriores. Dessa forma, ela não é adequada para o desenvolvimento de um sensor virtual, uma vez que não são utilizados todos os dados, que não são detectadas todas as instâncias, que o treinamento e o teste são feitos com poucos dados, que o treinamento e o teste são realizados com dados de um único tipo de instância e que instâncias reais normais foram descartadas. Assim, fez-se necessário abordar o problema de outra maneira.

4.2 Adaptação de resultados

Em (Junior (2022)), é empregada uma outra perspectiva para o problema, na qual agrupa-se os dados das amostras de treino e de teste. Para desenvolver o sensor virtual para detecção de anomalias, baseou-se nessa abordagem, mas ela foi adaptada. Os principais pontos de adaptação foram:

- Utilização das instâncias reais normais como dados de treino e dados de teste;
  - A abordagem anterior descarta esses dados. Entretanto, eles representam os dados reais capturados por sensores em operação normal. Ou seja, é o contexto ideal funcionamento. Dessa forma, modelar esse comportamento para detectar anomalias traduz a realidade.
- Extração de amostras de todos os tipos de eventos independente da quantidade de observações no período normal;
- Não há limitação para a quantidade de amostras;
- Utilização de algoritmos de aprendizado supervisionado e não supervisionado;

  - Na abordagem anterior, mesmo com os dados rotulados, os algoritmos não supervisionados foram empregados por serem apropriados para a detecção de anomalias e devido à baixa quantidade de dados e a um certo desbalanceamento da classe de interesse, o que poderia causar um sobreajuste aos dados com um algoritmo supervisionado. Entretanto, nesta outra abordagem, há maior quantidade de dados e um balanceamento maior das classes, por isso, foram testados modelos supervisionados também.

- Utilização dos dados desenhados como dados de validação do modelo;

  - Como a atividade de detecção de anomalias passa é feita por um especialista, as instâncias desenhadas, neste trabalho, são entendidas como um *Benchmark*.

- Observações não rotuladas foram desprezadas;

- A variável de processo T-JUS-CKGL foi retirada, pois ela apresenta valores ausentes para 100% das observações;

- As outras variáveis não foram retiradas por nenhum critério.

Alguns métodos da abordagem anterior foram reutilizados, a saber: a estratégia de amostragem, as características extraídas e a normalização dos dados. Além disso, foi desenvolvido um novo código em *Python*, aproveitando algumas partes do código anterior.

Dessa forma, os passos seguidos foram os seguintes:

1. Gerar amostras de instâncias reais com extração de características e com a retirada da variável T-JUS-CKGL;
2. Gerar amostras de instâncias desenhadas com extração de características e com a retirada da variável T-JUS-CKGL;
3. Rearranjar os dados;
4. Normalizar os dados;
5. Ajustar nos dados;
6. Treinar os modelos;
7. Calcular métricas de performance nos dados de teste (instâncias reais);
8. Calcular métricas de performance nos dados de validação (instâncias desenhadas).

Nesse contexto, inicialmente todas as amostras de instâncias reais seriam utilizadas como dados de treinamento e as amostras de períodos normais, transientes e estáveis de anomalias de instâncias de eventos seriam utilizadas como amostra de teste. Entretanto, havia

uma quantidade consideravelmente maior de dados de treino. Portanto, amostro-se 20% dos dados de treinamento para transformá-los em dados de teste. Além disso, criou-se oito *datasets*, cada um com todas as amostras de períodos normais de instâncias não normais e com os períodos transiente e estável de anomalia de um determinando tipo de evento para avaliar os resultados do modelo separadamente por tipo de anomalia. Todo esse processo foi feito para treinar os algoritmos não supervisionados, ou seja, são treinados apenas com instâncias normais. Por fim, criou-se um *dataset* com todas as amostras juntas para realizar uma amostragem de 30% para os dados de teste e o resto para dados de treino dos modelos supervisionado, isto é, os modelos foram treinados em amostras normais e não normais.

Ademais, o ajuste dos dados posterior a normalização foi feito para desconsiderar dados com erro que ainda restaram. Vale ressaltar que a normalização aplicada aos dados de instâncias desenhadas foi feita com os mesmos parâmetros encontrados em cada um dos *datasets* de treino. Além dos modelos utilizados nos primeiros resultados, aplicou-se o aprendizado supervisionado com o XGBoost e com a Floresta Aleatória tanto sem a otimização de hiperparâmetros como com a otimização de hiperparâmetro *max_depth* a fim de avaliar os efeitos do sobreajuste. Por fim, foram calculadas as mesmas métricas de anteriormente, mas empregou-se os parâmetros *micro*, *macro* e *none*.

4.3 Otimização dos resultados

A próxima etapa desenvolvida foi a otimização dos resultados obtidos na etapa de adaptação dos resultados. Assim, testou-se outras quantidades de observações na amostragem por janela deslizante, conjuntos de variáveis menores de acordo com uma etapa de seleção de variáveis e modelos customizados para cada evento.

Nesse contexto, para que os resultados fossem comparáveis, optou-se por seguir uma proporção padrão entre os dados de treino e de teste e entre dados sem anomalia e com anomalia, a saber: 80% dos dados para o treino e 20% para o teste, e 90% dos dados sem anomalia e 10% com anomalia. Devido ao fato de a quantidade de pontos variarem de acordo com a janela experimentada, com o tipo de evento e com o tipo de abordagem, supervisionada ou não supervisionada, foi implementado um problema de otimização linear para determina a quantidade de pontos ótima para adequar o dados à proporção citada.

4.3.1 Janelas diversas e modelos customizados

Inicialmente, foram determinados os tamanhos das janelas que seriam testadas. Para isso, utilizou-se como referência os dados da Fig. 2.5. Além disso, extraiu-se a média e a mediana da quantidade de observações por tipo de evento. Diante desses dados, foram selecionadas até cinco tamanhos de janelas para cada tipo de evento além da janela padrão de 180 segundos. É importante destacar que uma parte dos eventos não possuem séries

temporais com quantidade de pontos suficientes para cobrir uma janela de referência.

A partir disso, foram extraídos dados de acordo com os parâmetros determinados, e os modelos foram treinados por evento, isto é, modelos customizados. Além disso, treinou-se modelos genéricos, isto é, um modelo para todos os eventos ao mesmo tempo com a janela padrão de 180 segundos. É importante destacar que a comparação entre modelos genéricos e customizados foi feita apenas nos modelos supervisionados. Ao fim desse processo, os resultados foram comparados para determinar os modelos que seriam utilizados em etapas seguintes, e os modelos foram testados nas instâncias desenhadas e simuladas.

4.3.2 Seleção de variáveis

Diante da etapa anterior, empregou-se a importância relativa das variáveis calculada pelo modelo genérico escolhido na etapa anterior para ranquear as variáveis e para treinar modelos com subconjuntos das variáveis, os subconjuntos experimentados foram:

- Variáveis com importância maior que 0%;
- Variáveis com importância maior que 1%;
- Top 10 variáveis de maior importância;
- Top 15 variáveis de maior importância;
- Top 20 variáveis de maior importância;
- Top 25 variáveis de maior importância.

Nesse contexto, os modelos treinados foram testados nas instâncias desenhadas e simuladas a fim de comparar as métricas e, em seguida, de selecionar o modelo final supervisionado e não supervisionado.

4.4 Previsão de dados

Primeiramente, foram extraídas instâncias reais separadamente por variável que atendiam a dois critérios: não possuir valores faltantes e ter desvio superior à 0,01, isto é, instâncias minimamente livres de problemas associados aos sensores. Esse critério é estabelecido para que o modelo seja treinado com dados mais limpos, ou seja, para que esses erros não possam interferir no resultado. Com isso, não houve uma quantidade satisfatória de instâncias boa da variável QGL para treinar modelo. Então, essa variável não foi levada para os passos seguintes do desenvolvimento.

Em seguida, foram treinados modelos com diferentes parâmetros para determinar, mediante tentativa e erro, os parâmetros finais a serem utilizados nos passos subsequentes. Parâmetros considerados ruins são os que fazem o modelo não convergir ou que aumentam muito o custo computacional para treinar os modelos. Ademais, a estratégia empregada na modelagem do problema foi empregar a janela atual para prever a próxima janela. Essa estratégia permite analisar os efeitos

Posteriormente, foram determinados os parâmetros utilizados no modelo mediante tentativa e erro. Parâmetros considerados ruins são os que fazem o modelo não convergir ou que aumentam muito o custo computacional do processo. Nesse sentido, a metodologia empregada para treinar os modelos foi utilizar uma janela como dado de entrada para prever a próxima janela. Assim, os parâmetros e as abordagens utilizados no treinamento dos modelos de previsão de dados:

- Treinar um modelo para cada variável apenas com instâncias sem anomalia;
  - Amostragem de 40 instâncias devido a limitação computacional;
- Treinar um modelo para cada variável com instâncias com e sem anomalia;
  - Amostragem de 40 instâncias sem anomalia e, no máximo, 5 instâncias com anomalia por evento devido a limitação computacional;
- Utilização de uma camada LSTM com 64 unidades;
- Utilização de duas camadas densas de 32 unidades com função de ativação ReLU;
- Função de perda: erro quadrático médio;
- Otimizador: Adam;
- Quantidade de épocas: 20;
- Tamanho de batch: 500.

Ao fim, os resultados foram analisados com instâncias reais e com a métricas erro percentual.

4.5 Mesclagem de modelos

A partir da finalização de todos os modelos, é realizada a etapa de mesclagem deles para se chegar ao produto final pretendido pelo projeto. Para isso, seguiram-se os seguintes passos:

1. Extração de instâncias reais que atendem aos critérios citados anteriormente em todas as seis variáveis necessárias (P-PDG, P-TPT, T-TPT, P-MON-CKP, T-JUS-CKP e P-JUS-CKGL).

2. Previsão dos dados e atribuição a classe do dado real aos dados previstos, isto é, normal, transiente e estável de anomalia tanto nas instâncias reais extraídas quanto em instâncias desenhadas e simuladas;

3. Tratamento dos dados previstos para o formato do problema de detecção;

4. Aplicação dos modelos de detecção supervisionado e não supervisionado;

5. Avaliação dos resultados.

    (a) Cálculo da métrica de referência ponto a ponto entre uma janela de 180 segundos prevista e a janela real;

    (b) Cálculo da média da métrica entre os 180 pontos da janela prevista.

Capítulo 5

Resultados e Discussão

5.1 Algoritmos Não Supervisionados

Diante do exposto, a tabela 5.1 apresenta os resultados da métrica F1 *score* para os algoritmos testados em conformidade com a primeira abordagem. Os resultados obtidos para a Floresta Aleatória e para os modelos de *One Class* - SVM estão em conformidade com os obtidos por Vargas (2019), o que deveria acontecer, uma vez que houve poucas modificações no código. Além disso, o resultado obtido para o *Local Outlier Factor* está bastante próximo do encontrado por Junior (2022). Entretanto, o valor de F1 *score* do Envelope Elíptico se distanciou em 0,15 do estimado por Junior (2022). Essas diferenças ocorreram, pois o autor realizou uma otimização de hiperparâmetros com diversos parâmetros, o que não ocorreu neste trabalho, uma vez que o autor não especificou os hiperparâmetros finais.

Algoritmo	F1 *score*
Local Outlier Factor	0,83
Floresta de isolamento	0,72
One Class SVM - RBF	0,53
Envelope Elíptico	0,50
One Class SVM - SIGMOID	0,48
One Class SVM - POLY	0,42
One Class SVM - LINEAR	0,40

Tabela 5.1: Resultados dos algoritmos de acordo com a primeira abordagem do capítulo 4.

A tabela 5.2 exibe os resultados gerais da segunda abordagem desenvolvida para os algoritmos não supervisionados. É válido comparar com os valores da tabela 5.1 com a coluna F1 *micro*. Nesse sentido, a floresta de isolamento obteve uma performance inferior, enquanto os algoritmos de *One Class* - SVM tiveram performances melhores. Além disso, o *Local Outlier Factor* se manteve com o melhor F1, e o Envelope Elíptico teve um resultado aproximado.

Ainda com relação a tabela 5.2, o F1 foi computado para cada classe, e é possível ver

que o Envelope Elíptico, no geral, tem bastante dificuldade de distinguir as anomalias. A Floresta de Isolamento possui um valor inferior para essa métrica. Além disso, o tempo de execução de cada um dos algoritmos é exibido. Apenas o *One Class* - SVM com *kernel* linear necessitou de um tempo maior, mas não é um valor exorbitante. Vale ressaltar que esse tempo depende da capacidade computacional disponível.

Algoritmo	F1 *micro*	F1 *macro*	F1 normal	F1 anomalia	Tempo de execução
Local Outlier Factor	0,93	0,93	0,91	0,95	17 segundos
One Class SVM - RBF	0,73	0,68	0,56	0,81	4 minutos
One Class SVM - LINEAR	0,67	0,64	0,53	0,74	1 hora e 28 minutos
One Class SVM - POLY	0,58	0,56	0,45	0,66	2 minutos
One Class SVM - SIGMOID	0,55	0,54	0,45	0,62	5 minutos
Floresta de Isolamento	0,52	0,49	0,60	0,39	17 segundos
Envelope Elíptico	0,38	0,29	0,54	0,03	17 segundos

Tabela 5.2: Resultados dos algoritmos não supervisionados de acordo com a segunda abordagem do capítulo 4.

A partir desse ponto, são apresentados os resultados dos algoritmos para cada tipo de evento separadamente. As tabelas 5.3, 5.4, 5.5, 5.6, 5.7, 5.8, 5.9 e 5.10 exibem as quatro métricas de F1 *score* para os eventos um, dois, três, quatro, cinco, seis, sete e oito, respectivamente.

Algoritmo	F1 *micro*	F1 *macro*	F1 normal	F1 anomalia
One Class SVM - POLY	0,40	0,36	0,23	0,50
Local Outlier Factor	0,32	0,24	0,00	0,49
One Class SVM - RBF	0,30	0,26	0,08	0,44
One Class SVM - LINEAR	0,49	0,42	0,63	0,20
One Class SVM - SIGMOID	0,38	0,33	0,52	0,14
Floresta de Isolamento	0,65	0,40	0,79	0,00
Envelope Eliptico	0,65	0,40	0,79	0,00

Tabela 5.3: Resultados da avaliação dos algoritmos treinados conforme a segunda abordagem do capítulo 4 em dados do evento um.

Algoritmo	F1 *micro*	F1 *macro*	F1 normal	F1 anomalia
One Class SVM - LINEAR	0,78	0,78	0,77	0,78
One Class SVM - SIGMOID	0,67	0,67	0,64	0,70
One Class SVM - RBF	0,42	0,33	0,08	0,58
Local Outlier Factor	0,40	0,29	0,00	0,57
One Class SVM - POLY	0,10	0,10	0,15	0,05
Floresta de Isolamento	0,59	0,38	0,74	0,03
Envelope Elíptico	0,58	0,37	0,73	0,00

Tabela 5.4: Resultados da avaliação dos algoritmos treinados conforme a segunda abordagem do capítulo 4 em dados do evento dois.

Algoritmo	F1 *micro*	F1 *macro*	F1 normal	F1 anomalia
One Class SVM - SIGMOID	0,87	0,77	0,61	0,92
One Class SVM - LINEAR	0,81	0,73	0,59	0,88
Local Outlier Factor	0,78	0,44	0,00	0,87
One Class SVM - RBF	0,77	0,47	0,07	0,87
Floresta de Isolamento	0,74	0,70	0,61	0,80
One Class SVM - POLY	0,41	0,33	0,09	0,57
Envelope Elíptico	0,21	0,17	0,34	0,01

Tabela 5.5: Resultados da avaliação dos algoritmos treinados conforme a segunda abordagem do capítulo 4 em dados do evento três.

Algoritmo	F1 *micro*	F1 *macro*	F1 normal	F1 anomalia
Local Outlier Factor	0,87	0,46	0,00	0,93
One Class SVM - RBF	0,79	0,45	0,02	0,88
One Class SVM - LINEAR	0,70	0,51	0,20	0,82
One Class SVM - POLY	0,63	0,41	0,04	0,77
One Class SVM - SIGMOID	0,42	0,33	0,09	0,57
Floresta de Isolamento	0,23	0,22	0,13	0,31
Envelope Elíptico	0,06	0,06	0,11	0,00

Tabela 5.6: Resultados da avaliação dos algoritmos treinados conforme a segunda abordagem do capítulo 4 em dados do evento quatro.

Algoritmo	F1 *micro*	F1 *macro*	F1 normal	F1 anomalia
One Class SVM - LINEAR	0,88	0,84	0,77	0,92
One Class SVM - SIGMOID	0,79	0,72	0,59	0,85
One Class SVM - POLY	0,70	0,52	0,22	0,82
One Class SVM - RBF	0,69	0,45	0,08	0,81
Local Outlier Factor	0,68	0,41	0,00	0,81
Floresta de Isolamento	0,32	0,25	0,48	0,03
Envelope Elíptico	0,31	0,24	0,47	0,00

Tabela 5.7: Resultados da avaliação dos algoritmos treinados conforme a segunda abordagem do capítulo 4 em dados do evento cinco.

Algoritmo	F1 *micro*	F1 *macro*	F1 normal	F1 anomalia
One Class SVM - SIGMOID	0,53	0,47	0,65	0,29
Local Outlier Factor	0,11	0,10	0,00	0,19
One Class SVM - POLY	0,20	0,20	0,23	0,18
One Class SVM - RBF	0,13	0,13	0,08	0,18
One Class SVM - LINEAR	0,58	0,39	0,73	0,06
Floresta de Isolamento	0,86	0,46	0,93	0,00
Envelope Elíptico	0,86	0,46	0,93	0,00

Tabela 5.8: Resultados da avaliação dos algoritmos treinados conforme a segunda abordagem do capítulo 4 em dados do evento seis.

Algoritmo	F1 *micro*	F1 *macro*	F1 normal	F1 anomalia
One Class SVM - SIGMOID	0,65	0,65	0,65	0,64
One Class SVM - RBF	0,34	0,28	0,08	0,48
Local Outlier Factor	0,31	0,24	0,00	0,48
One Class SVM - LINEAR	0,58	0,54	0,67	0,40
One Class SVM - POLY	0,31	0,30	0,21	0,39
Floresta de Isolamento	0,66	0,40	0,80	0,00
Envelope Elíptico	0,66	0,40	0,80	0,00

Tabela 5.9: Resultados da avaliação dos algoritmos treinados conforme a segunda abordagem do capítulo 4 em dados do evento sete.

Algoritmo	F1 *micro*	F1 *macro*	F1 normal	F1 anomalia
Envelope Elíptico	0,85	0,82	0,89	0,75
One Class SVM - LINEAR	0,75	0,75	0,77	0,73
One Class SVM - SIGMOID	0,66	0,66	0,65	0,67
Floresta de Isolamento	0,78	0,71	0,85	0,58
One Class SVM - RBF	0,37	0,30	0,08	0,52
Local Outlier Factor	0,35	0,26	0,00	0,51
One Class SVM - POLY	0,09	0,08	0,16	0,00

Tabela 5.10: Resultados da avaliação dos algoritmos treinados conforme a segunda abordagem do capítulo 4 em dados do evento oito.

A fim entender e comparar melhor os resultados de cada tipo de anomalia, a principal métrica será o F1 *score* para as anomalias, isto é, qual bem o algoritmo performa para esse evento, e, em segundo lugar, o F1 *score macro*.

Nesse cenário, devido heterogeneidade da dinâmica de cada evento, diferentes algoritmos performam melhor para cada um deles. Dessa forma, o evento seis foi o que apresentou maior dificuldade para os algoritmos detectarem. Além disso, o Envelope Elíptico, em geral, não performa bem na detecção das anomalias, ele detecta melhor classes classificadas como normais. Entretanto, há uma exceção para o evento oito, no qual o Envelope Elíptico detém a melhor performance. Ademais, os algoritmos detectam bem o evento três, mas têm problema para distinguir amostras normais. Em seguida, são destacados os melhores modelos para cada evento, e as métricas são comparadas com o *benchmark* de Allwright (2022) apresentado no capítulo 3:

- Evento um: *One Class* SVM - POLY;
  - F1 *score* anomalia: No limite do Ok;
  - F1 *score macro*: Ruim.
- Evento dois: *One Class* SVM - LINEAR;
  - F1 *score* anomalia: Ok, próximo de bom;

  - F1 *score macro*: Ok, próximo de bom.

- Evento três: *One Class* SVM - SIGMOID;
  - F1 *score* anomalia: Muito bom;
  - F1 *score macro*: Ok, próximo de bom.
- Evento quatro: *One Class* SVM - LINEAR;
  - F1 *score* anomalia: Bom;
  - F1 *score macro*: No limite do Ok.
- Evento cinco: *One Class* SVM - LINEAR;
  - F1 *score* anomalia: Muito bom;
  - F1 *score macro*: Bom;
- Evento seis: *One Class* SVM - SIGMOID;
  - F1 *score* anomalia: Ruim;
  - F1 *score macro*: Ruim;
- Evento sete: *One Class* SVM - SIGMOID;
  - F1 *score* anomalia: Ok;
  - F1 *score macro*: Ok;
- Evento oito: Envelope Elíptico;
  - F1 *score* anomalia: Ok, próximo de bom;
  - F1 *score macro*: Bom.

As tabelas 5.11 e 5.12 expõem as métricas encontradas nos dados de validação, isto é, instâncias desenhadas. Os números mostram que, nos dois cenários, os algoritmos ou entendem bem amostras normais ou entendem bem amostras de anomalia, não há um balanceamento. Entretanto, no evento sete, há algoritmos com F1 *score* anomalia muito bom, e, no evento um, há F1 *score* anomalia Ok de acordo com a escala de Allwright (2022).

Algoritmo	F1 *micro*	F1 *macro*	F1 normal	F1 anomalia
One Class SVM - RBF	0,57	0,36	0,00	0,72
Local Outlier Factor	0,57	0,36	0,00	0,72
One Class SVM - LINEAR	0,32	0,31	0,23	0,39
One Class SVM - POLY	0,48	0,44	0,59	0,30
One Class SVM - SIGMOID	0,29	0,29	0,30	0,28
Floresta de Isolamento	0,43	0,30	0,61	0,00
Envelope Elíptico	0,43	0,30	0,61	0,00

Tabela 5.11: Resultados da avaliação dos algoritmos treinados conforme a segunda abordagem do capítulo 4 em dados desenhados do evento um.

Algoritmo	F1 *micro*	F1 *macro*	F1 normal	F1 anomalia
One Class SVM - RBF	0,86	0,46	0,00	0,92
Local Outlier Factor	0,86	0,46	0,00	0,92
One Class SVM - LINEAR	0,78	0,55	0,23	0,87
One Class SVM - SIGMOID	0,72	0,55	0,26	0,83
One Class SVM - POLY	0,26	0,26	0,24	0,27
Floresta de Isolamento	0,14	0,12	0,25	0,00
Envelope Elíptico	0,14	0,12	0,25	0,00

Tabela 5.12: Resultados da avaliação dos algoritmos treinados conforme a segunda abordagem do capítulo 4 em dados desenhados do evento sete.

5.2 Algoritmos Supervisionados

A tabela 5.13 salienta os resultados obtidos com os modelos de aprendizado supervisionado. Em geral, todos os F1 *score* são bons ou muito bons. Porém, esses resultados podem estar sobreajustados aos dados. Por isso, é realizada a validação dos modelos nas instâncias *benchmark*, instâncias desenhadas. Portanto, a tabela 5.14 exibe os valores encontrados para as amostras de validação, os quais são menores, ou seja, é um indicativo de sobreajuste aos dados de treinamento. Ainda assim, os modelos performam bem na identificação das anomalias.

Além disso, os modelos ajustado possuem um hiperparâmetro a mais para tentar controlar o sobreajuste. Ao comparar os resultados dos modelos sem ajuste e com ajuste das tabelas 5.13 e 5.14, é possível deduzir que o parâmetro reduziu as métricas da Floresta Aleatória nos dados de teste, mas os valores permaneceram os mesmos nos dados de validação. Desse modo, há um indício de que o sobreajuste desse modelo não é tão alto. Entretanto, ao olhar os resultados do XGBoost, percebe-se que as métricas reduzem pouco no conjunto de teste. Por outro lado, no conjunto de validação, as métricas são mais elevadas, ou seja, há uma redução do sobreajuste do modelo sem ajustes, o que viabiliza uma melhor generalização para detectar anomalias.

Algoritmo	F1 *micro*	F1 *macro*	F1 normal	F1 anomalia
Floresta Aleatória	0,99	0,99	0,99	0,99
Floresta Aleatória ajustada	0,90	0,86	0,93	0,80
XGBoost	0,98	0,98	0,99	0,97
XGBoost ajustado	0,97	0,96	0,98	0,95

Tabela 5.13: Resultados da avaliação dos algoritmos supervisionados treinados conforme a segunda abordagem do capítulo 4 nos dados de teste. Os modelos ajustados foram ajustados pelo hiperparâmetro *max_depth*.

Algoritmo	F1 *micro*	F1 *macro*	F1 normal	F1 anomalia
Floresta Aleatória	0,67	0,49	0,18	0,80
Floresta Aleatória ajustada	0,67	0,49	0,19	0,80
XGBoost	0,60	0,47	0,20	0,73
XGBoost ajustado	0,67	0,49	0,18	0,80

Tabela 5.14: Resultados da avaliação dos algoritmos supervisionados treinados conforme a segunda abordagem do capítulo 4 nos dados de validação. Os modelos ajustados foram ajustados pelo hiperparâmetro *max_depth*.

Para fins de comparação com os algoritmos não supervisionados, as tabelas 5.15 e 5.16 apresentam os resultados dos modelos supervisionados nas instâncias de validação separadas por evento. Desse modo, mesmo com uma performance não tão boa, no evento um, os algoritmos não supervisionados vão melhor. Em contrapartida, no evento sete, os algoritmos supervisionados detém resultados melhores na detecção das amostras com anomalias, mas o F1 *score macro* dos algoritmos não supervisionados é melhor.

Algoritmo	F1 *micro*	F1 *macro*	F1 normal	F1 anomalia
Floresta Aleatória	0,21	0,21	0,24	0,17
Floresta Aleatória ajustada	0,21	0,21	0,24	0,17
XGBoost	0,22	0,20	0,32	0,09
XGBoost ajustado	0,18	0,17	0,25	0,09

Tabela 5.15: Resultados da avaliação dos algoritmos supervisionados treinados conforme a segunda abordagem do capítulo 4 nos dados de validação do evento um. Os modelos ajustados foram ajustados pelo hiperparâmetro *max_depth*.

Algoritmo	F1 *micro*	F1 *macro*	F1 normal	F1 anomalia
Floresta Aleatória	0,77	0,57	0,27	0,86
Floresta Aleatória ajustada	0,78	0,58	0,29	0,87
XGBoost	0,70	0,55	0,28	0,81
XGBoost ajustado	0,77	0,57	0,27	0,87

Tabela 5.16: Resultados da avaliação dos algoritmos supervisionados treinados conforme a segunda abordagem do capítulo 4 nos dados de validação do evento sete. Os modelos ajustados foram ajustados pelo hiperparâmetro *max_depth*.

Uma vantagem dos algoritmos supervisionados é a possibilidade de quantificar quanto cada variável contribui para a previsão final. Desse modo, as Figs. 5.1 e 5.2 apresentam a medida de importância relativa das variáveis para a Floresta Aleatória e para o XGBoost, respectivamente. Os gráficos mostram que a variável que mais contribui para a detecção de anomalias na Floresta Aleatória, T-TPT__maximum, possui aproximadamente 12% da importância total, e que a variável que mais contribui para a detecção de anomalias no XGBoost, T-JUS-CKP__median, detém aproximadamente 18% da importância total. Além disso, algumas variáveis são encontradas nos dois gráficos. A seguir são mostradas as variáveis que resultaram em importância zero em cada um dos modelos, isto é, não agregam em nada.

- Floresta Aleatória;
  - T-JUS-CKP__variance;
  - T-JUS-CKP__standard_deviation;
  - P-PDG__mean;
  - P-PDG__minimum;
  - P-PDG__maximum;
  - P-PDG__root_mean_square;
  - P-PDG__variance;
  - P-PDG__standard_deviation.
- XGBoost;
  - T-JUS-CKP__mean;
  - P-PDG__maximum;
  - QGL__maximum;
  - QGL__root_mean_square;
  - QGL__variance;
  - QGL__standard_deviation;
  - P-PDG__standard_deviation;
  - P-PDG__variance;
  - P-PDG__root_mean_square;
  - P-PDG__minimum;
  - T-JUS-CKP__variance;
  - P-JUS-CKGL__variance;
  - P-TPT__variance;
  - P-TPT__root_mean_square;
  - T-TPT__variance;
  - T-JUS-CKP__minimum;
  - T-TP__root_mean_square;
  - P-MON-CKP__mean;
  - P-MON-CKP__variance;

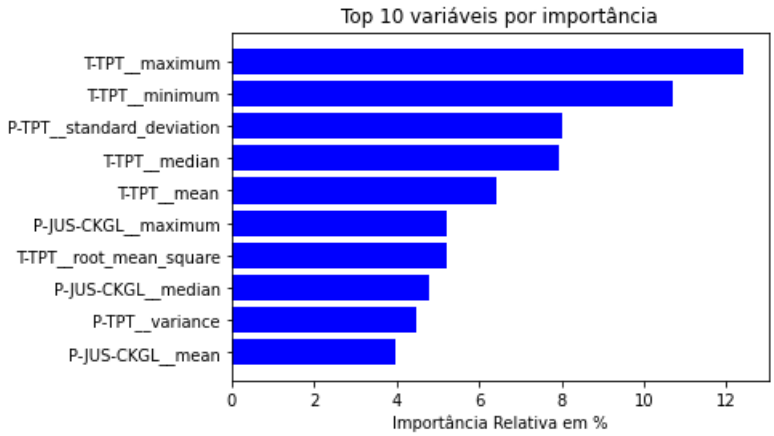

Figura 5.1: Top dez variáveis em importância relativa calculada pelo algoritmo Floresta Aleatória.

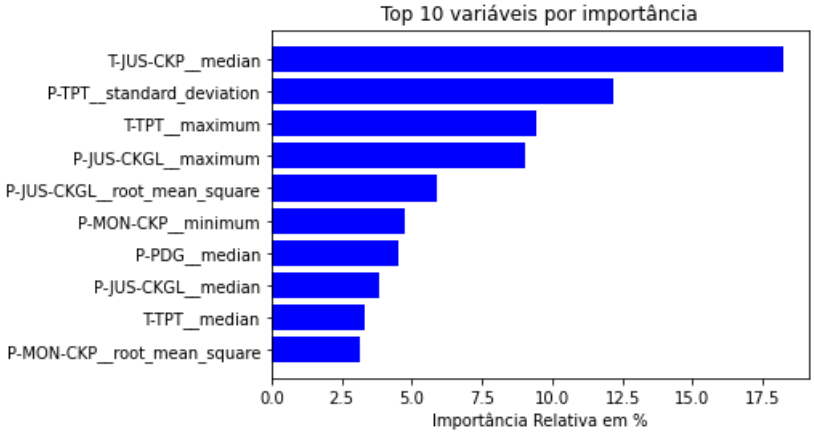

Figura 5.2: Top dez variáveis em importância relativa calculada pelo algoritmo XGBoost.

5.3 Otimização dos resultados

5.3.1 Janelas diferentes e modelos customizados

A quantidade de observações em cada janela escolhida para testar os modelos é apresenta na tabela 5.17. Nas tabelas 5.18 à 5.33, é exibida a quantidade de pontos de cada conjunto de dados extraídos de acordo com as janelas determinadas.

Evento	Janela 1	Janela 2	Janela 3	Janela 4	Janela 5	Janela padrão
1	7200 s	3600 s	1800 s	900 s	600 s	180 s
2	1200 s	900 s	600 s	300 s	-	180 s
3	16200 s	14400 s	10800 s	7200 s	3600 s	180 s
4	900 s	720 s	600 s	420 s	300 s	180 s
5	16200 s	10800 s	7200 s	3600 s	1800 s	180 s
6	900 s	720 s	600 s	420 s	300 s	180 s
7	16200 s	14400 s	7200 s	3600 s	-	180 s
8	7200 s	3600 s	1800 s	900 s	-	180 s

Tabela 5.17: Janelas escolhidas para treinar os modelos.

A fim de facilitar compreensão das tabelas 5.18 à 5.33, segue legenda.

- # pontos treino: quantidade de pontos total na amostra de treino;
- # pontos eventos treino: quantidade de pontos classificados como anomalia na amostra de treino;
- # pontos teste: quantidade de pontos total na amostra de teste;
- # pontos evento teste: quantidade de pontos classificados como anomalia na amostra de teste.

Janela	# pontos treino	# pontos teste	# pontos eventos teste
7200 s	320	80	8
3600 s	720	180	18
1800 s	1680	420	42
900 s	3520	880	88
600 s	5360	1340	134
180 s	18400	4600	460

Tabela 5.18: Quantidade de pontos do evento 1 na abordagem não supervisionada.

Janela	# pontos treino	# pontos eventos treino	# pontos teste	# pontos eventos teste
7200 s	63	6	15	1
3600 s	143	14	35	3
1800 s	335	33	83	3
900 s	706	70	175	17
600 s	1071	107	267	26
180 s	3680	368	920	92

Tabela 5.19: Quantidade de pontos do evento 1 na abordagem supervisionada.

Janela	# pontos treino	# pontos teste	# pontos eventos teste
1200 s	2880	720	72
900 s	4000	1000	100
600 s	6360	1590	159
300 s	13280	3320	332
180 s	22800	5700	570

Tabela 5.20: Quantidade de pontos do evento 2 na abordagem não supervisionada.

Janela	# pontos treino	# pontos eventos treino	# pontos teste	# pontos eventos teste
1200 s	575	57	143	14
900 s	800	80	200	20
600 s	1271	127	317	31
300 s	2655	265	663	66
180 s	4560	456	1140	114

Tabela 5.21: Quantidade de pontos do evento 2 na abordagem supervisionada.

Janela	# pontos treino	# pontos teste	# pontos eventos teste
16200 s	416	103	10
14400 s	416	103	10
10800 s	418	104	10
7200 s	880	220	22
3600 s	1788	446	44
180 s	44890	11222	1122

Tabela 5.22: Quantidade de pontos do evento 3 na abordagem não supervisionada.

Janela	# pontos treino	# pontos eventos treino	# pontos teste	# pontos eventos teste
16200 s	247	24	61	6
14400 s	247	24	61	6
10800 s	255	25	63	6
7200 s	503	50	125	12
3600 s	1012	101	253	25
180 s	25135	2513	6283	628

Tabela 5.23: Quantidade de pontos do evento 3 na abordagem supervisionada.

Janela	# pontos treino	# pontos teste	# pontos eventos teste
900 s	8616	2153	215
720 s	10892	2722	272
600 s	13166	3291	329
420 s	19073	4767	476
300 s	26770	6692	669
180 s	44890	11222	1122

Tabela 5.24: Quantidade de pontos do evento 4 na abordagem não supervisionada.

Janela	# pontos treino	# pontos eventos treino	# pontos teste	# pontos eventos teste
900 s	9382	938	2345	234
720 s	11860	1186	2964	296
600 s	14336	1433	3583	358
420 s	20768	2076	5192	519
300 s	29150	2915	7286	728
180 s	48880	4888	12220	1222

Tabela 5.25: Quantidade de pontos do evento 4 na abordagem supervisionada.

Janela	# pontos treino	# pontos teste	# pontos eventos teste
16200 s	416	103	10
10800 s	418	104	10
7200 s	880	220	22
3600 s	1792	447	44
1800 s	4074	1017	101
180 s	45032	11257	1125

Tabela 5.26: Quantidade de pontos do evento 5 na abordagem não supervisionada.

Janela	# pontos treino	# pontos eventos treino	# pontos teste	# pontos eventos teste
16200 s	120	12	30	3
10800 s	191	19	47	4
7200 s	320	32	80	8
3600 s	655	65	163	16
1800 s	1391	139	347	34
180 s	14535	1453	3633	363

Tabela 5.27: Quantidade de pontos do evento 5 na abordagem supervisionada.

Janela	# pontos treino	# pontos teste	# pontos eventos teste
900 s	600	150	15
720 s	840	210	21
600 s	1160	290	29
420 s	1560	390	39
300 s	2360	590	59
180 s	4040	1010	101

Tabela 5.28: Quantidade de pontos do evento 6 na abordagem não supervisionada.

Janela	# pontos treino	# pontos eventos treino	# pontos teste	# pontos eventos teste
900 s	120	12	30	3
720 s	167	16	41	4
600 s	231	23	57	5
420 s	311	31	77	7
300 s	471	47	117	11
180 s	807	80	201	20

Tabela 5.29: Quantidade de pontos do evento 6 na abordagem supervisionada.

Janela	# pontos treino	# pontos teste	# pontos eventos teste
16200 s	416	103	10
14400 s	416	103	10
7200 s	880	220	22
3600 s	1790	446	44
180 s	44946	11235	1123

Tabela 5.30: Quantidade de pontos do evento 7 na abordagem não supervisionada.

Janela	# pontos treino	# pontos eventos treino	# pontos teste	# pontos eventos teste
16200 s	103	10	25	2
14400 s	127	12	31	3
7200 s	271	27	67	6
3600 s	551	55	137	13
180 s	11471	1147	2867	286

Tabela 5.31: Quantidade de pontos do evento 7 na abordagem supervisionada.

Janela	# pontos treino	# pontos teste	# pontos eventos teste
7200 s	360	90	9
3600 s	800	200	20
1800 s	1680	420	42
900 s	3520	880	88
180 s	18160	4540	454

Tabela 5.32: Quantidade de pontos do evento 8 na abordagem não supervisionada.

Janela	# pontos treino	# pontos eventos treino	# pontos teste	# pontos eventos teste
7200 s	71	7	17	1
3600 s	160	16	40	4
1800 s	335	33	83	8
900 s	703	70	175	17
180 s	3631	363	907	90

Tabela 5.33: Quantidade de pontos do evento 8 na abordagem supervisionada.

As tabelas 5.34 à 5.41 exibem os resultados da melhor janela e da janela de referência de 180 s para cada algoritmo não supervisionado. Diante dos dados expostos, percebe-se que, para alguns algoritmos e eventos, uma quantidade maior de pontos otimiza a performance do modelo. Entretanto, esse fenômeno não é consenso para todas as soluções. Além disso, não há consenso entre as janelas para um mesmo evento, por exemplo, no evento 1 há algoritmos que performam melhor com o intervalo de 7200 segundos e algoritmos que performam melhor com os intervalos de 600 segundos. Ainda com relação aos resultados, a tabela 5.42 apresenta as duas melhores combinações entre janela e algoritmo para cada evento. Exceto para o evento o 8, o algoritmo *Local Outlier Factor* e o intervalo padrão de 180 segundos.

Dessa forma, o caminho de aumentar a dimensão da janela para aproxima-lá da usada pelos especialistas para confirmar o acontecimento de um evento indesejável parece natural para otimizar a performance dos modelos, uma vez que imagina-se que os pontos extraídos carregaram uma informação que traduz o fenômeno físico. Entretanto, os resultados não estão de acordo com isso, um dos principais motivos que levam a essa discordância é a quantidade de observações por instância, o que acarreta em uma quantidade reduzida de pontos para ajuste do modelo conforme é mostrado nas tabelas 5.18 à 5.33. Isso se apresenta como uma limitação intrínseca aos dados disponibilizados. Assim, a janela de 180 segundos viabiliza uma quantidade de pontos maior para que a solução generalize melhor para dados novos.

Nesse sentido, devido ao algoritmo *Local Outlier Factor* e à janela de 180 segundos performar melhor para os eventos de 1 a 7, essa combinação foi escolhida para passar para a próxima etapa. Apesar de, no evento 8, essa não ser a melhor solução, ainda é uma boa solução, com F1 normal de 0,98 e com F1 anomalia de 0,85.

Algoritmo	Janela (s)	F1 *micro*	F1 *macro*	F1 normal	F1 anomalia
Envelope Elíptico	7200	0,91	0,59	0,95	0,22
Envelope Elíptico	180	0,9	0,57	0,95	0,2
Floresta de Isolamento	7200	0,9	0,47	0,95	0
Floresta de Isolamento	180	0,9	0,47	0,95	0
Local Outlier Factor	180	0,94	0,86	0,96	0,76
Local Outlier Factor	600	0,9	0,81	0,94	0,68
One Class SVM - LINEAR	600	0,57	0,47	0,7	0,24
One Class SVM - LINEAR	180	0,54	0,37	0,7	0,05
One Class SVM - POLY	7200	0,62	0,54	0,74	0,35
One Class SVM - POLY	180	0,52	0,46	0,65	0,27
One Class SVM - RBF	7200	0,54	0,48	0,65	0,3
One Class SVM - RBF	180	0,52	0,45	0,64	0,26
One Class SVM - SIGMOID	900	0,48	0,36	0,63	0,09
One Class SVM - SIGMOID	180	0,47	0,34	0,63	0,05

Tabela 5.34: Resultados da melhor janela e da janela padrão por algoritmo não supervisionado para o evento 1.

Algoritmo	Janela (s)	F1 *micro*	F1 *macro*	F1 normal	F1 anomalia
Envelope Elíptico	300	0,9	0,48	0,95	0,01
Envelope Elíptico	180	0,9	0,47	0,95	0
Floresta de Isolamento	900	0,91	0,6	0,95	0,25
Floresta de Isolamento	180	0,9	0,47	0,95	0
Local Outlier Factor	180	0,93	0,85	0,96	0,74
Local Outlier Factor	300	0,92	0,83	0,95	0,71
One Class SVM - LINEAR	180	0,36	0,33	0,47	0,19
One Class SVM - LINEAR	900	0,47	0,37	0,62	0,12
One Class SVM - POLY	1200	0,46	0,34	0,62	0,06
One Class SVM - POLY	180	0,44	0,32	0,61	0,02
One Class SVM - RBF	1200	0,53	0,48	0,65	0,3
One Class SVM - RBF	180	0,52	0,46	0,64	0,29
One Class SVM - SIGMOID	180	0,57	0,5	0,69	0,31
One Class SVM - SIGMOID	1200	0,56	0,49	0,68	0,31

Tabela 5.35: Resultados da melhor janela e da janela padrão por algoritmo não supervisionado para o evento 2.

Algoritmo	Janela (s)	F1 *micro*	F1 *macro*	F1 normal	F1 anomalia
Envelope Elíptico	16200	0,91	0,57	0,95	0,18
Envelope Elíptico	180	0,9	0,48	0,95	0,01
Floresta de Isolamento	180	0,93	0,69	0,96	0,42
Floresta de Isolamento	16200	0,91	0,57	0,95	0,18
Local Outlier Factor	180	0,98	0,95	0,99	0,91
Local Outlier Factor	3600	0,9	0,8	0,94	0,67
One Class SVM - LINEAR	16200	0,51	0,46	0,63	0,29
One Class SVM - LINEAR	180	0,3	0,23	0,46	0,01
One Class SVM - POLY	7200	0,57	0,5	0,69	0,31
One Class SVM - POLY	180	0,55	0,48	0,66	0,31
One Class SVM - RBF	3600	0,59	0,51	0,7	0,32
One Class SVM - RBF	180	0,55	0,48	0,67	0,3
One Class SVM - SIGMOID	16200	0,44	0,33	0,6	0,06
One Class SVM - SIGMOID	180	0,45	0,31	0,62	0

Tabela 5.36: Resultados da melhor janela e da janela padrão por algoritmo não supervisionado para o evento 3.

Algoritmo	Janela (s)	F1 *micro*	F1 *macro*	F1 normal	F1 anomalia
Envelope Elíptico	420	0,9	0,48	0,95	0
Envelope Elíptico	180	0,9	0,47	0,95	0
Floresta de Isolamento	420	0,9	0,48	0,95	0
Floresta de Isolamento	180	0,9	0,47	0,95	0
Local Outlier Factor	180	0,97	0,92	0,98	0,87
Local Outlier Factor	420	0,97	0,92	0,98	0,86
One Class SVM - LINEAR	300	0,48	0,41	0,61	0,21
One Class SVM - LINEAR	180	0,35	0,32	0,48	0,15
One Class SVM - POLY	300	0,5	0,41	0,64	0,17
One Class SVM - POLY	180	0,5	0,4	0,64	0,16
One Class SVM - RBF	300	0,54	0,47	0,66	0,28
One Class SVM - RBF	180	0,53	0,46	0,66	0,26
One Class SVM - SIGMOID	720	0,48	0,37	0,63	0,1
One Class SVM - SIGMOID	180	0,48	0,37	0,63	0,1

Tabela 5.37: Resultados da melhor janela e da janela padrão por algoritmo não supervisionado para o evento 4.

Algoritmo	Janela (s)	F1 *micro*	F1 *macro*	F1 normal	F1 anomalia
Envelope Elíptico	16200	0,91	0,57	0,95	0,18
Envelope Elíptico	180	0,9	0,47	0,95	0
Floresta de Isolamento	180	0,9	0,48	0,95	0,01
Floresta de Isolamento	16200	0,9	0,47	0,95	0
Local Outlier Factor	180	0,97	0,92	0,98	0,85
Local Outlier Factor	1800	0,92	0,83	0,95	0,71
One Class SVM - LINEAR	10800	0,49	0,44	0,61	0,27
One Class SVM - LINEAR	180	0,61	0,41	0,75	0,06
One Class SVM - POLY	1800	0,55	0,48	0,67	0,3
One Class SVM - POLY	180	0,54	0,48	0,66	0,3
One Class SVM - RBF	3600	0,58	0,51	0,7	0,32
One Class SVM - RBF	180	0,54	0,48	0,66	0,3
One Class SVM - SIGMOID	7200	0,52	0,42	0,66	0,19
One Class SVM - SIGMOID	180	0,49	0,4	0,64	0,16

Tabela 5.38: Resultados da melhor janela e da janela padrão por algoritmo não supervisionado para o evento 5.

Algoritmo	Janela (s)	F1 *micro*	F1 *macro*	F1 normal	F1 anomalia
Envelope Elíptico	900	0,9	0,47	0,95	0
Envelope Elíptico	180	0,9	0,47	0,95	0
Floresta de Isolamento	900	0,9	0,47	0,95	0
Floresta de Isolamento	180	0,9	0,47	0,95	0
Local Outlier Factor	300	0,76	0,65	0,85	0,46
Local Outlier Factor	180	0,77	0,66	0,85	0,46
One Class SVM - LINEAR	600	0,47	0,39	0,61	0,16
One Class SVM - LINEAR	180	0,49	0,34	0,65	0,02
One Class SVM - POLY	300	0,45	0,42	0,56	0,27
One Class SVM - POLY	180	0,45	0,4	0,56	0,25
One Class SVM - RBF	720	0,48	0,44	0,59	0,28
One Class SVM - RBF	180	0,47	0,42	0,59	0,26
One Class SVM - SIGMOID	720	0,48	0,43	0,59	0,27
One Class SVM - SIGMOID	180	0,46	0,42	0,58	0,25

Tabela 5.39: Resultados da melhor janela e da janela padrão por algoritmo não supervisionado para o evento 6.

Algoritmo	Janela (s)	F1 *micro*	F1 *macro*	F1 normal	F1 anomalia
Envelope Elíptico	3600	0,98	0,93	0,99	0,88
Envelope Elíptico	180	0,97	0,9	0,98	0,82
Floresta de Isolamento	3600	0,9	0,5	0,95	0,04
Floresta de Isolamento	180	0,9	0,48	0,95	0,01
Local Outlier Factor	180	0,97	0,94	0,99	0,89
Local Outlier Factor	3600	0,9	0,8	0,94	0,66
One Class SVM - LINEAR	180	0,4	0,32	0,55	0,08
One Class SVM - LINEAR	3600	0,44	0,33	0,6	0,07
One Class SVM - POLY	7200	0,58	0,51	0,69	0,32
One Class SVM - POLY	180	0,53	0,47	0,66	0,28
One Class SVM - RBF	3600	0,58	0,51	0,7	0,32
One Class SVM - RBF	180	0,55	0,48	0,66	0,31
One Class SVM - SIGMOID	7200	0,5	0,37	0,65	0,1
One Class SVM - SIGMOID	180	0,48	0,36	0,63	0,1

Tabela 5.40: Resultados da melhor janela e da janela padrão por algoritmo não supervisionado para o evento 7.

Algoritmo	Janela (s)	F1 *micro*	F1 *macro*	F1 normal	F1 anomalia
Envelope Elíptico	3600	1	0,99	1	0,98
Envelope Elíptico	180	0,97	0,9	0,98	0,82
Floresta de Isolamento	180	0,94	0,77	0,97	0,58
Floresta de Isolamento	1800	0,94	0,75	0,97	0,53
Local Outlier Factor	180	0,97	0,92	0,98	0,85
Local Outlier Factor	900	0,91	0,82	0,95	0,7
One Class SVM - LINEAR	1800	0,45	0,41	0,56	0,27
One Class SVM - LINEAR	180	0,38	0,28	0,56	0
One Class SVM - POLY	7200	0,5	0,41	0,64	0,18
One Class SVM - POLY	180	0,45	0,31	0,62	0
One Class SVM - RBF	180	0,55	0,49	0,67	0,31
One Class SVM - RBF	900	0,53	0,47	0,65	0,3
One Class SVM - SIGMOID	7200	0,52	0,47	0,64	0,3
One Class SVM - SIGMOID	180	0,49	0,39	0,64	0,14

Tabela 5.41: Resultados da melhor janela e da janela padrão por algoritmo não supervisionado para o evento 8.

Algoritmo	Evento	Janela (s)	F1 *micro*	F1 *macro*	F1 normal	F1 anomalia
Local Outlier Factor	1	180	0,94	0,86	0,96	0,76
Local Outlier Factor	1	600	0,9	0,81	0,94	0,68
Local Outlier Factor	2	180	0,93	0,85	0,96	0,74
Local Outlier Factor	2	300	0,92	0,83	0,95	0,71
Local Outlier Factor	3	180	0,98	0,95	0,99	0,91
Local Outlier Factor	3	3600	0,9	0,8	0,94	0,67
Local Outlier Factor	4	180	0,97	0,92	0,98	0,87
Local Outlier Factor	4	420	0,97	0,92	0,98	0,86
Local Outlier Factor	5	180	0,97	0,92	0,98	0,85
Local Outlier Factor	5	1800	0,92	0,83	0,95	0,71
Local Outlier Factor	6	180	0,77	0,66	0,85	0,46
Local Outlier Factor	6	300	0,76	0,65	0,85	0,46
Local Outlier Factor	7	180	0,97	0,94	0,99	0,89
Envelope Elíptico	7	3600	0,98	0,93	0,99	0,88
Envelope Elíptico	8	3600	1	0,99	1	0,98
Envelope Elíptico	8	1800	0,99	0,98	1	0,97

Tabela 5.42: Resultados das duas melhores combinações entre janela e algoritmo por evento.

No caso dos algoritmos supervisionados, a quantidade reduzida de pontos é um problema maior devido à natureza dos modelos que tentam se ajustar não só aos eventos normais como também aos eventos indesejados, o que pode gerar sobreajuste ou subajuste. Como exemplo, a tabela 5.43 apresenta a melhor janela e a janela padrão para o evento 3, o resultado ligeiramente maior da janela de 16200 segundos pode ser devido ao sobreajuste aos dados. Como há mais pontos para o intervalo de 180 segundos conforme mostrado na tabela 5.23, o resultado é mais confiável. Os outros eventos seguem esse padrão. Seguindo essas definições, a

tabela 5.43 apresenta os resultados dos modelos supervisionados customizados por evento. Os indicadores são todos elevados com exceção do evento 6 e do evento 1, o primeiro mostra que há dificuldades de identificar tanto eventos normais quanto eventos indesejados, já o segundo exibe que há dificuldades de classificar eventos normais.

Por outro lado, a tabela 5.45 exibe as métricas para os modelos genéricos e para a janela de 180 segundos tanto nos eventos separadamente como em todos eles em conjunto. Ao comparar com os indicadores do modelo customizado, percebe-se, principalmente pela melhoria nas deficiências para os eventos 1 e 6, que o modelo genérico de Floresta Aleatória é uma solução melhor para o problema. Empregar dados de todos os eventos ao mesmo tempo no treinamento do modelo força o modelo a se adaptar a um conjunto mais diverso de pontos, o que resulta em um modelo com menos sobreajuste, isto é, generaliza melhor para dados novos. Portanto, o modelo genérico de Floresta Aleatória é o escolhido para as etapas seguintes.

Algoritmo	Janela (s)	F1 micro	F1 macro	F1 normal	F1 anomalia
Floresta Aleatória	16200	1	1	1	1
Floresta Aleatória	180	1	1	1	0,99
XGBoost	16200	1	1	1	1
XGBoost	180	1	1	1	0,99

Tabela 5.43: Resultados da melhor janela e da janela padrão por algoritmo supervisionado para o evento 3.

Algoritmo	Evento	F1 micro	F1 macro	F1 normal	F1 anomalia
XGBoost	1	0,9	0,81	0,68	0,94
Floresta Aleatória	1	0,88	0,78	0,62	0,93
XGBoost	2	0,98	0,96	0,92	0,99
Floresta Aleatória	2	0,97	0,92	0,86	0,98
Floresta Aleatória	3	1	1	1	1
XGBoost	3	1	1	1	1
Floresta Aleatória	4	1	1	0,99	1
XGBoost	4	1	0,99	0,98	1
Floresta Aleatória	5	0,95	0,89	0,81	0,97
XGBoost	5	0,95	0,89	0,81	0,97
Floresta Aleatória	6	0,1	0,09	0,18	0
XGBoost	6	0,1	0,09	0,18	0
Floresta Aleatória	7	0,98	0,94	0,89	0,99
XGBoost	7	0,98	0,94	0,89	0,99
Floresta Aleatória	8	0,95	0,88	0,8	0,97
XGBoost	8	0,95	0,88	0,8	0,97

Tabela 5.44: Resultados finais para os modelos supervisionados customizados por evento.

Algoritmo	Evento	F1 micro	F1 macro	F1 normal	F1 anomalia
Floresta Aleatória	1	0,997	0,991	0,983	0,998
XGBoost	1	0,972	0,91	0,837	0,984
Floresta Aleatória	2	0,999	0,998	0,997	1
XGBoost	2	0,99	0,971	0,948	0,994
Floresta Aleatória	3	1	0,999	0,999	1
XGBoost	3	0,998	0,995	0,991	0,999
Floresta Aleatória	4	0,999	0,997	0,994	0,999
XGBoost	4	0,995	0,985	0,973	0,997
Floresta Aleatória	5	0,998	0,994	0,989	0,999
XGBoost	5	0,99	0,973	0,95	0,995
Floresta Aleatória	6	0,974	0,921	0,857	0,986
XGBoost	6	0,911	0,623	0,294	0,953
Floresta Aleatória	7	0,999	0,996	0,993	0,999
XGBoost	7	0,983	0,95	0,91	0,991
Floresta Aleatória	8	0,999	0,996	0,994	0,999
XGBoost	8	0,998	0,995	0,991	0,999
Floresta Aleatória	todos	0,998	0,995	0,991	0,999
XGBoost	todos	0,989	0,969	0,944	0,994

Tabela 5.45: Resultados finais para os modelos supervisionados genéricos por evento.

5.3.2 Seleção de variáveis

A partir dos modelos selecionados na etapa de seleção de janelas, o processo de seleção de variáveis apresentou os resultados das tabelas 5.46 e 5.47 dos modelos não supervisionado e supervisionado, respectivamente. Para o modelo não supervisionado, o melhor conjunto de variáveis foi as top 20 variáveis mais importantes. Já para o modelo supervisionado, o critério que melhor se adaptou foi o de excluir variáveis com menos de 1% de importância, isto é, são excluídas 10 variáveis, restando 31. Os indicadores do modelo não supervisionado são iguais ou marginalmente melhores do que os do modelo treinado com todas as 41 variáveis. Isso mostra que a contribuição das variáveis retiradas não é tão relevante. Assim, reduzir a quantidade de variáveis reduz a dimensão da solução e retira redundâncias de informações, as quais podem acarretar em erros maiores nas classificações. O mesmo ocorreu para o algoritmo supervisionado, vale destacar que para o caso de aplicação, isto é, classificar os eventos de forma geral, nas instâncias simuladas, houve uma melhora de performance de 0,02, o que é interessante para a solução. Por fim, destaca-se que houve conjuntos de variáveis que performaram melhor conforme o evento, o que pode ser uma oportunidade para otimizar ainda mais o modelo.

Algoritmo	Evento	Instancias	Variáveis	F1 micro	F1 macro	F1 normal	F1 anomalia
LOF	1	desenhadas	Top 20 variáveis	0,944	0,787	0,97	0,604
LOF	1	desenhadas	Todas as variáveis	0,942	0,782	0,969	0,596
LOF	7	desenhadas	Top 20 variáveis	0,951	0,923	0,97	0,877
LOF	7	desenhadas	Todas as variáveis	0,95	0,921	0,969	0,873
LOF	todos	desenhadas	Top 20 variáveis	0,953	0,933	0,97	0,896
LOF	todos	desenhadas	Todas as variáveis	0,951	0,931	0,969	0,893
LOF	1	simuladas	Top 20 variáveis	0,914	0,913	0,922	0,904
LOF	1	simuladas	Todas as variáveis	0,913	0,912	0,921	0,903
LOF	2	simuladas	Top 20 variáveis	0,861	0,626	0,922	0,329
LOF	2	simuladas	Todas as variáveis	0,859	0,624	0,921	0,326
LOF	3	simuladas	Top 20 variáveis	0,896	0,881	0,922	0,84
LOF	3	simuladas	Todas as variáveis	0,894	0,88	0,921	0,838
LOF	5	simuladas	Top 20 variáveis	0,932	0,931	0,922	0,939
LOF	5	simuladas	Todas as variáveis	0,931	0,93	0,921	0,939
LOF	6	simuladas	Top 20 variáveis	0,902	0,895	0,922	0,868
LOF	6	simuladas	Todas as variáveis	0,901	0,894	0,921	0,867
LOF	8	simuladas	Top 20 variáveis	0,878	0,818	0,922	0,714
LOF	8	simuladas	Todas as variáveis	0,877	0,817	0,921	0,712
LOF	todos	simuladas	Top 20 variáveis	0,963	0,949	0,922	0,975
LOF	todos	simuladas	Todas as variáveis	0,962	0,948	0,921	0,975

Tabela 5.46: Resultados com e sem seleção do melhor conjunto de variáveis dos modelos não supervisionados. "LOF"se refere ao algoritmo *Local Outlier Factor*.

Algoritmo	Evento	Instancias	Variáveis	F1 micro	F1 macro	F1 normal	F1 anomalia
FA	1	desenhadas	Importância > 1%	0,937	0,52	0,073	0,968
FA	1	desenhadas	Todas as variáveis	0,937	0,517	0,066	0,967
FA	7	desenhadas	Todas as variáveis	0,956	0,922	0,87	0,973
FA	7	desenhadas	Importância > 1%	0,956	0,921	0,87	0,973
FA	todos	desenhadas	Importância > 1%	0,923	0,872	0,791	0,953
FA	todos	desenhadas	Todas as variáveis	0,923	0,872	0,791	0,953
FA	1	simuladas	Importância > 1%	0,927	0,926	0,917	0,935
FA	1	simuladas	Todas as variáveis	0,917	0,915	0,904	0,927
FA	2	simuladas	Todas as variáveis	0,886	0,656	0,374	0,937
FA	2	simuladas	Importância > 1%	0,882	0,651	0,367	0,935
FA	3	simuladas	Importância > 1%	0,696	0,542	0,277	0,807
FA	3	simuladas	Todas as variáveis	0,689	0,521	0,238	0,804
FA	5	simuladas	Importância > 1%	0,821	0,821	0,82	0,823
FA	5	simuladas	Todas as variáveis	0,802	0,802	0,796	0,808
FA	6	simuladas	Todas as variáveis	0,907	0,898	0,869	0,928
FA	6	simuladas	Importância > 1%	0,903	0,895	0,864	0,925
FA	8	simuladas	Todas as variáveis	0,899	0,842	0,748	0,936
FA	8	simuladas	Importância > 1%	0,896	0,84	0,745	0,935
FA	todos	simuladas	Importância > 1%	0,819	0,792	0,867	0,716
FA	todos	simuladas	Todas as variáveis	0,8	0,773	0,851	0,696

Tabela 5.47: Resultados com e sem seleção do melhor conjunto de variáveis dos modelos supervisionados. "FA"se refere ao algoritmo Floresta Aleatória.

5.4 Previsão de dados futuros

A tabela 5.48 exibe os indicadores da LSTM treinada apenas com instâncias sem anomalia medidos nas instâncias sem anomalia. A porcentagem de cada célula da tabela representa a porcentagem de janelas previstas que apresentaram erro percentual inferior a um limiar, por exemplo, 5%. Diante do exposto, o comportamento das variáveis P-PDG, T-TPT e T-JUS-CKP é bem capturado pelo modelo de previsão. Por outro lado, o resultado da variável P-TPT é mediano, e o resultado do restante das variáveis é ruim. O resultado ruim pode estar associado ao fato de os parâmetros utilizados para ajustar o modelo não serem os melhores para a variável específico e à quantidade de janelas empregadas como dado de entrada.

O gráfico da Fig. 5.3 apresenta um exemplo de uma séria prevista da variável T-TPT cujo erro é baixo. É possível ver que, nesse caso, o erro baixo é traduzido em uma série que se assemelha com a série real apesar de a série real variar em uma amplitude maior.

Variável	erro 5%	erro 10%	erro 15%	erro 20%	erro 25%
P_PDG	81,1%	83,9%	84%	92,8%	92,8%
P_TPT	30,1%	35,4%	36,2%	38,9%	38,9%
T_TPT	95,9%	100%	100%	100%	100%
P_MON_CKP	5,6%	13,1%	35,3%	52,6%	61,8%
T_JUS_CKP	65,2%	85,8%	92,2%	100%	100%
P_JUS_CKGL	7,6%	7,6%	7,8%	8,8%	14,3%

Tabela 5.48: Indicadores da LSTM treinada apenas com instâncias sem anomalia avaliados nas instâncias normais.

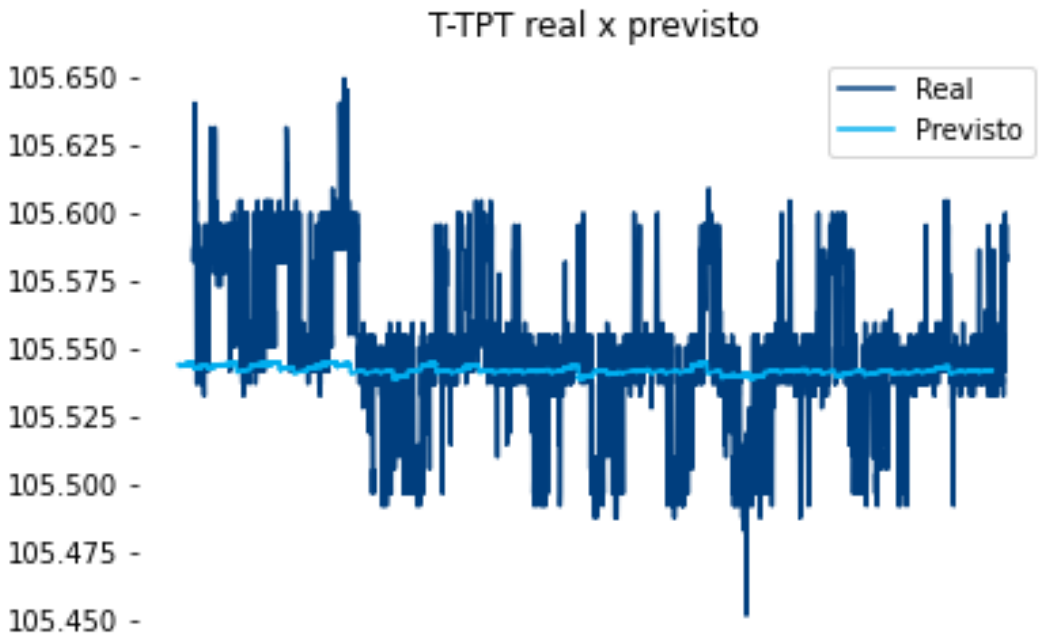

Figura 5.3: Comparação da série real com a série prevista para a variável T-TPT sem anomalia. LSTM treinada apenas com instâncias sem anomalia.

A tabela 5.49 expõe as métricas da LSTM treinada apenas com instâncias sem anomalia apuradas em instâncias com anomalia. A única variável que se destaca um pouco é a P-PDG, para as outras, os valores vão de mediano para ruim em grande parte. A Fig. 5.4 exibe um exemplo de instância do evento 1 da variável P-PDG. Apesar de os valores divergirem bastante em boa parte da série, percebe-se que há uma certa tendência que da série prevista que assemelha-se a da série real. Por outro lado, o gráfico da Fig. 5.5 apresenta uma única janela de 180 segundos prevista e os valores reais da mesma janela do sensor T-JUS-CKP. Essa série é um exemplo de resultado bom numericamente apesar de, visualmente, haver uma distância entre as duas séries.

Variável	erro 5%	erro 10%	erro 15%	erro 20%	erro 25%
P_PDG	2%	37,8%	75,2%	81,1%	82,1%
P_TPT	13,8%	15,8%	31,5%	42,7%	48,8%
T_TPT	30,9%	46,3%	47%	47,4%	56,1%
P_MON_CKP	8,7%	14,8%	21,9%	33,9%	39,8%
T_JUS_CKP	5,1%	15,7%	25,7%	31,8%	34,1%
P_JUS_CKGL	0,2%	0,5%	0,8%	1,1%	8,5%

Tabela 5.49: Indicadores da LSTM treinada apenas com instâncias sem anomalia avaliados nas instâncias com anomalia.

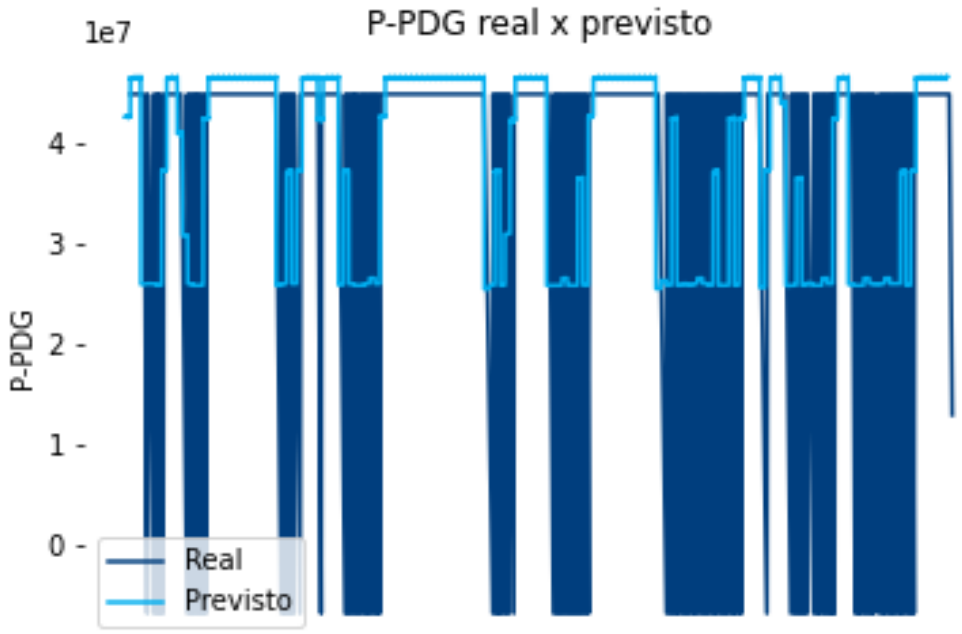

Figura 5.4: Comparação da série real com a série prevista para a variável P-PDG com evento 1. LSTM treinada apenas com instâncias sem anomalia.

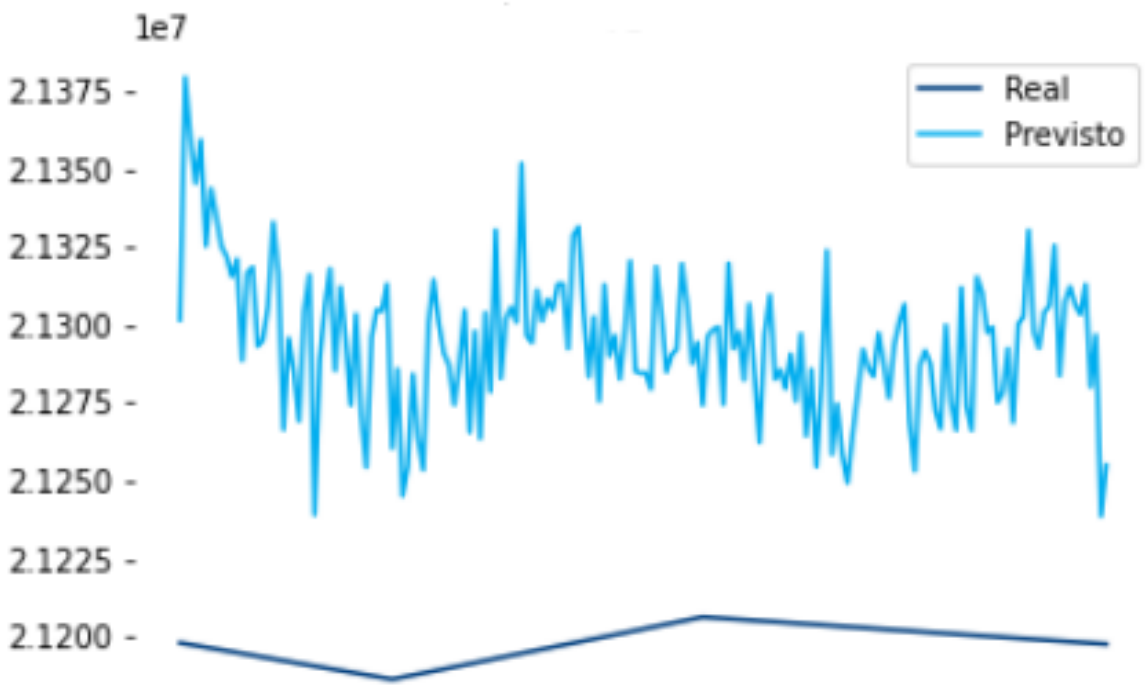

Figura 5.5: Comparação uma janela da série real com uma janela da série prevista para a variável P-PDG com evento 7. LSTM treinada apenas com instâncias sem anomalia.

A tabela 5.50 apresenta os resultados da LSTM treinada com instâncias com e sem anomalia auferidos em instâncias sem anomalia. Apesar de instâncias sem eventos indesejados

serem contempladas no ajuste do modelo, a performance para as variáveis P-PDG, T-TPT e P-JUS-CKGL é prejudicada quando comparada a do modelo treinado apenas com instâncias sem anomalia. Por outro lado, há um certo aprimoramento para as outras variáveis. Destaca-se que o modelo treinado para o sensor P-PDG não convergia durante o treinamento do modelo, e uma investigação do motivo mostrou que era devido ao fato de treinar a rede com instâncias do evento 8, o que mostra uma maior dificuldade de ajuste de um modelo genérico a esse evento para essa variável específica. Dessa forma, o evento 8 foi retirado do treinamento

Variável	erro 5%	erro 10%	erro 15%	erro 20%	erro 25%
P_PDG	0%	0%	0%	0%	0%
P_TPT	30,2%	42,6%	45,7%	48,3%	63,5%
T_TPT	8,9%	16,3%	18,6%	20,3%	22,3%
P_MON_CKP	5,6%	25,4%	50,1%	61%	63,8%
T_JUS_CKP	35,1%	40,6%	41,5%	41,6%	41,9%
P_JUS_CKGL	0%	0%	0%	0%	0%

Tabela 5.50: Indicadores da LSTM treinada com instâncias com e sem anomalia avaliados nas instâncias sem anomalia.

A tabela 5.51 exibe os resultados da LSTM treinada com instâncias com e sem anomalia calculados em instâncias com anomalia. Em geral, os indicadores são piores do que os do modelo treinado apenas com instâncias normais com fig:P-PDG-anorm-anorm3exceção da variável P-MON-CKP. A Fig. 5.6 apresenta o gráfico de uma série prevista da variável P-JUS-CKGL para o evento 2 bastante divergente do valor real. Por outro lado, a Fig. 5.8 expõe uma série da variável P-JUS-CKGL para o evento 5 que se assemelha bastante da série real. Por fim, a Fig. 5.8 apresenta uma série da variável P-PDG para o evento 3 que, em algumas partes, se aproxima da série real e que, em outras partes, se distancia bastante. Esse fenômeno é interessante, pois mostra o efeito de treinar o modelo com instâncias com todos os tipos de anomalia, isto é, o padrão previsto muito provavelmente é influenciado pelo comportamento dos outros tipos de evento indesejado que não o evento 3. Assim, um próximo passo é desenvolver uma metodologia para treinar modelos de previsão mais específicos por evento a fim de isolar os efeitos de cada um deles.

Variável	erro 5%	erro 10%	erro 15%	erro 20%	erro 25%
P_PDG	38,7%	40,1%	40,3%	40,3%	40,4%
P_TPT	6,8%	10%	15,8%	21,5%	34%
T_TPT	11,5%	14,5%	18,2%	19%	20,4%
P_MON_CKP	31,9%	59,6%	74,4%	77,8%	79,4%
T_JUS_CKP	12,1%	15,5%	16%	18,3%	20,4%
P_JUS_CKGL	0%	0,1%	0,1%	0,1%	0,1%

Tabela 5.51: Indicadores da LSTM treinada com instâncias com e sem anomalia avaliados nas instâncias com anomalia.

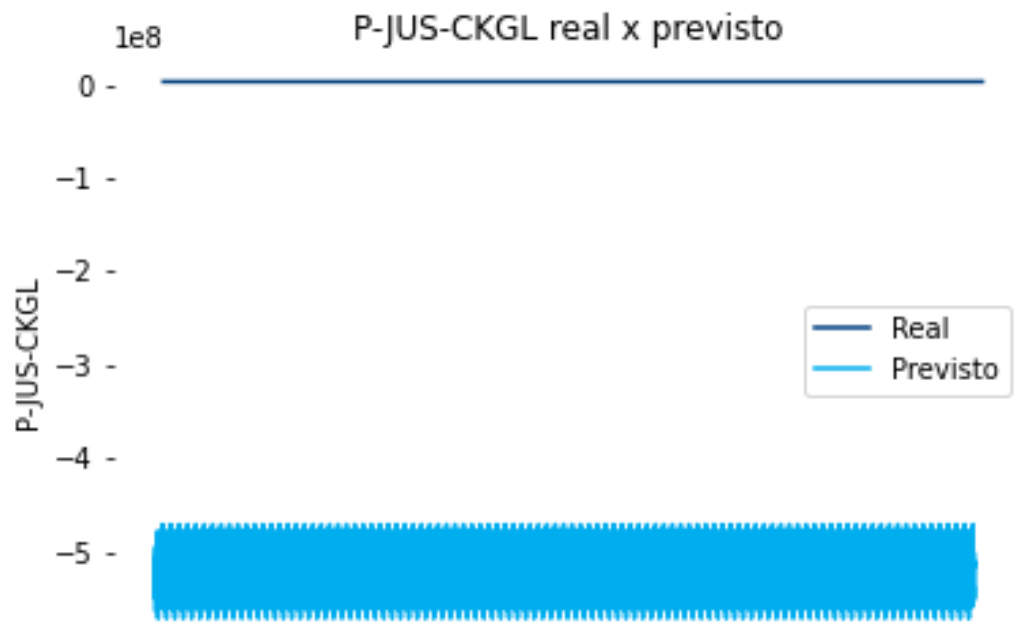

Figura 5.6: Comparação da série real com a série prevista para a variável P-JUS-CKGL com o evento 2. LSTM treinada com instâncias com e sem anomalia.

Figura 5.7: Comparação da série real com a série prevista para a variável P-MON-CKP com o evento 5. LSTM treinada com instâncias com e sem anomalia.

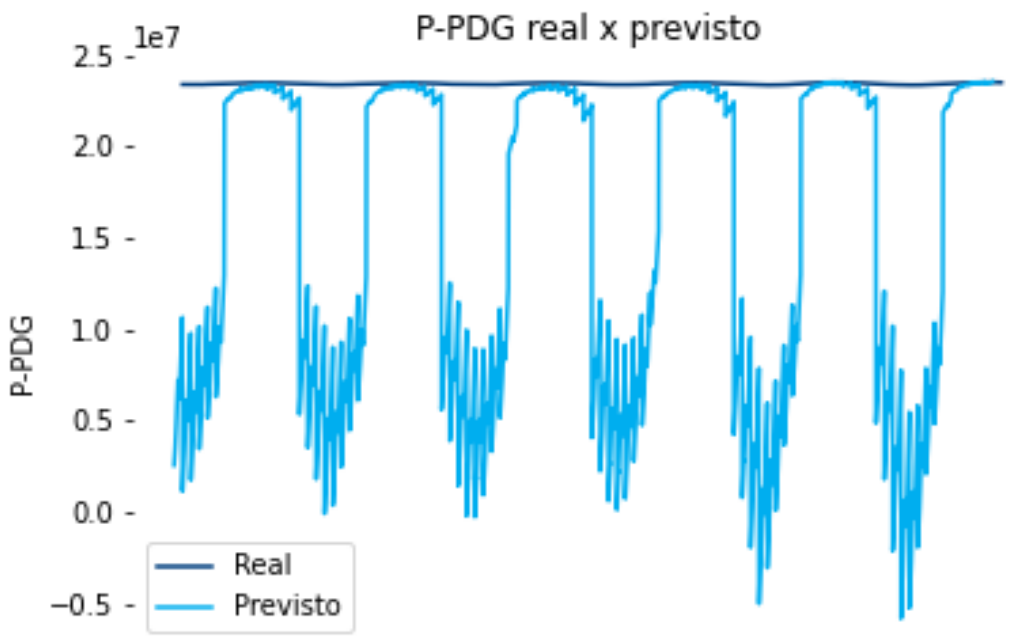

Figura 5.8: Comparação da série real com a série prevista para a variável P-PDG com o evento 3. LSTM treinada com instâncias com e sem anomalia.

Nesse contexto, a próxima seção apresenta outros resultados que evidenciam os pontos fortes e fracos das duas abordagens de treinamento do modelo.

5.5 Mesclagem de modelos

Nessa seção, são apresentados os resultados do produto final, isto é, o sensor virtual como é pretendido nos objetivos, a mesclagem entre o modelo de previsão e o de classificação. Os resultados da combinação entre a LSTM e o modelo de classificação não supervisionado não foram bons, pois o modelo de classificação não conseguiu distinguir entre o evento e o não evento. Por outro lado, as tabelas 5.52 à 5.55 apresentam os resultados da mesclagem entre os modelos de previsão e de classificação supervisionado em diferentes cenários, os quais são:

- Modelo supervisionado + LSTM treinada com instâncias sem anomalia, resultados avaliados em instâncias reais;
- Modelo supervisionado + LSTM treinada com instâncias com e sem anomalia, resultados avaliados em instâncias reais;
- Modelo supervisionado + LSTM treinada com instâncias sem anomalia, resultados avaliados em instâncias simuladas e desenhadas;

- Modelo supervisionado + LSTM treinada com instâncias com e sem anomalia, resultados avaliados em instâncias simuladas e desenhadas.

Cenário	% Evento detectado	% Não evento detectado
Instâncias normais	-	100%
Instâncias com anomalia	1,37%	100%

Tabela 5.52: Resultados da mesclagem entre o modelo de previsão treinado apenas com instâncias sem anomalia e o modelo de classificação supervisionado avaliados em séries reais.

Cenário	% Evento detectado	% Não evento detectado
Instâncias normais	-	0,13%
Instâncias com anomalia	98%	55%

Tabela 5.53: Resultados da mesclagem entre o modelo de previsão treinado com instâncias com e sem anomalia e o modelo de classificação supervisionado avaliados em séries reais.

Cenário	% Evento detectado	% Não evento detectado
Instâncias com anomalia	12%	81%

Tabela 5.54: Resultados da mesclagem entre o modelo de previsão treinado apenas com instâncias sem anomalia e o modelo de classificação supervisionado avaliados em séries simuladas e desenhadas.

Cenário	% Evento detectado	% Não evento detectado
Instâncias com anomalia	92%	0,42%

Tabela 5.55: Resultados da mesclagem entre o modelo de previsão treinado com instâncias com e sem anomalia e o modelo de classificação supervisionado avaliados em séries simuladas e desenhadas.

Os resultados acima mostram que a combinação de modelos funciona bem separadamente para detecção do evento e do não evento a depender se a LSTM foi treinada apenas com instâncias apenas sem anomalia ou com instâncias com e sem anomalia. Dessa forma, nenhuma das duas duas abordagens de treino do modelo de previsão é adequado para ambos os casos de classificação.

5.6 Modelo Final

Combinando os resultados apresentados com os conceitos de sensor virtual exposto no capítulo 1, chega-se ao modelo final de um sensor virtual para classificação exibido na Fig. 5.9. Com a metodologia desenvolvida, utiliza-se os dados dos sensores com 180 segundos de

atraso, tamanho da janela deslizante utilizada. Com essas 180 observações, as características são extraídas e normalizadas, então, o modelo é acionado, o qual gera o *output* do sensor virtual: uma classificação que indica a ocorrência de algum evento indesejado. Além disso, os resultados mostram que pode ser feito um sensor virtual para detectar anomalias em geral ou sensores virtuais diferentes para cada tipo de evento. Por outro lado, o modelo completo de sensor virtual para prognóstico de anomalias em poços de petróleo conforme é pretendido nos objetivos é apresentado na Fig. 5.10, no qual são usados ambos os modelos de previsão e de classificação. Portanto, os passos a serem seguidos são:

1. Normalização do *input* de cada sensor;
2. Previsão dos próximos 180 segundos de cada sensor;
3. Reversão da normalização;
4. Extração de variáveis dos 180 segundos previstos de todos os sensores;
5. Normalização das variáveis;
6. Seleção das variáveis;
7. Classificação do modelo;
8. *Output* final: detecção se há ou não anomalia na próxima janela de 180 segundos.

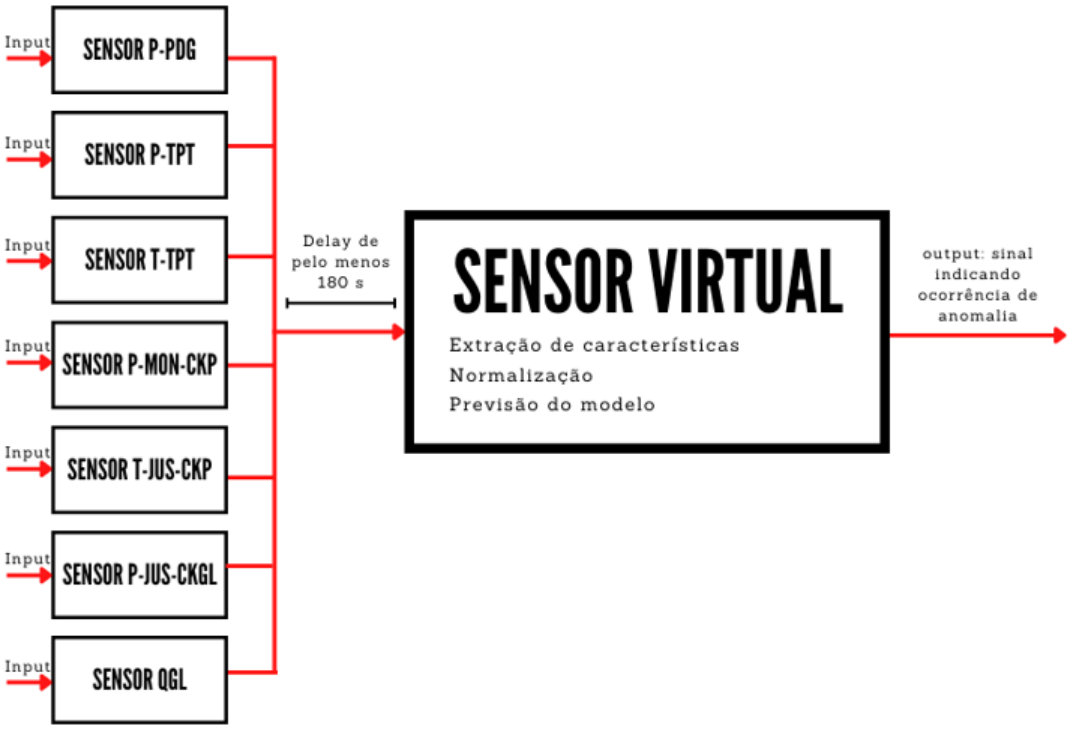

Figura 5.9: Esquema do sensor virtual proposto com base nos resultados obtidos.

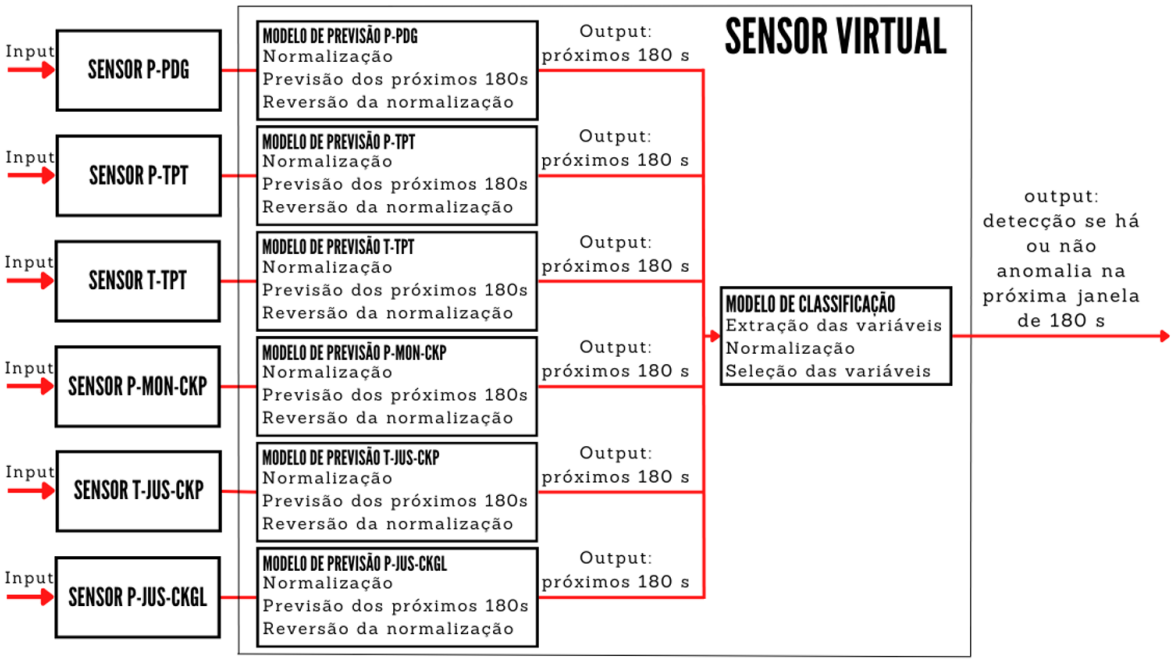

Figura 5.10: Esquema do sensor virtual proposto com base nos resultados obtidos.

Capítulo 6

Conclusão e Sugestões de Trabalhos Futuros

Diante do contexto de elevação e escoamento em poços de petróleo, faz-se necessário o monitoramento de eventos e de anomalias, objetivo que pode ser obtido mediante o uso de sensores virtuais e gêmeos digitais. Dessa forma, este trabalho buscou utilizar técnicas de Aprendizado de Máquina para detectar eventos indesejados em poços de petróleo mediante os dados do 3W *dataset*.

Portanto, de acordo com a metodologia desenvolvida por Vargas (2019) para *benchmark* dos algoritmos, o *Local Outlier Factor* obteve o maior valor para a métrica de referência, se mostrando a melhor alternativa em geral. Em consonância com isso, na segunda abordagem apresentada, utilizando instâncias reais como treino e teste e instâncias desenhadas como validação, o *Local Outlier Factor* também tem o melhor resultado. Diante disso, os modelos foram testados separadamente por tipo de evento, e constatou-se que, para cada evento, o algoritmo que melhor performa pode ser diferente. Em geral, os modelos de *One Class* SVM com diferentes *kernels* obtiveram as melhores performances. Isso acontece devida à diferença das dinâmicas de cada evento. Ademais, os modelos continuaram com uma boa performance nos dados de validação tanto para o evento um quanto para o evento sete.

Além disso, testou-se algoritmos supervisionados, os quais desempenharam bem tanto nos dados de teste quantos nos dados de validação. Entretanto, há redução de performance entre os dados de teste e de validação, o que pode acontecer devido ao sobreajuste aos dados de treinamento. Diante disso, a profundida das Árvores de Decisão foi limitada por intermédio de um hiperparâmetro a fim de investigar os efeitos no sobreajuste. Para a Floresta Aleatória, este processo causou apenas redução de desempenho nos dados de teste, ou seja, possivelmente não há sobreajuste. Por outro lado, o XGBoost apresentou quase a mesma performance nos dados de teste e melhorou o desempenho nos dados de validação, o que é um indicativo significativo de sobreajuste do modelo sem ajuste de hiperparâmetro, uma vez que, com o ele ajustado, o modelo generaliza melhor. Quando os dados de validação são separados em evento um e em evento sete, os modelos supervisionados continuam com bom

desempenho apenas para o evento sete. Ademais, esse algoritmos viabilizaram o conhecimento de quais variáveis mais contribuem na detecção de anomalias e de quais variáveis não agregam muito.

Nesse contexto, a utilização de janelas maiores realçou os resultados de alguns modelos, mas não foi, de fato, efetiva em melhorar a performance do modelo final. Esse fenômeno exibe uma limitação da base de dados que é possuir séries temporais curtas, o que inviabiliza a utilização das janelas de referência da Fig. 2.5 ou até de janelas menores, pois a quantidade de pontos extraídos com essas janelas é baixa, dificultando o treinamento dos modelos. Além disso, a aplicação de uma etapa de seleção de variáveis é importante, uma vez que elimina variáveis redundantes e diminui a dimensão do problema. Com isso, os modelos ou mantém ou melhora a performance, além de ficarem mais simples. Ainda no contexto de otimização da performance, a utilização de um modelo de classificação específico para cada evento não tem efeitos positivos sobre a performance, pois o modelo genérico se ajusta melhor para generalizar para dados novos, por exemplo, os indicadores encontrados para o evento 6 na abordagem supervisionado são melhores no modelo genérico. Em comparação com os resultados obtidos antes da etapa de otimização dos resultados, o modelo *Local Outlier Factor* com uma janela de 180 segundos continua como destaque quando avaliado em todos os eventos juntos. Entretanto, ele deixa de ser apenas um bom modelo e passa a ser destaque quando avaliado para cada evento separadamente, isso ocorre devido à utilização de uma metodologia padrão para amostragem dos dados de treino e de teste, a qual permite um balanceamento maior e facilita a comparação entre os resultados. Dessa forma, os modelos de detecção de anomalias finais utilizados em outras etapas são o *Local Outlier Factor* e a Floresta Aleatória, um não supervisionado e outro supervisionado.

Ademais, o modelo de previsão de de dados performa bem em dados sem anomalia quando é treinado apenas com esses dados, mas tem uma performance mediana quando aplicado em dados com anomalia. Por outro lado, o treinamento com instâncias com e sem anomalia não apresenta melhora significante na performance do modelo. A partir dos gráficos de séries previstas comparadas as séries reais, as previsões com erros pequenos ainda possuem, visualmente, uma certa distância dos dados reais, mas apresentam tendências semelhantes. Além disso, a partir dos gráficos, é perceptível que treinar o modelo com instâncias com anomalia de todos os 8 tipos de evento pode fazer com que o modelo construa tendências para um evento que não necessariamente aproximam-se da forma do dado real, isto é, ele reflete um comportamento de outros eventos nas previsões. Portanto, faz-se necessário desenvolver uma abordagem que permita treinar e aplicar os modelos de previsão separadamente para cada evento.

O objetivo inicial de desenvolver uma metodologia para prognóstico de anomalias é alcançado ao combinar os modelos de detecção com os modelos de previsão, ou seja, o modelo de previsão passa os dados futuros como *input* para o outro modelo detectar se há ou não anomalia nesses dados. Dessa forma, a mesclagem da LSTM com o modelo não supervisionado não foi positiva, pois o modelo de detecção não conseguiu distinguir entre o evento e o não

evento. Em contrapartida, a mesclagem com o modelo supervisionado mostra que, quando a LSTM é treinada apenas com instâncias sem anomalia, o produto final consegue detectar bem o não evento, mas não detecta o evento. O contrário também acontece, quando a LSTM é treinado com instâncias com e sem anomalia, o produto final detecta bem o evento mas não detecta o não evento. Isso mostra que nenhuma das duas abordagens para treinar a LSTM apresentam balanceamento quanto a performance para o evento e para o não evento. Portanto, a solução final de prognóstico de anomalias funciona separadamente para dados com anomalia e sem anomalias, porém esse fato inviabiliza sua aplicação direta em produção, isto é, faz-se necessário aprimorar a metodologia de treino do modelo de previsão para que, na combinação dos modelos, haja um *trade-off* melhor na detecção de eventos normais e de eventos indesejados.

Portanto, são apresentados dois modelos de sensores virtuais, o primeiro, representado na Fig. 5.9 no capítulo 5, deve ser usado exclusivamente para detecção de anomalias. O segundo, representado na Fig. 5.10 no capítulo 5, deve ser usado exclusivamente para prognóstico de anomalias. Dessa forma, o aprimoramento da metodologia apresentada para treinamento das LSTM apresentará o segundo sensor virtual mais assertivo, o que viabilizará que a área de Acompanhamento de Produção possa monitorar os poços, em conjunto com a aplicação do primeiro sensor virtual, com o objetivo final de garantir o escoamento e de reduzir os problemas na produção.

6.1 Sugestões de Trabalhos Futuros

Diante do exposto, há sugestões de pontos que podem ser melhorados para a aperfeiçoar a metodologia e para viabilizar a aplicação da solução em produção, as quais são:

- Treinar a LSTM com mais janelas como dados de entrada;
- Utilizar um algoritmo de otimização para os parâmetros da rede neural para cada sensor;
- Testar a abordagem de prever ponto a ponto da janela seguinte ao invés de prever uma janela de uma só vez;
- Otimizar a LSTM para que o modelo seja capaz de prever mais de uma janela no futuro;
- Mensurar a performance dos modelos de detecção na primeira janela em que o evento indesejado começa a acontecer.

Referências Bibliográficas

ALLWRIGHT, S. *What is a good F1 score and how do I interpret it?* 2022. <https://stephenallwright.com/good-f1-score/>. Acessado: 12/09/2022.

ANDREOLLI, I. Introdução à elevação e escoamento monofásico e multifásico de petróleo. *Editora Interciência*, v. 1, p. 594–595, 2016.

ANDRIANOV, N. *A Machine Learning Approach for Virtual Flow Metering and Forecasting*. 2018. Disponível em: <https://github.com/nikolai-andrianov/VFM/.>

BANERJEE, P. *Comprehensive Guide on Feature Selection*. 2020. <https://www.kaggle.com/code/prashant111/comprehensive-guide-on-feature-selection>. Acessado: 01/09/2022.

BARRICELLI, B. R.; CASIRAGHI, E.; FOGLI, D. A survey on digital twin: Definitions, characteristics, applications, and design implications. *IEEE Access*, Institute of Electrical and Electronics Engineers Inc., v. 7, 2019. ISSN 21693536.

BREUNIG, M. M. et al. Lof: Identifying density-based local outliers. 2000.

CADEI, L. et al. Achieving digital-twin through advanced analytics support: A novelty detection framework to highlight real-time anomalies in time series. 2020.

CHEN, T.; GUESTRIN, C. XGBoost: A scalable tree boosting system. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: ACM, 2016. (KDD '16), p. 785–794. ISBN 978-1-4503-4232-2. Disponível em: <http://doi.acm.org/10.1145/2939672.2939785>.

GÉRON, A. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. [S.l.]: "O'Reilly Media, Inc.", 2019.

GRIEVES, M.; VICKERS, J. *Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems*. [S.l.]: Springer International Publishing, 2016. 85-113 p.

JONES, D. et al. Characterising the digital twin: A systematic literature review. *CIRP Journal of Manufacturing Science and Technology*, Elsevier Ltd, v. 29, p. 36–52, 5 2020. ISSN 17555817.

JOVE, E. et al. Virtual sensor for fault detection, isolation and data recovery for bicomponent mixing machine monitoring. *Informatica (Netherlands)*, IOS Press, v. 30, p. 671–687, 2019. ISSN 08684952.

JUNIOR, W. F. *Comparação de Classificadores para Detecção de Anomalias em Poços Produtores de Petróleo*. Dissertação (Mestrado), 2022.

LIU, F. T.; TING, K. M.; ZHOU, Z. H. Isolation forest. *Proceedings - IEEE International Conference on Data Mining, ICDM*, p. 413–422, 2008. ISSN 15504786.

LIU, L.; KUO, S. M.; ZHOU, M. C. Virtual sensing techniques and their applications. *Proceedings of the 2009 IEEE International Conference on Networking, Sensing and Control, ICNSC 2009*, p. 31–36, 2009.

MARINS, M. A. et al. Fault detection and classification in oil wells and production/service lines using random forest. *Journal of Petroleum Science and Engineering*, Elsevier B.V., v. 197, 2 2021. ISSN 09204105.

MEGLIO, F. D. et al. Stabilization of slugging in oil production facilities with or without upstream pressure sensors. *Journal of Process Control*, Elsevier Ltd, v. 22, p. 809–822, 2012. ISSN 09591524.

MEHTA, P. et al. A high-bias, low-variance introduction to machine learning for physicists. *Physics Reports*, Elsevier B.V., v. 810, p. 1–124, 5 2019. ISSN 03701573.

MIN, Q. et al. Machine learning based digital twin framework for production optimization in petrochemical industry. *International Journal of Information Management*, Elsevier Ltd, v. 49, p. 502–519, 12 2019. ISSN 02684012.

MOHR, J.-P. *Digital twins for the oil and gas industry*. [S.l.: s.n.], 2018.

PARROTT, A.; WARSHAW, L. *Industry 4.0 and the digital twin*. 2017.

PEDREGOSA, F. et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, v. 12, p. 2825–2830, 2011.

PODDAR, T. *Digital twin bridging intelligence among man, machine and environment*. [S.l.: s.n.], 2018. 1-4 p.

SAINI, G. et al. Accelerating well construction using a digital twin demonstrated on unconventional well data in north america. p. 3264–3276, 2018.

SHARMA, P. et al. *The Dawn of the New Age of the Industrial Internet and How it can Radically Transform the Offshore Oil and Gas Industry*. 2017.

SHARMA, P. et al. Rb-fea based digital twin for structural integrity assessment of offshore structures. p. 1–6, 2018.

VARGAS, R. Base de dados e *Benchmarks* para prognóstico de anomalias em sistemas de elevação de petróleo. 2019.

VARGAS, R. E. V. et al. A realistic and public dataset with rare undesirable real events in oil wells. *Journal of Petroleum Science and Engineering*, Elsevier B.V., v. 181, p. 106223, 10 2019. ISSN 09204105.

VLASVELD, R. *Introduction to One-class Support Vector Machines*. 2013. <http://rvlasveld.github.io/blog/2013/07/12/introduction-to-one-class-support-vector-machines/>.

WAGG, D. J. et al. Digital twins: State-of-the-art and future directions for modeling and simulation in engineering dynamics applications. *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part B: Mechanical Engineering*, American Society of Mechanical Engineers (ASME), v. 6, 9 2020. ISSN 23329025.

WANASINGHE, T. R. et al. Digital twin for the oil and gas industry: Overview, research trends, opportunities, and challenges. *IEEE Access*, Institute of Electrical and Electronics Engineers Inc., v. 8, p. 104175–104197, 2020. ISSN 21693536.

Apêndice

6.2 Códigos Utilizados no Trabalho

Os códigos ficaram com uma quantidade de linhas grande. Por isso, os dados foram disponibilizados em <https://github.com/ArthurAlves7/PG_ArthurAlves>.