

INSTITUTO FEDERAL DO ESPÍRITO SANTO  
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO APLICADA

**WANDER FERNANDES JÚNIOR**

**COMPARAÇÃO DE CLASSIFICADORES PARA DETECÇÃO DE ANOMALIAS  
EM POÇOS PRODUTORES DE PETRÓLEO**

Serra  
2022

WANDER FERNANDES JÚNIOR

**COMPARAÇÃO DE CLASSIFICADORES PARA DETECÇÃO DE ANOMALIAS  
EM POÇOS PRODUTORES DE PETRÓLEO**

Dissertação apresentada ao Programa de Pós-Graduação em Computação Aplicada do Instituto Federal do Espírito Santo como requisito parcial para obtenção do título de Mestre em Computação Aplicada.

Orientadora: Prof<sup>a</sup>. Dr<sup>a</sup>. Kelly Assis de Souza Gazolli

Orientadora: Prof<sup>a</sup>. Dr<sup>a</sup>. Karin Satie Komati

Serra  
2022

**WANDER FERNANDES JÚNIOR**

**COMPARAÇÃO DE CLASSIFICADORES PARA DETECÇÃO DE ANOMALIAS  
EM POÇOS PRODUTORES DE PETRÓLEO**

Dissertação apresentada ao Programa de Pós-Graduação em Computação Aplicada do Instituto Federal do Espírito Santo como requisito parcial para obtenção do título de Mestre em Computação Aplicada.

Aprovado em 28 de janeiro de 2022.

**COMISSÃO EXAMINADORA**

---

Prof<sup>a</sup>. Dr<sup>a</sup>. Kelly Assis de Souza Gazolli  
Instituto Federal do Espírito Santo  
Orientadora

---

Prof<sup>a</sup>. Dr<sup>a</sup>. Karin Satie Komati  
Instituto Federal do Espírito Santo  
Orientadora

---

Prof. Dr. Hilário Seibel Júnior  
Instituto Federal do Espírito Santo  
Membro interno

---

Prof. Dr. Patrick Marques Ciarelli  
Universidade Federal do Espírito Santo  
Membro externo

*Para Juliana, Davi e Mariana.*

## **AGRADECIMENTOS**

À Prof<sup>a</sup>. Dr<sup>a</sup>. Kelly Assis de Souza Gazolli e à Prof<sup>a</sup>. Dr<sup>a</sup>. Karin Satie Komati pela orientação, paciência e motivação.

Aos membros da banca examinadora, Prof. Dr. Hilário Seibel Júnior e Prof. Dr. Patrick Marques Ciarelli, pelas contribuições na geração desta dissertação.

Ao colega Dr. Ricardo Emanuel Vaz Vargas pelo indispensável apoio às dúvidas sobre a base de dados e o *benchmark* aqui utilizado.

À Petrobras pela liberação parcial no período entre o exame de qualificação e a defesa, em especial ao gestor Pedro Benoni Santos Gonçalves pela oportunidade de aprimoramento profissional.

Aos meus pais Wander e Joana pelo suporte em todas as fases da vida e por sempre terem priorizado o estudo dos filhos.

À minha esposa Juliana pelo amor e incentivo em todos os momentos.

Aos meus filhos Davi e Mariana por existirem.

E a Deus, pela família que tenho e por ter me dado saúde e força realizar este trabalho.

## RESUMO

Anomalias em poços produtores de petróleo podem causar impactos financeiros significativos. O uso de aprendizado de máquina para detectar essas situações pode prevenir interrupções indesejadas de produção bem como custos de manutenção. Neste contexto, este trabalho realizou a aplicação e comparação de classificadores para detecção de anomalias em poços marítimos produtores surgentes de petróleo e gás (poços que conseguem escoar os fluidos produzidos até a plataforma com sua própria pressão) utilizando os dados da base pública denominada 3W *dataset*. Por serem citados pela literatura apresentarem diferentes características em seus modelos preditivos, foram aplicados os seguintes classificadores de classe única: Floresta de Isolamento, *One-class Support Vector Machine* (OCSVM), *Local Outlier Factor* (LOF), Envelope Elíptico e *Autoencoder* com camadas *feedforward* e LSTM (*Long short-term memory*). Os experimentos realizados foram divididos em duas partes. Na primeira parte foi usado o *benchmark* para detecção de anomalias proposto por Vargas (2019). Esse *benchmark* demanda a geração de classificadores a nível de instância, ou seja, é gerado um classificador para cada instância treinada. Foram feitos experimentos com e sem a etapa de extração de características. Nos experimentos com extração de características, foram extraídas a mediana, média, desvio padrão, variância, máximo e mínimo para cada variável. Nos experimentos sem extração de características, as próprias séries temporais foram utilizadas como entrada para os classificadores. Os testes estatísticos de Friedman e Wilcoxon foram utilizados para avaliar se os classificadores testados geram métricas de desempenho cuja média é diferente em relação às demais. O melhor desempenho foi obtido pelo LOF com medidas F1 de 87,0% e 85,9% nos experimentos com e sem extração de características, respectivamente. Os resultados obtidos apresentaram melhoria estatística em comparação ao *benchmark* de referência. Na segunda parte do experimento, a fim de verificar o desempenho das redes neurais em um cenário com maior quantidade de dados, foram realizadas experimentações com o agrupamento das amostras das instâncias. Esse agrupamento significa que todas as amostras das instâncias foram utilizadas conjuntamente como entrada para o classificador, ou seja, foi gerado um classificador único de cada tipo para todo o conjunto de instâncias treinada. Como esse uso conjunto de instâncias não foi previsto no *benchmark* para detecção de anomalias original, esses experimentos foram denominados complementares. Esse cenário com agrupamento mostrou que a maior disponibilidade de dados aumentou o desempenho numérico das redes neurais, com medida F1 de 81,5%.

Palavras-chave: Detecção de Anomalias; Monitoramento de Poços de Petróleo; Séries Temporais Multivariadas.

## ABSTRACT

Anomalies in oil-producing wells can have significant financial impacts. Using machine learning to detect these situations can prevent unwanted production disruptions and maintenance costs. In this context, this work compared classifiers for anomalies detection in naturally flowing offshore oil and gas producing wells (wells that manage to drain the fluids produced to the platform with their pressure) using data from the public database called 3W dataset. As cited in the literature as having different characteristics in their predictive models, the following one-class classifiers were applied: Isolation Forest, One-class Support Vector Machine (OCSVM), Local Outlier Factor (LOF), Elliptical envelope, and Autoencoder with layers feedforward and LSTM (Long short-term memory). The experiments performed were divided into two parts. In the first part, the anomalies detection used the benchmark proposed by Vargas (2019). This benchmark demands the generation of classifiers at the instance level, which means one model for each trained instance. The first experiment analyses the results with and without the feature extraction step. The feature extraction, for each variable, the median, mean, standard deviation, variance, maximum, and minimum were extracted. In experiments without feature extraction, the time series themselves were the input to the classifiers. Friedman and Wilcoxon's statistical tests assess if the classifiers generate performance metrics whose average is different from the others. LOF classifier presented the best performance, with F1-measure of 87.0% and 85.9% in the experiments with and without feature extraction, respectively. The results obtained showed statistical improvement compared to the benchmark. In the second experiment, the performance of neural networks in a scenario with more data, grouping samples of instances. This grouping means that all data are input to the classifier, one model of each type for the entire set of instances. The joint of instances was not foreseen in the benchmark for original anomaly detection were called complementary. This clustered scenario showed that greater data availability increased the numerical performance of neural networks with an F1 measure of 81.5%.

Keywords: Anomaly Detection; Oil Well Monitoring; Multivariate Time Series.

## LISTA DE FIGURAS

|   |    |
|---|----|
| Figura 1 – Projeção da demanda de energia no mundo por tipo de fonte. . . . .   | 11 |
| Figura 2 – Esquema simplificado de um sistema de produção de petróleo. . . . .  | 16 |
| Figura 3 – Exemplo de instância da classe “Aumento Abrupto de BSW” da base de dados 3W <i>dataset</i> . . . . .   | 20 |
| Figura 4 – Exemplo de instância da classe “Fechamento Espúrio de DHSV” da base de dados 3W <i>dataset</i> . . . . .   | 20 |
| Figura 5 – Etapas do processo de classificação. . . . .   | 23 |
| Figura 6 – Representação da etapa de pré-processamento. . . . .   | 24 |
| Figura 7 – Representação do processo de extração de características. . . . .  | 25 |
| Figura 8 – <i>One-class SVM</i> (OCSVM): tem como objetivo separar o conjunto de dados da origem e construir uma hiperesfera que engloba todas as instâncias normais em um espaço. . . . .                                    | 27 |
| Figura 9 – Floresta de Isolamento: as anomalias em geral estão em regiões menos populosas e geralmente são necessárias menos partições aleatórias para isolá-las em nós da árvore. . . . .                                    | 28 |
| Figura 10 – <i>Local Outlier Factor</i> (LOF): comparação da densidade local de um ponto com as densidades dos vizinhos. O elemento 'A' tem uma densidade muito menor do que seus vizinhos. . . . .                           | 30 |
| Figura 11 – Envelopes elípticos gerados com estatística clássica e estatística robusta, em que todos os elementos internos à figura geométrica são normais e todos os elementos externos à elipse são anormalidades . . . . . | 31 |
| Figura 12 – Unidade básica da rede neural Perceptron. . . . .   | 32 |
| Figura 13 – Exemplo de uma rede neural do tipo Autoencoder. . . . .   | 34 |
| Figura 14 – Estrutura básica de uma rede neural recorrente (RNN). . . . .   | 35 |
| Figura 15 – Fluxo de informações nas estruturas internas das redes RNN e LSTM. .  | 36 |
| Figura 16 – Detalhamento das partes presentes nas estruturas internas da célula LSTM. . . . .   | 37 |
| Figura 17 – Combinação entre LOF e características reduzidas geradas pelo <i>Autoencoder</i> LSTM. . . . .  | 45 |
| Figura 18 – Curvas de aprendizado e distribuição do erro absoluto médio de reconstrução para as amostras das instâncias. . . . .  | 47 |
| Figura 19 – Curvas de aprendizado do experimento com agrupamento de instâncias e sem extração de características, utilizando instâncias reais e inclusão gradativa das demais instâncias simuladas de 10% em 10% até 50%. . . | 49 |

## LISTA DE TABELAS

|  |    |
|--|----|
| Tabela 1 – Descrição das variáveis das séries temporais presentes no 3W <i>dataset</i> . . . . .   | 17 |
| Tabela 2 – Divisão das instâncias por classe e por fonte no 3W <i>dataset</i> . . . . .  | 18 |
| Tabela 3 – Tamanhos de janela de tempo até o alcance do estado estável de anomalia no 3W <i>dataset</i> . . . . .  | 19 |
| Tabela 4 – Matriz de confusão para avaliação de desempenho. . . . .  | 39 |
| Tabela 5 – Médias da medida F1 e desvio padrão (entre parênteses) do experimento com extração de características, por algoritmo (com melhores parâmetros destacados em negrito), para o melhor caso nas instâncias simuladas e nas instâncias reais. . . . . | 41 |
| Tabela 6 – Valores $p$ dos testes estatísticos de Wilcoxon com ajuste de Bonferroni dos experimentos com extração de características. . . . .  | 42 |
| Tabela 7 – Médias da medida F1 e desvio padrão (entre parênteses) do experimento sem extração de características, por algoritmo (com melhores parâmetros destacados em negrito), para o melhor caso nas instâncias simuladas e nas instâncias reais. . . . . | 43 |
| Tabela 8 – Valores $p$ dos testes estatísticos de Wilcoxon com ajuste de Bonferroni dos experimentos sem extração de características. . . . .  | 44 |
| Tabela 9 – Medida F1 do experimento com agrupamento de instâncias e sem extração de características, por algoritmo (com melhores parâmetros destacados em negrito), para o melhor caso nas instâncias simuladas e nas instâncias reais. . . . .              | 46 |
| Tabela 10 – Medida F1 do experimento com agrupamento de instâncias e sem extração de características, utilizando 50% das instâncias simuladas. . . . .   | 48 |
| Tabela 11 – Medida F1 do experimento com agrupamento de instâncias e sem extração de características, utilizando instâncias reais e inclusão gradativa das demais instâncias simuladas de 10% em 10% até 50%. . . . .  | 48 |

## LISTA DE SIGLAS

|            |  |
|------------|--|
| ANP        | – Agência Nacional do Petróleo, Gás Natural e Biocombustíveis            |
| BSW        | – <i>Basic Sediment and Water</i>  |
| CKP        | – <i>Choke</i> de produção (válvula de controle)                         |
| CKGL       | – <i>Choke</i> de <i>gas lift</i> (válvula de controle)                  |
| CSV        | – <i>Comma-Separated Values</i>  |
| DHSV       | – <i>Down Hole Safety Valve</i>  |
| FN         | – Falso Negativo   |
| FP         | – Falso Positivo   |
| KNN        | – <i>K-Nearest Neighbors</i>   |
| LSTM       | – <i>Long short-term memory</i>  |
| PDG        | – <i>Permanent Downhole Gauge</i>  |
| P-MON-CKP  | – Pressão do fluido montante à válvula CKP                               |
| P-JUS-CKGL | – Pressão do fluido jusante à válvula de controle de <i>gas lift</i>     |
| P-PDG      | – Pressão do fluido no PDG   |
| P-TPT      | – Pressão do fluido no TPT   |
| QGL        | – Vazão de <i>gas lift</i>   |
| RNN        | – <i>Recurrent Neural Network</i>  |
| SVM        | – <i>Support Vector Machine</i>  |
| TN         | – Verdadeiro Negativo  |
| TP         | – Verdadeiro Positivo  |
| TPT        | – <i>Temperature and Pressure Transducer</i>                             |
| T-JUS-CKP  | – Temperatura do fluido jusante à válvula CKP                            |
| T-JUS-CKGL | – Temperatura do fluido jusante à válvula de controle de <i>gas lift</i> |
| T-TPT      | – Temperatura do fluido no TPT   |

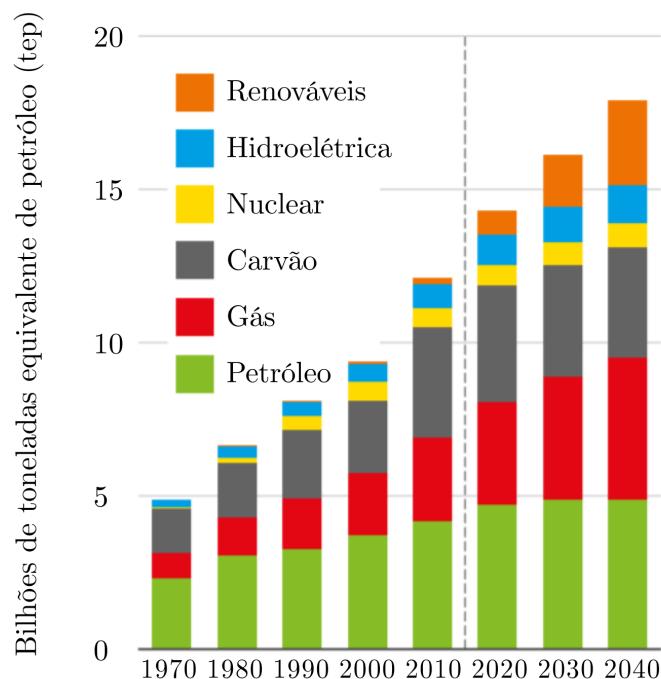
## SUMÁRIO

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>INTRODUÇÃO . . . . .</b>                                       | <b>11</b> |
| 1.1      | OBJETIVO GERAL . . . . .  | 13        |
| 1.2      | OBJETIVOS ESPECÍFICOS . . . . .                                   | 14        |
| 1.3      | PUBLICAÇÃO ASSOCIADA . . . . .                                    | 14        |
| 1.4      | ORGANIZAÇÃO DO TRABALHO . . . . .                                 | 14        |
| <b>2</b> | <b>REFERENCIAL TEÓRICO . . . . .</b>                              | <b>15</b> |
| 2.1      | SISTEMAS DE PRODUÇÃO DE PETRÓLEO . . . . .                        | 15        |
| 2.2      | ANOMALIAS EM POÇOS DE PETRÓLEO . . . . .                          | 15        |
| 2.3      | TRABALHOS CORRELATOS . . . . .                                    | 21        |
| <b>3</b> | <b>METODOLOGIA . . . . .</b>                                      | <b>23</b> |
| 3.1      | PRÉ-PROCESSAMENTO . . . . .                                       | 23        |
| 3.2      | EXTRAÇÃO DE CARACTERÍSTICAS . . . . .                             | 25        |
| 3.3      | REDUÇÃO DE DIMENSIONALIDADE . . . . .                             | 25        |
| 3.4      | CLASSIFICAÇÃO . . . . .   | 26        |
| 3.4.1    | One-class Support Vector Machine (OCSVM) . . . . .                | 26        |
| 3.4.2    | Floresta de Isolamento . . . . .                                  | 27        |
| 3.4.3    | Local Outlier Factor (LOF) . . . . .                              | 29        |
| 3.4.4    | Envelope Elíptico . . . . .                                       | 30        |
| 3.4.5    | Redes Neurais Artificiais . . . . .                               | 32        |
| 3.4.5.1  | Autoencoder . . . . .   | 33        |
| 3.4.5.2  | <i>Long short-term memory</i> (LSTM) . . . . .                    | 35        |
| 3.5      | AVALIAÇÃO DE DESEMPENHO . . . . .                                 | 38        |
| <b>4</b> | <b>RESULTADOS E DISCUSSÕES . . . . .</b>                          | <b>40</b> |
| 4.1      | EXPERIMENTOS COM REGRAS DO <i>BENCHMARK</i> . . . . .             | 40        |
| 4.1.1    | Experimentos com extração de características . . . . .            | 40        |
| 4.1.2    | Experimentos sem extração de características . . . . .            | 43        |
| 4.2      | EXPERIMENTOS COM AGRUPAMENTO DE INSTÂNCIAS . . . . .              | 45        |
| 4.2.1    | Uso conjunto de Autoencoder LSTM e Local Outlier Factor . . . . . | 45        |
| 4.2.2    | Combinação de instâncias reais e simuladas . . . . .              | 47        |
| <b>5</b> | <b>CONCLUSÃO . . . . .</b>  | <b>50</b> |
| 5.1      | TRABALHOS FUTUROS . . . . .                                       | 51        |
|          | <b>REFERÊNCIAS . . . . .</b>                                      | <b>52</b> |

## 1 INTRODUÇÃO

Petróleo é uma matéria-prima essencial à vida moderna, sendo componente básico para diversos tipos de indústrias. Dele, se produz gasolina, combustível de aviação, gás de cozinha, lubrificantes, borrachas, plásticos, tecidos sintéticos, tintas e energia elétrica (GAUTO et al., 2016). A base da matriz energética mundial mantém grande dependência da indústria de petróleo e gás (PORTELA, 2015). Conforme ilustrado na Figura 1, que apresenta os bilhões de toneladas equivalentes de petróleo no eixo das ordenadas e as décadas no eixo das abscissas, de 1970 a 2040, a demanda mundial por energia apresentou crescimento e tem projeção de aumento no futuro, sendo o petróleo e o gás (as partes verde e vermelha de cada barra) responsáveis por aproximadamente metade de toda a demanda em 2040.

Figura 1 – Projeção da demanda de energia no mundo por tipo de fonte.



Fonte: traduzido do relatório da British Petroleum Energy Outlook (BP, 2019).

Conforme dados da Agência Nacional do Petróleo, Gás Natural e Biocombustíveis (ANP, 2020), a produção de petróleo e gás natural no Brasil em janeiro de 2020 foi de 3,168 MMbbl/d (milhões de barris por dia) e 139 MMm<sup>3</sup>/d (milhões de metros cúbicos por dia), respectivamente. Essa produção foi proveniente de 7.227 poços, sendo 649 marítimos e 6.558 terrestres. A produção do pré-sal<sup>1</sup> correspondeu a 66,4% desse total e foi oriunda de 119 poços marítimos, o que equivale a uma média de produção por poço de aproximadamente 18 Mbbl/d (milhares de barris por dia).

<sup>1</sup> “Pré-sal” refere-se à produção de hidrocarbonetos realizada no horizonte geológico denominado “Pré-sal”, em campos localizados na área definida no inciso IV do caput do art. 2º da Lei nº 12.351, de 2010.

Durante a produção de petróleo, é possível a ocorrência de eventos indesejados denominados anomalias, que podem provocar impactos financeiros significativos. Como exemplo, pode-se citar a incrustação (ocorrência de depósitos inorgânicos) em válvulas que podem reduzir drasticamente a produção de petróleo (VARGAS, 2019).

Assim, é importante que o processo de produção de petróleo seja monitorado a fim de detectar e classificar as anomalias. De acordo com Qin (2012), uma possível solução é a aplicação de estatísticas multivariadas e métodos de aprendizado de máquina para detecção e classificação de anomalias. O monitoramento de processos industriais orientado a dados aplica estatísticas multivariadas e métodos de aprendizado de máquina para detectar e classificar anomalias em processos operacionais. A detecção de anomalias é uma classificação do tipo binária (entre normalidade e anormalidade), na qual identifica-se a ocorrência de anomalia, porém sem especificá-la (VARGAS, 2019).

Na área de aprendizado de máquina, o problema de classificação pode ser definido como a categorização de uma determinada entrada em uma ou mais classes discretas e pré-definidas (KADHIM, 2019). Em muitos processos industriais, busca-se detectar padrões raros, nos quais a maioria das observações referem-se a situações de normalidade, e a minoria, às situações raras que se deseja identificar (SANTOS; KERN, 2016). Nesses casos, a detecção de padrões novos (*novelty detection*) pode ser feita com classificadores de classe única, nos quais utiliza-se no treinamento apenas dados associados à classe comum (normalidade) (KHAN; MADDEN, 2014).

Em processos industriais, os dados de entrada para o monitoramento são provenientes de vários sensores e indexados no tempo, ou seja, são séries temporais multivariadas. Conforme escrito por Fawaz et al. (2019), nas duas últimas décadas a classificação de séries temporais tem sido considerada como um dos problemas mais desafiadores em mineração de dados. Uma das dificuldades é que as anomalias não possuem um conjunto de características ou regras que as agregam. Uma anomalia pode ser pontual, isto é, um único valor extremo (como um valor de temperatura acima de um limiar) pode ser o suficiente para caracterizar uma anomalia. Porém, um valor que pontualmente pode ser considerado normal pode ser considerado anormal em um determinado contexto, por exemplo, uma mudança brusca de temperatura durante um processo industrial, mesmo que o valor inicial e final da mudança não sejam atípicos isoladamente (CHANDOLA; BANERJEE; KUMAR, 2009).

Este trabalho aplicou e comparou técnicas de aprendizado de máquina na detecção de anomalias em poços produtores de petróleo, utilizando a base de dados 3W dataset (VARGAS et al., 2019), composta por séries temporais multivariadas. As seguintes técnicas de classificadores de classe única foram comparadas: Floresta de Isolamento (em inglês, *Isolation Forest*), Máquina de Vetor de Suporte de Classe Única (OCSVM do inglês,

*One-class Support Vector Machine*), Fator de Anomalia Local (LOF do inglês *Local Outlier Factor*), Envelope Elíptico (MCD, do inglês *Minimum Covariance Determinant*) e redes neurais do tipo *Autoencoder* com camadas *feedforward* e recorrentes do tipo LSTM (*Long Short-Term Memory*).

A base de dados utilizada 3W *dataset* (VARGAS et al., 2019) é pública e contém 1.984 instâncias de séries temporais da produção de poços de petróleo marítimos do tipo surgente (poços que conseguem escoar os fluidos produzidos até a plataforma com sua própria pressão). Essas instâncias foram separadas em: operação em condições normais e anomalias. As anomalias foram organizadas em oito classes. Essa base pode ser utilizada tanto para detecção quanto para classificação de anomalias em poços de petróleo. Além da base com dados de produção reais, Vargas (2019) também elaborou dois *benchmarks* específicos, sendo um para avaliação do impacto do uso de instâncias simuladas e desenhadas à mão e outro para detecção de anomalias, que podem ser utilizados para permitir que algoritmos propostos por diferentes pesquisadores tenham seus desempenhos avaliados e comparados.

Os experimentos realizados nessa dissertação foram divididos em duas partes. Na primeira parte foi usado o *benchmark* para detecção de anomalias proposto por Vargas (2019). Esse *benchmark* demanda a geração dos classificadores a nível de instância, ou seja, é gerado um classificador para cada instância treinada. Foram feitos experimentos com e sem a etapa de extração de características. Nos experimentos com extração de características, foram extraídas a mediana, média, desvio padrão, variância, máximo e mínimo para cada variável. Nos experimentos sem extração de características, as próprias séries temporais foram utilizadas como entrada para os classificadores. Os testes estatísticos de Friedman e de Wilcoxon foram utilizados para avaliar se os classificadores testados geram métricas de desempenho cuja média é diferente em relação às demais. Na segunda parte do experimento, a fim de verificar o desempenho das redes neurais em um cenário com maior quantidade de dados, foram realizadas experimentações com o agrupamento de instâncias. Esse agrupamento significa que todas as instâncias foram utilizadas conjuntamente como entrada para o classificador, ou seja, foi gerado um classificador único de cada tipo para todo o conjunto de instâncias treinada. Como esse uso conjunto de instâncias não foi previsto no *benchmark* para detecção de anomalias proposto por Vargas (2019), esses experimentos foram denominados complementares.

## 1.1 OBJETIVO GERAL

O objetivo geral deste trabalho é aplicar e comparar quantitativamente o resultado das técnicas de detecção de anomalias em poços de petróleo marítimos do tipo surgente utilizando a base de dados pública 3W e o *benchmark* associado para detecção de anomalias.

## 1.2 OBJETIVOS ESPECÍFICOS

Para alcançar o objetivo geral, os seguintes objetivos específicos foram planejados:

1. Realizar levantamento bibliográfico sobre trabalhos correlatos sobre detecção de anomalias;
2. Realizar levantamento bibliográfico sobre algoritmos de detecção de anomalias de classe única: OCSVM, Envelope Elíptico, LOF, Floresta de Isolamento e redes neurais com arquitetura do tipo *autoencoder* com camadas *feedforward* e LSTM;
3. Aplicar os métodos de detecção à base de dados de anomalias em poços de petróleo do tipo surgente 3W *dataset*;
4. Comparar os resultados obtidos com os demais classificadores testados e com os resultados obtidos no *benchmark* proposto por Vargas (2019);
5. Realizar testes estatísticos entre os resultados para avaliar se algum dos classificadores pode ser considerado melhor que os demais.

## 1.3 PUBLICAÇÃO ASSOCIADA

Resultados obtidos foram apresentados em artigo intitulado “Detecção de anomalias em poços produtores de petróleo usando aprendizado de máquina”, aceito e apresentado na XXIII Congresso Brasileiro de Automática (CBA 2020) (JÚNIOR et al., 2020).

## 1.4 ORGANIZAÇÃO DO TRABALHO

O presente trabalho está dividido em 5 capítulos. Este Capítulo 1 traz uma introdução ao assunto abordado, a justificativa para a sua realização, os objetivos pretendidos e a forma como o trabalho foi organizado.

No Capítulo 2 apresenta-se o referencial teórico com os principais conceitos presentes na literatura sobre o tema. São apresentados os principais conceitos relativos a sistemas de produção de petróleo, a descrição da base de dados utilizada e trabalhos correlatos sobre detecção de anomalias.

Em seguida, o Capítulo 3 traz a metodologia aplicada à base de dados de anomalias em poços de petróleo. Apresenta-se a abordagem utilizada para o realização dos experimentos e coleta de resultados.

No Capítulo 4 são apresentados e discutidos os resultados obtidos. Por fim, no Capítulo 5 o trabalho é encerrado com as conclusões e trabalhos futuros.

## 2 REFERENCIAL TEÓRICO

Neste capítulo são apresentados os principais conceitos relativos a sistemas de produção de petróleo, anomalias em poços de petróleo além de uma descrição detalhada da base 3W dataset e os trabalhos correlatos e recentes sobre detecção de anomalias.

### 2.1 SISTEMAS DE PRODUÇÃO DE PETRÓLEO

Um poço de petróleo é uma estrutura perfurada no solo em etapas que formam um telescópio invertido (os diâmetros diminuem à medida que a profundidade aumenta) e munida com equipamentos e sensores que permitem o fluxo de petróleo e gás da rocha reservatório de petróleo até a superfície (GUO, 2011).

Para que a produção de petróleo e gás seja possível no ambiente marítimo, os poços são conectados a sistemas compostos de equipamentos submarinos instalados no leito marinho e linhas que permitem o controle do poço e o escoamento do petróleo até uma plataforma de produção, armazenamento e transferência (BAI; BAI, 2015).

A Figura 2 traz um esquema simplificado de um sistema de produção de petróleo, contemplando o poço, o sistema submarino e a plataforma. O óleo e o gás fluem de uma rocha reservatório de petróleo através da coluna de produção e, em seguida, através de uma linha de produção para uma plataforma.

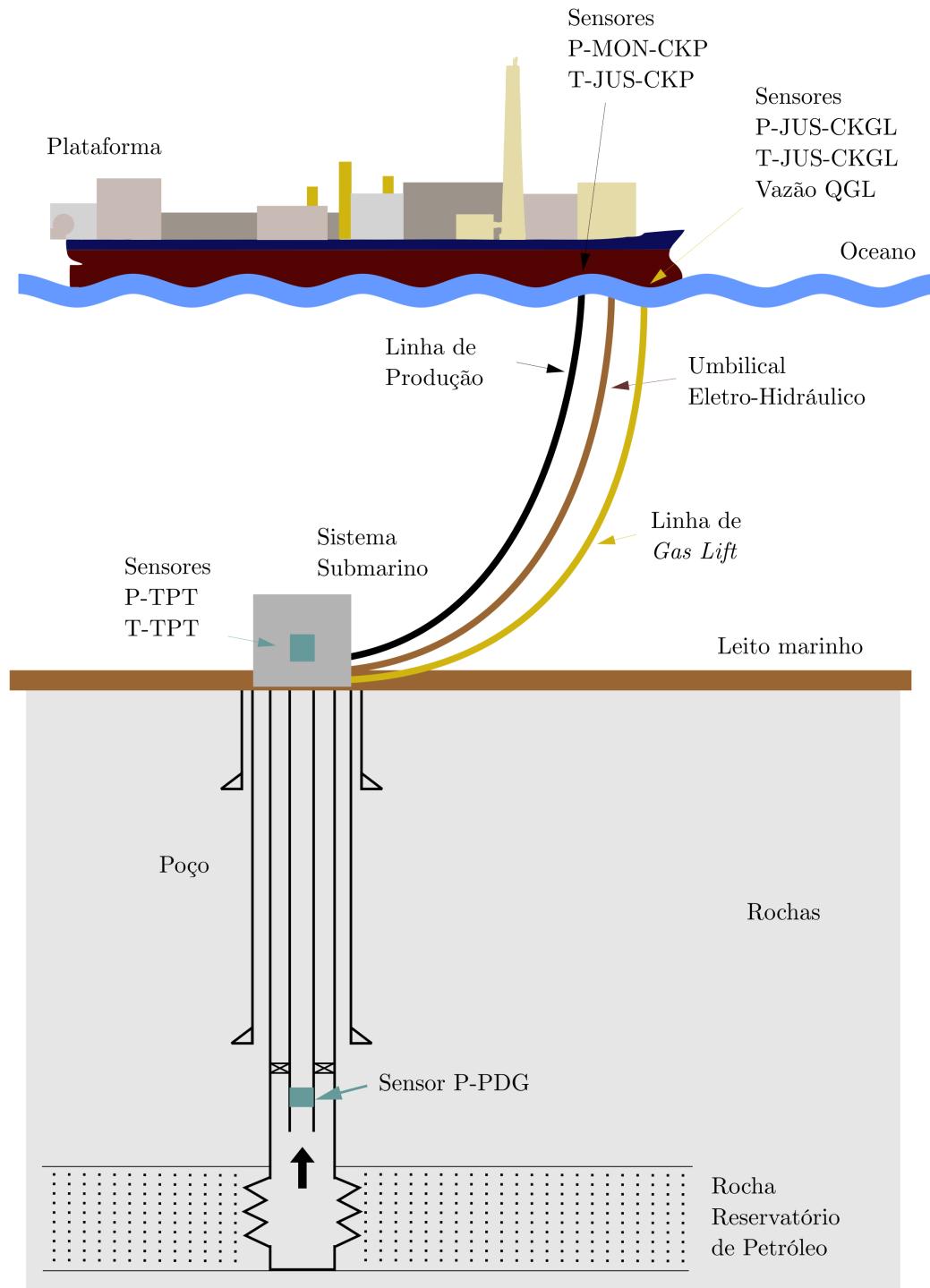
As válvulas instaladas no fundo do mar são operadas remotamente por um umbilical eletro-hidráulico. Existem dispositivos sensores que auxiliam no monitoramento: um manômetro permanente de fundo de poço (P-PDG), um transdutor de temperatura (T-TPT) e um transdutor de pressão (P-TPT).

A DHSV (*Down Hole Safety Valve*) é uma válvula de segurança instalada na coluna de produção de poços. Seu objetivo é garantir o fechamento do poço no caso de uma situação em que a unidade de produção e o poço estejam fisicamente desconectados ou no caso de uma emergência ou falha catastrófica do equipamento de superfície. A válvula CKP (*Choke de Produção*) localiza-se na plataforma e é responsável pelo controle da abertura do poço, possui sensores de temperatura (T-JUS-CKP) e de pressão (P-MON-CKP). A linha de *gas lift* na plataforma tem sensores de vazão (vazão QGL), temperatura (T-JUS-CKGL) e pressão (P-JUS-CKGL).

### 2.2 ANOMALIAS EM POÇOS DE PETRÓLEO

Anomalias em poços de petróleo podem provocar impactos financeiros significativos (VARGAS, 2019). Considerando a produção média por poço no pré-sal de 18 Mbbl/d (milhares de barris por dia) (ANP, 2020) e o preço médio do petróleo no primeiro semestre

Figura 2 – Esquema simplificado de um sistema de produção de petróleo.



Fonte: elaborado pelo próprio autor (2020).

de 2021 de U\$64,95 (MACROTRENDS, 2020), a perda de faturamento em caso de uma anomalia que interrompa a produção de um poço é da ordem de U\$1 milhão de dólares por dia. Adicionalmente, as embarcações que realizam reparos em poços danificados (denominadas sondas) têm custos elevados que chegam a U\$500 mil dólares por dia (ANDREOLLI, 2016).

A base de dados utilizada neste trabalho foi publicada por Vargas et al. (2019) e é intitulada 3W *dataset*. Esse nome foi escolhido devido a base de dados ser composta por instâncias de 3 (três) origens diferentes (reais, simuladas e desenhadas à mão) e conter anomalias que ocorrem em poços surgentes de petróleo (*Wells*, em inglês). Cada instância representa uma condição operacional de um poço e é composta por oito variáveis (oito séries temporais), conforme descrito na Tabela 1, provenientes de sensores de sistemas de produção de petróleo, conforme a localização física aproximada mostrada na Figura 2. Para cada instância, existe uma variável adicional que é um vetor de rótulos no nível de observação que estabelece até três períodos em cada instância de qualquer tipo: normal, transiente de anomalia e estado estável de anomalia.

Tabela 1 – Descrição das variáveis das séries temporais presentes no 3W *dataset*.

| Variável        | Descrição  | Unidade           |
|-----------------|--|-------------------|
| P-PDG           | Pressão do fluido no PDG   | Pa                |
| P-TPT           | Pressão do fluido no TPT   | Pa                |
| T-TPT           | Temperatura do fluido no TPT   | °C                |
| P-MON-CKP       | Pressão do fluido montante à válvula CKP   | Pa                |
| T-JUS-CKP       | Temperatura do fluido jusante à válvula CKP  | °C                |
| P-JUS-CKGL      | Pressão do fluido jusante à válvula de controle de <i>gas lift</i>   | Pa                |
| T-JUS-CKGL      | Temperatura do fluido jusante à válvula de controle de <i>gas lift</i>   | °C                |
| QGL             | Vazão de <i>gas lift</i> .   | m <sup>3</sup> /s |
| Vetor de Rótulo | Valor numérico que indica o estado de cada anomalia ao longo da série temporal: período normal, transiente de anomalia e estado estável de anomalia. | -                 |

Fonte: adaptado de Vargas et al. (2019).

Do total de 1.984 instâncias da base 3W *dataset*, 597 são normais e 1.397 são anomalias. Essas instâncias foram geradas a partir de três diferentes fontes, conforme descrito nos itens abaixo:

- Instâncias reais: são dados históricos, que ocorreram de fato em poços produtores;
- Instâncias simuladas: obtidas com a utilização de simulador dinâmico multifásico no qual os modelos foram calibrados por especialistas na área de petróleo;
- Instâncias desenhadas à mão: digitalizadas a partir de formulários em papel, nos quais especialistas desenharam à mão especificando seus atributos tais como grandeza (variável), escalas, tipo de evento (anomalia), início dos períodos normal, transiente de anomalia e estado estável de anomalia e escalas.

Tabela 2 – Divisão das instâncias por classe e por fonte no 3W dataset

| Classe                            | Real | Simulada | Desenhada | Total |
|-----------------------------------|------|----------|-----------|-------|
| 0 - Normal                        | 597  | -        | -         | 597   |
| 1 - Aumento Abrupto de BSW        | 5    | 14       | 10        | 129   |
| 2 - Fechamento Espúrio de DHSV    | 22   | 16       | -         | 38    |
| 3 - Intermitênciá Severa          | 32   | 74       | -         | 106   |
| 4 - Instabilidade de Fluxo        | 344  | -        | -         | 344   |
| 5 - Perda Rápida de Produtividade | 12   | 439      | -         | 451   |
| 6 - Restrição Rápida em CKP       | 6    | 215      | -         | 221   |
| 7 - Incrustação em CKP            | 4    | -        | 10        | 14    |
| 8 - Hidrato em Linha de Produção  | 3    | 81       | -         | 84    |
| Total                             | 1025 | 939      | 20        | 1984  |

Fonte: Vargas et al. (2019).

A Tabela 2 mostra a quantidade de instâncias por classe e por fonte: instâncias reais, simuladas e desenhadas à mão. Das 1.984 instâncias, 1.025 são reais, 939 simuladas e 20 desenhadas à mão. A base 3W dataset categoriza as anomalias em oito classes, que são descritas a seguir.

1. Aumento Abrupto de BSW (129 instâncias): o *Basic Sediment and Water* (BSW) é definido como a razão entre a produção de água e a produção total (óleo+água) do poço. Um aumento abrupto desse valor pode acarretar em dificuldades de escoamento e elevação, menor produção de óleo e menor fator de recuperação de petróleo.
2. Fechamento Espúrio de DHSV (38 instâncias): o fechamento espúrio dessa válvula causa paradas de produção. DHSV é uma válvula de segurança instalada dentro do poço para assegurar seu fechamento em caso de emergência.
3. Intermitênciá Severa (106 instâncias): condição de instabilidade de escoamento de grande duração, grande amplitude e periodicidade definida que causa redução da produção e danos às instalações.
4. Instabilidade de Fluxo (344 instâncias): comportamento instável de escoamento (também chamado de golfadas) que ocorre com frequência em poços de produção de petróleo e gás. Essa condição pode evoluir para intermitênciá severa.
5. Perda Rápida de Produtividade (451 instâncias): alteração de propriedades do reservatório de petróleo (pressão, razão gás/óleo) ou do fluido produzido (densidade, viscosidade) que dificulta o escoamento do petróleo.
6. Restrição Rápida em CKP (221 instâncias): eventuais restrições rápidas e indesejadas podem ocorrer nessa válvula por problemas operacionais.

7. Incrustação em CKP (14 instâncias): ocorrência de depósitos inorgânicos na válvula CKP, os quais podem reduzir drasticamente a produção de petróleo.
8. Hidrato em Linha de Produção (84 instâncias): hidrato é um composto cristalino sólido formado por água e gás natural (assemelha-se ao gelo comum). É uma anomalia de difícil solução e que pode causar paradas de produção durante dias ou até semanas.

Os diferentes tipos de anomalias também possuem dinâmicas distintas em termos da velocidade de ocorrência, conforme ilustrado na Tabela 3. Por exemplo, para um anomalia da classe “Incrustação em CKP” o tamanho da janela de ocorrência é de até 72 horas, enquanto para a classe “Instabilidade de Fluxo” essa janela é de 15 minutos.

Tabela 3 – Tamanhos de janela de tempo até o alcance do estado estável de anomalia no 3W *dataset*.

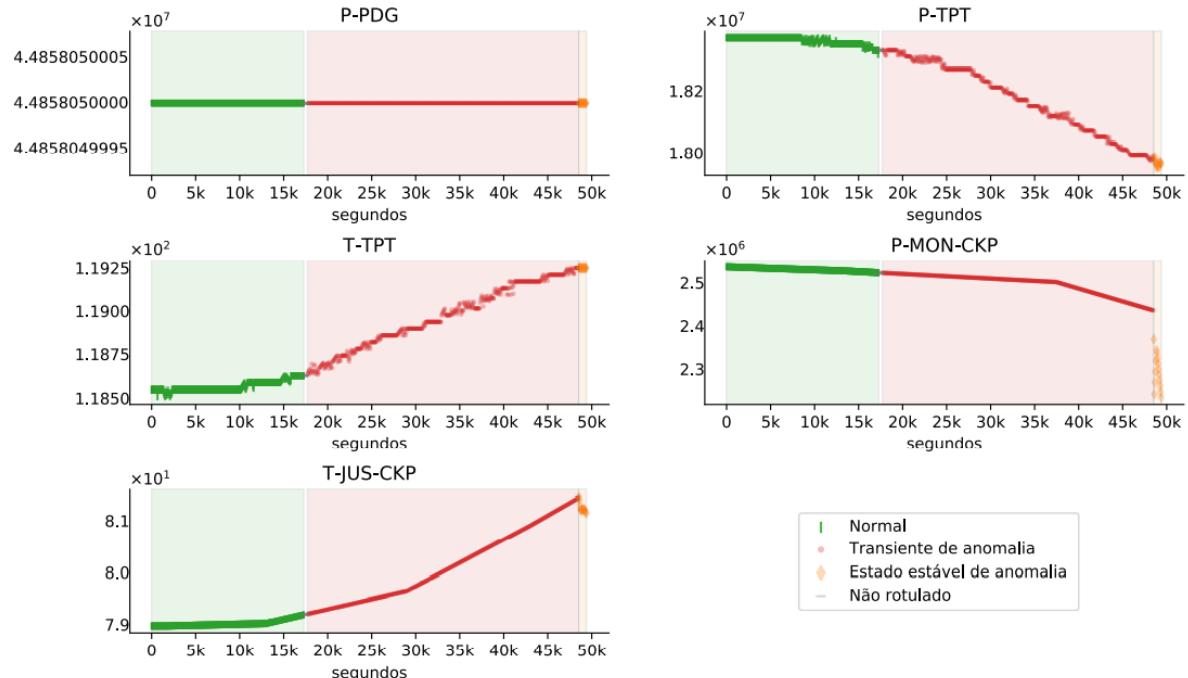
| Tipo de anomalia                  | Tamanho da janela |
|-----------------------------------|-------------------|
| 1 - Aumento Abrupto de BSW        | 12 h              |
| 2 - Fechamento Espúrio de DHSV    | 5 min - 20 min    |
| 3 - Intermittência Severa         | 5 h               |
| 4 - Instabilidade 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      |

Fonte: Vargas et al. (2019).

A Figura 3 apresenta exemplo de instância da classe “Aumento Abrupto de BSW” com os gráficos das variáveis P-PDG, P-TPT, T-TPT, P-MON-CKP e T-JUS-CKP. Nos gráficos é possível verificar que o poço inicialmente estava em operação normal (verde), em seguida mudou para transiente de anomalia (gráficos começam a apresentar alteração de valores) até atingir o estado estável de anomalia (laranja). Nesse exemplo as variáveis atingiram o estado estável de anomalia somente após cerca de 8 horas e 30 minutos desde o início da anormalidade. Também percebe-se também a existência de uma variável congelada (P-PDG) causada possivelmente por problemas operacionais.

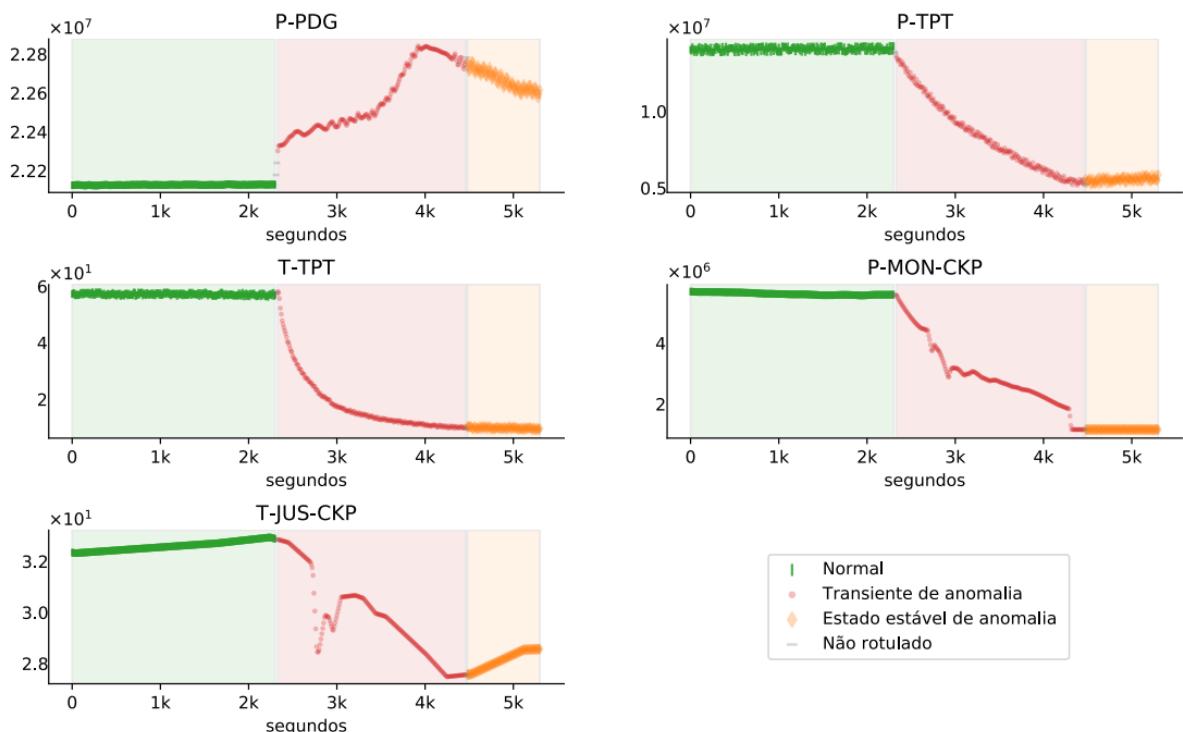
A Figura 4 apresenta exemplo da instância de classe “Fechamento Espúrio de DHSV” com os gráficos das variáveis P-PDG, P-TPT, T-TPT, P-MON-CKP e T-JUS-CKP. É possível verificar que o poço inicialmente estava em operação normal (verde), passando em seguida pelo transiente de anomalia (oscilação dos valores de pressão e temperatura e vazão sensores) até atingir o estado estável de anomalia no (laranja). Observa-se que o poço levou cerca de 35 minutos para entrar estado estável de anomalia (laranja).

Figura 3 – Exemplo de instância da classe “Aumento Abrupto de BSW” da base de dados 3W dataset.



Fonte: reproduzido de Vargas (2019).

Figura 4 – Exemplo de instância da classe “Fechamento Espúrio de DHSV” da base de dados 3W dataset



Fonte: reproduzido de Vargas (2019).

Para a organização da base de dados, Vargas et al. (2019) geraram arquivos específicos e padronizados em formato *Comma-Separated Values* (CSV) para cada instância. Esses arquivos foram agrupados em diretórios baseados na classe da anomalia. Todas as instâncias foram geradas com observações obtidas com taxa de amostragem fixa (1 Hz) e ao nome do arquivo foi incorporada a origem de cada instância (real, simulada ou desenhada).

A base de dados tem 4.947 variáveis ausentes (valores indisponíveis por problemas em sensores ou redes de comunicação ou por inaplicabilidade da variável à instância) que representam 31,17% de todas as 15.872 variáveis de todas as 1.984 instâncias. Também tem 1.535 variáveis congeladas (valores que se mantêm fixos por conta de problemas em sensores ou redes de comunicação) que representam 9,67% de todas as 15.872 variáveis de todas as 1.984 instâncias.

### 2.3 TRABALHOS CORRELATOS

O trabalho de Chandola, Banerjee e Kumar (2009) é um importante artigo *survey* no tema detecção de anomalias. Apresenta contribuições e discussões sobre o conceito de anomalia, seus diferentes aspectos em cada domínio de aplicação, dando uma visão geral estruturada, agrupando técnicas existentes em diferentes categorias, identificando as vantagens e desvantagens de cada uma. Também fornece uma discussão sobre a complexidade computacional das técnicas.

Barbariol, Feltresi e Susto (2019) propuseram uma abordagem de detecção de anomalias em módulos de metrologia de medidores de fluxo multifásicos. Esses equipamentos são importantes ferramentas no setor de petróleo e gás, pois fornecem simultaneamente dados em tempo real dos fluxos de óleo, gás e água. Os algoritmos *Cluster Based Local Outlier Factor* e Floresta de Isolamento foram utilizados para detectar alterações de qualidade nas medições realizadas, tendo sido utilizado um conjuntos de dados semi-sintéticos.

Chan et al. (2019) realizaram detecção de anomalias em controladores lógicos programáveis (CLPs) que compõem sistemas de controle de supervisão e aquisição de dados (SCADA). Esses equipamentos gerenciam operações de equipamentos industriais baseados em sensores e estão expostos a ameaças cibernéticas. Foi realizado um estudo de caso envolvendo uma simulação de semáforo que demonstrou que as anomalias são detectadas com alta precisão utilizando *One-class SVM*.

Khan et al. (2019) aplicaram técnicas de detecção de anomalias em veículos aéreos não tripulados. Foram utilizados dados de uma base denominada *Aero-Propulsion System Simulation* e realizados experimentos em um veículo real. Foram investigados os requisitos para aplicativos de engenharia e demonstrada uma implementação do algoritmo de Floresta de Isolamento.

Tan et al. (2020) compararam o desempenho de diversos classificadores para detecção de anomalias em máquinas de embarcações marítimas. A segurança e a confiabilidade da navegação dependem do desempenho dessas máquinas e o monitoramento inteligente de condições é importante para as atividades de manutenção. Um conjunto de dados de um sistema de propulsão de turbina a gás de um navio foi utilizado. Foi investigado o desempenho de classificadores de classe única: OCSVM, SVDD (*Support Vector Data Description*), GKNN (*Global K-Nearest Neighbors*), LOF, Floresta de Isolamento e ABOD (*Angle-based Outlier Detection*). Em termos de desempenho, os algoritmos ABOD e OCSVM obtiveram os melhores resultados de acurácia.

Grashorn, Hansen e Rummens (2020) descrevem uso de redes neurais para detecção de anomalias na operação do módulo Columbus da Estação Espacial Internacional. Trata-se de um laboratório científico que transmite para a Terra cerca de 17.000 parâmetros de telemetria por segundo. A equipe de operações do *Columbus Control Center*, em colaboração com a Airbus, acompanha esses parâmetros e utiliza algoritmos do tipo *autoencoders* com células do tipo LSTM para apoiar na detecção de anomalias durante o fluxo de trabalho do centro de controle.

Elsayed et al. (2020) utilizaram uma combinação de estrutura de rede neural *autoencoder* com células do tipo LSTM, juntamente com o algoritmo OCSVM, para modelagem do fluxo de dados normais em uma rede computacional. Os experimentos mostraram que o modelo proposto pode detectar com eficiência as anomalias apresentadas nos dados de tráfego da rede.

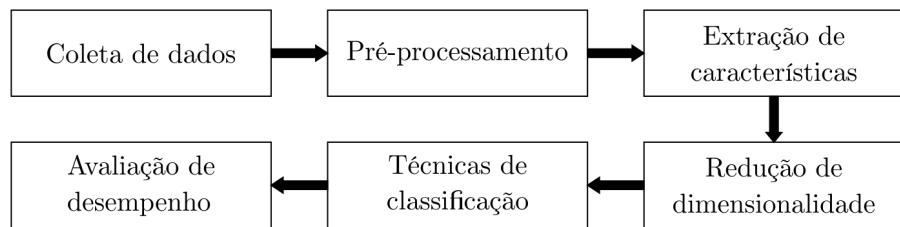
Os trabalhos de Vargas et al. (2019) e Vargas (2019) foram a base deste trabalho. Eles elaboraram e tornaram pública a base 3W gerada com instâncias provenientes de três fontes: reais, simuladas e desenhadas à mão. Vargas (2019) também elaborou dois *benchmarks* específicos que podem ser utilizados para permitir que algoritmos propostos por diferentes pesquisadores tenham seus desempenhos avaliados e comparados, sendo um para avaliação do impacto do uso de instâncias simuladas e desenhadas à mão e outro para detecção de anomalias. No *benchmark* para detecção de anomalias foram usadas as técnicas de Floresta de Isolamento e OCSVM.

O presente trabalho utilizou o *benchmark* para detecção de anomalias proposto por Vargas (2019), e estendeu seus resultados pois usou mais classificadores (LOF, Envelope Elíptico, Floresta de Isolamento, OCSVM e Redes Neurais do tipo *autoencoder*) e também incluiu a etapa de calibração dos hiperparâmetros.

### 3 METODOLOGIA

Neste capítulo apresentam-se os métodos empregados na proposta deste trabalho. Um processo de classificação, conforme ilustrado na Figura 5, pode ser dividido em seis etapas (KADHIM, 2019): coleta de dados, pré-processamento, extração de características, redução de dimensionalidade, aplicação da técnica de classificação e avaliação de desempenho.

Figura 5 – Etapas do processo de classificação.



Fonte: adaptado de Kadhim (2019).

A coleta de dados é a fase inicial, na qual é realizado o levantamento dos dados a serem utilizados, que neste caso é a base 3W. As subseções a seguir descrevem cada uma das etapas após a coleta de dados.

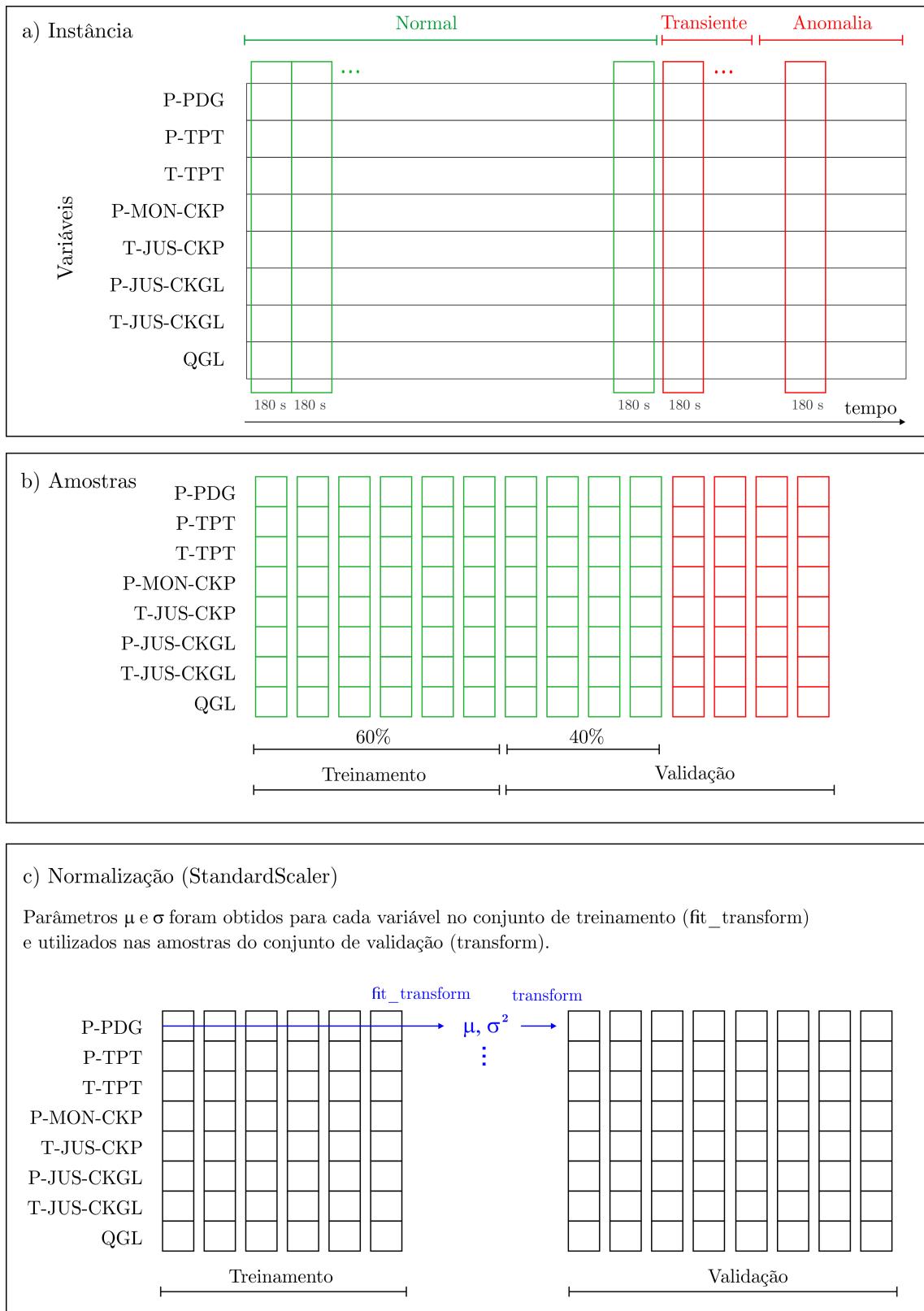
#### 3.1 PRÉ-PROCESSAMENTO

A etapa de pré-processamento trata da preparação inicial do conjunto de dados coletado. No caso de séries temporais, nessa etapa, inclui-se a análise dos dados, geração de gráficos para entendimento do dados, remoção de valores nulos e/ou congelados e re-amostragens de observações das séries temporais para balanceamento da base de dados (PAL; PRAKASH, 2017).

Na etapa de pré-processamento foi realizada amostragem das instâncias com janela deslizante com geração de até 15 amostras com 180 observações cada, conforme ilustrado na Figura 6.a. Dos períodos normais, as primeiras observações foram utilizadas para treinamento (60%) e as últimas, para validação (40%). Dos períodos rotulados como anomalias, as observações foram utilizadas apenas para validação, conforme ilustrado na Figura 6.b.

As variáveis das amostras utilizadas que tinham quantidades de valores ausentes acima de um limiar (10%) ou que tinham desvios padrões abaixo de outro limiar (1%) foram totalmente descartadas. As variáveis das amostras de treinamento foram normalizadas utilizando a média e desvio padrão (utilizando o *StandardScaler*) conforme ilustrado na

Figura 6 – Representação da etapa de pré-processamento.



Fonte: elaborado pelo próprio autor (2020).

Figura 6.c e na Equação 1.

$$z = \frac{x - \mu}{\sigma} \quad (1)$$

Os valores de média ( $\mu$ ) e desvio padrão ( $\sigma$ ) foram calculados e, posteriormente, utilizados para normalizar os dados de validação de cada instância individualmente.

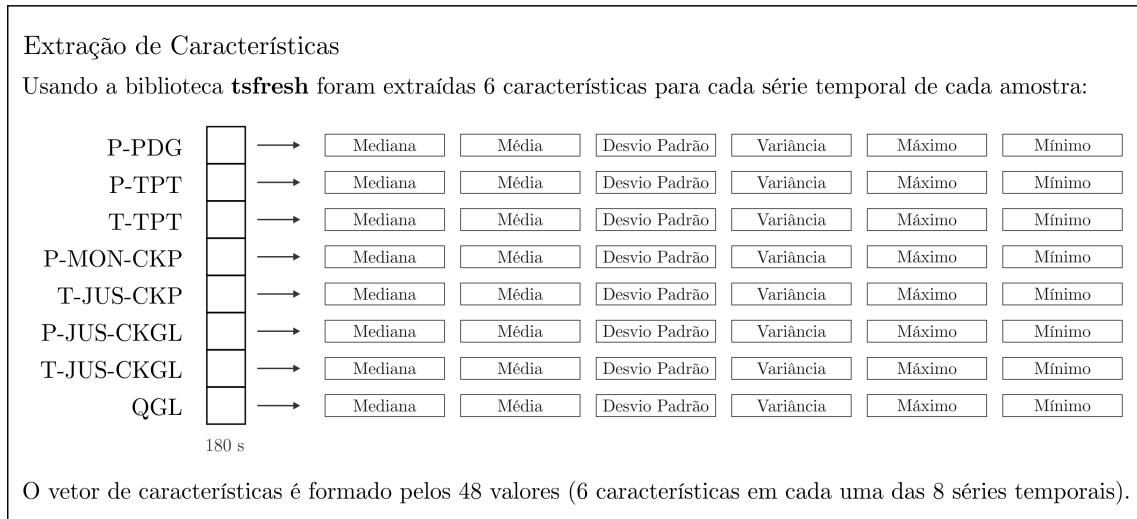
### 3.2 EXTRAÇÃO DE CARACTERÍSTICAS

Na etapa de extração de características, os dados pré-processados são trabalhados para obter as características mais relevantes para a classificação.

Uma série temporal univariada  $x = [x_1, x_2, \dots, x_T]$  é um conjunto ordenado de valores reais. O comprimento de  $x$  é igual ao número de valores reais  $T$ . A série temporal é multivariável  $X = [X^1, X^2, \dots, X^M]$  quando consiste em  $M$  séries temporais univariadas diferentes com  $X^i \in \mathbb{R}^T$  (FAWAZ et al., 2019).

A partir de cada amostra de série temporal, foram extraídas e utilizadas como características a mediana, média, desvio padrão, variância, máximo e mínimo para cada variável. Para esta extração de características das séries temporais foi utilizada a biblioteca *tsfresh*<sup>1</sup> (*Time series feature extraction*) (CHRIST et al., 2018) na configuração de parâmetros mínimos, de forma a ser possível reproduzir os resultados iniciais gerados por Vargas (2019).

Figura 7 – Representação do processo de extração de características.



Fonte: elaborado pelo próprio autor (2020).

### 3.3 REDUÇÃO DE DIMENSIONALIDADE

Após a extração de características, é possível que se tenha uma grande quantidade de características, e pode ser necessário reduzir os custos de processamento. Esse processo é denominado de redução de dimensionalidade. Esta etapa não existiu na abordagem proposta deste trabalho.

<sup>1</sup> <https://tsfresh.readthedocs.io>

As técnicas mais comuns de redução de dimensionalidade são PCA (*Principal Component Analysis*), LDA (*Linear Discriminant Analysis*) e NMF (*Non-negative matrix factorization*) (KOWSARI et al., 2019). O *Self-Organizing Map* (SOM) é um tipo específico de rede neural, proposto por Kohonen (2013), que também é utilizado como uma ferramenta de redução de dimensionalidade para extração de características em classificação de dados de alta dimensionalidade, sendo considerado como uma alternativa ao PCA na detecção e diagnóstico de falhas para processos industriais complexos (YU et al., 2014).

### 3.4 CLASSIFICAÇÃO

Classificadores de classe única podem ser utilizados na detecção de padrões raros. Em geral, utiliza-se apenas a classe comum (normalidade) no treinamento (*novelty detection*) e nos testes há uma mistura de instâncias normais e anormais (KHAN; MADDEN, 2014).

Por serem citadas como referências em trabalhos correlatos, foram escolhidos os seguintes classificadores para realização dos experimentos: OCSVM, Floresta de Isolamento, LOF, Envelope Elíptico e redes neurais (*autoencoder feedforward* e *autoencoder LSTM*). Para realização dos experimentos foram utilizados os algoritmos implementados nas bibliotecas *Scikit-Learn*<sup>2</sup> desenvolvida por Pedregosa et al. (2011a) e *Tensorflow*<sup>3</sup> desenvolvida por Abadi et al. (2015).

#### 3.4.1 One-class Support Vector Machine (OCSVM)

Baseado em otimização, o SVM constrói hiperplanos para separar diferentes classes no espaço. O objetivo é maximizar o parâmetro  $b$  (chamado de margem) que representa a distância entre o hiperplano e o ponto mais próximo de cada classe, de forma a criar um limiar de decisão (DUDA; HART; STORK, 2012).

Existem versões alternativas que utilizam funções matemáticas não lineares (chamadas *kernels*) que mapeiam o espaço de características em outro de maior dimensionalidade no qual a separabilidade entre as classes tende a ser maior (GÉRON, 2019).

O SVM de classe única foi introduzido por Schölkopf et al. (2001) e é ilustrado na Figura 8. Tem como objetivo separar o conjunto de dados da origem e construir uma hiperesfera que engloba todas as instâncias normais em um espaço. Uma nova instância é classificada como uma anomalia quando não se enquadra no espaço dessa hiperesfera (MISRA; LI; HE, 2020).

A principal diferença entre o SVM padrão e o SVM de classe única é que o OCSVM fornece um hiperparâmetro  $\nu$  ( $\eta_i$ ), que é usado para controlar a sensibilidade dos vetores

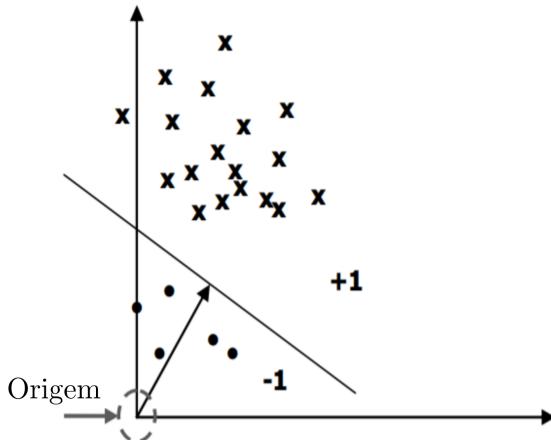
---

<sup>2</sup> <https://scikit-learn.org/>

<sup>3</sup> <https://www.tensorflow.org/>

de suporte, em vez dos hiperparâmetros normais como  $C$  no SVM padrão, que é usado para ajustar a margem.

Figura 8 – *One-class SVM* (OCSVM): tem como objetivo separar o conjunto de dados da origem e construir uma hiperesfera que engloba todas as instâncias normais em um espaço.



Fonte: imagem GNU Free Licence.

É possível ajustar os hiperparâmetros de *kernel* (linear, polinomial, radial),  $\gamma$  (gama) e  $\nu$  (ni). O parâmetro  $\gamma$  influencia o raio da hiperesfera gaussiana que separa as instâncias normais das anomalias - grandes valores de gama resultam em uma hiperesfera menor e em um modelo “mais rígido” que encontra mais discrepâncias. A fração  $\nu$  define a porcentagem do conjunto de dados que é discrepante e ajuda a criar limites de decisão mais rígidos (MISRA; LI; HE, 2020).

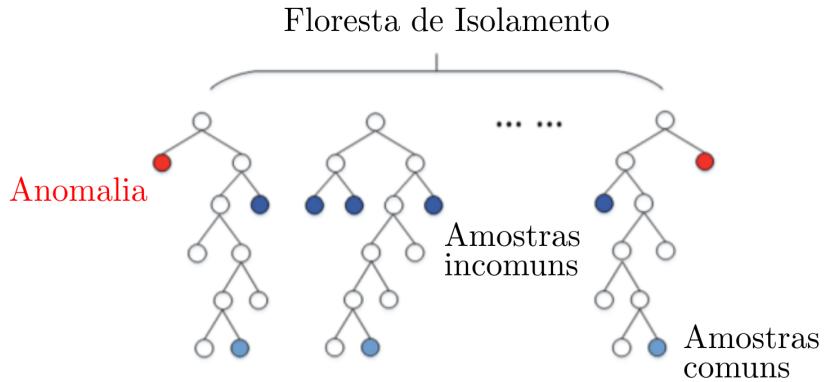
### 3.4.2 Floresta de Isolamento

Baseadas em busca, árvores de decisão são construídas com base em regras inferidas a partir dos atributos. Embora não tenham sido projetadas originalmente para o problema de detecção de anomalias, é possível sua utilização por meio da análise dos caminhos percorridos na árvore durante a divisão em cada nó (AGGARWAL, 2016).

Liu, Ting e Zhou (2012) denominaram árvore de isolamento (ou *isolation tree* - IT) quando em cada nó um atributo é selecionado aleatoriamente e, em seguida, divide-se o conjunto de dados em dois a partir de um valor limite aleatório (entre os valores mínimo e máximo). O conjunto de dados é gradualmente dividido até que todas as instâncias sejam isoladas (GÉRON, 2019). Como anomalias em geral estão em regiões menos populosas do conjunto de dados, geralmente, são necessárias menos partições aleatórias para isolá-las em nós da árvore (MISRA; LI; HE, 2020).

Uma técnica é chamada de *ensemble* quando um conjunto de classificadores é treinado individualmente, sendo que cada modelo é exposto a um subconjunto diferente de dados,

Figura 9 – Floresta de Isolamento: as anomalias em geral estão em regiões menos populosas e geralmente são necessárias menos partições aleatórias para isolá-las em nós da árvore.



Fonte: adaptado de Chen et al. (2016).

mas as decisões são tomadas de forma combinada para se ter uma resposta única. Métodos *ensemble* tendem a apresentar um menor *overfitting* (AGGARWAL; SATHE, 2017), que é a dificuldade de generalizar nos dados de teste os resultados obtidos pelos modelos nos dados de treinamento. O método *ensemble* Floresta de Isolamento (ou *Isolation Forest - IF*) busca criar uma estrutura de árvores aleatórias para isolar as anomalias das instâncias. Conforme ilustrado na Figura 9, as anomalias tem maior suscetibilidade ao isolamento e ficam mais perto das raízes das árvores, enquanto os pontos normais são difíceis de isolar e geralmente estão no extremo mais profundo da árvore. Os comprimentos médios de caminho em várias árvores são utilizados para obter uma pontuação e classificar a instância (CHEN et al., 2016).

Desse modo, uma Floresta de Isolamento (IF) é então definida por inúmeras árvores de isolamento (IT):

$$IF = \{IT_1, IT_2, \dots, IT_t\}$$

onde, para cada árvore  $IT_t$ , é possível calcular o número de iterações  $h_t(x)$  necessárias para isolar uma amostra  $x$ . O número médio de etapas necessárias para isolar uma amostra  $x$  em uma floresta é representado na Equação 2:

$$h(x) = \frac{1}{T} \sum h_t(x) \quad (2)$$

onde  $T$  é o total de árvores da floresta e  $h_t(x)$  o número de iterações necessárias para isolar uma amostra  $x$  na árvore  $IT_t$ .

Quanto menos passos para isolar uma anomalia, melhor (FILHO et al., 2020). O número

de etapas necessárias para isolar uma observação  $x$  é influenciado pelo tamanho da amostragem  $n$ . Uma pontuação de anomalia normalizado  $s(x, n)$  é definido como:

$$s(x, n) = 2^{-\frac{h(x)}{c(n)}} \quad (3)$$

onde  $c(n)$  representa um fator de normalização baseado na média da altura da árvore:

$$c(n) = \begin{cases} 2H(n - 1) - (2(n - 1)/n) & \text{se } n > 2 \\ 1 & \text{se } n = 2 \\ 0 & \text{se } n < 2 \end{cases} \quad (4)$$

e  $H(i)$  é o número harmônico estimado por  $\ln(i) + 0.577216649$  (constante de Euler).

Utilizando a técnica de Floresta de Isolamento, implementada na biblioteca Scikit-learn (PEDREGOSA et al., 2011b), é possível experimentar com os seguintes hiperparâmetros: número de estimadores (quantidade de árvores), número máximo de amostras utilizadas por árvore, número de características utilizadas por cada árvore e contaminação no conjunto de dados (estimativa de *outliers*) (MISRA; LI; HE, 2020).

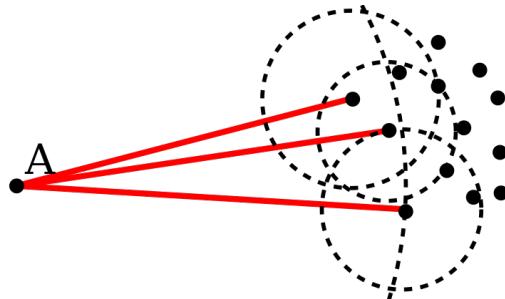
### 3.4.3 Local Outlier Factor (LOF)

Baseado em densidade, o LOF foi desenvolvido por Breunig et al. (2000) e compara a densidade de instâncias em torno de uma determinada instância com a densidade em torno de seus vizinhos. Uma anomalia geralmente é mais isolada que seus vizinhos mais próximos.

O algoritmo calcula a pontuação LOF de uma observação como a razão entre a densidade local média de seus  $k$  vizinhos mais próximos e sua própria densidade local. Ao comparar a densidade local de uma amostra com as densidades locais de seus vizinhos, pode-se identificar amostras que possuem uma densidade substancialmente menor do que seus vizinhos. São consideradas como instâncias normais as que têm densidade local semelhante a de seus vizinhos, enquanto são consideradas anomalias as que têm densidade local menor (MISRA; LI; HE, 2020), conforme ilustração da Figura 10.

No LOF é possível ajustar os hiperparâmetros: número  $k$  de vizinhos a serem considerados, tamanhos de folha do algoritmo, métrica de distância e contaminação no conjunto de dados. O parâmetro  $novelty = True$  permite que a técnica seja aplicada em novos dados.

Figura 10 – *Local Outlier Factor* (LOF): comparação da densidade local de um ponto com as densidades dos vizinhos. O elemento 'A' tem uma densidade muito menor do que seus vizinhos.



Fonte: imagem com permissão de uso de domínio público.

### 3.4.4 Envelope Elíptico

O Envelope Elíptico implementa a técnica *Minimum Covariance Determinant* (MCD) que assume que as instâncias normais são geradas a partir de uma única distribuição gaussiana. Essa distribuição fornece uma estimativa de envelope elíptico a partir do qual é possível a identificação das anomalias (GÉRON, 2019).

As Equações 5 e 6 representam os casos uni e multivariados da distribuição gaussiana, respectivamente, onde  $\mu$  é a média,  $\sigma^2$  é a variância e  $\Sigma$  é a matriz de covariância (BRANCO, 2017).

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x - \mu)^2}{2\sigma^2}\right) \quad (5)$$

$$f(x) = \frac{1}{\sqrt{(2\pi)^n |\Sigma|}} \exp\left(-\frac{1}{2}(x - \mu)^T \Sigma^{-1} (x - \mu)\right) \quad (6)$$

A distância de cada observação deve ser calculada em relação a alguma medida de centralização dos dados, sendo considerada uma anomalia a observação cuja distância seja maior que algum valor predeterminado (HARDIN; ROCKE, 2004).

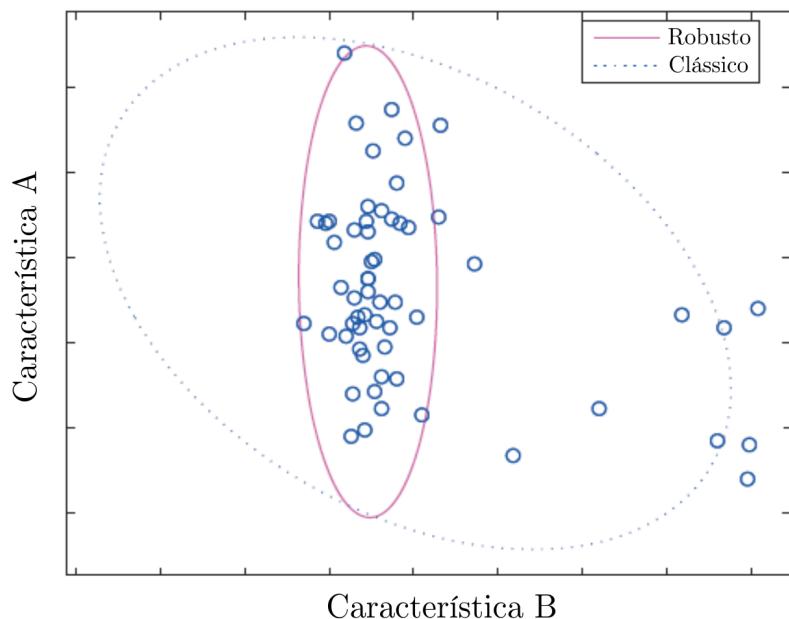
A distância de uma observação em relação à distribuição pode ser calculada usando a distância de Mahalanobis, mostrada na Equação 7.

$$MD(x) = \sqrt{(x - \bar{x})^T S^{-1} (x - \bar{x})} \quad (7)$$

onde  $\bar{x}$  e  $S$  são as médias e a matriz de covariância das amostras.

Estimativas como a distância de Mahalanobis são consideradas parte da estatística clássica e são muito afetadas por valores discrepantes (chamado efeito de mascaramento). Para obter melhores resultados foi proposto por Rousseeuw e Driessen (1999) a utilização de estimadores robustos, cuja característica é serem menos influenciados por desvios provocados pela presença de valores discrepantes no conjunto de dados. A Figura 11 ilustra dois envelopes elípticos gerados com a utilização de estatística clássica e robusta (HUBERT; DEBRUYNE, 2010). É possível verificar que a abordagem robusta tem uma área menor e engloba os elementos em espaço de maior densidade.

Figura 11 – Envelopes elípticos gerados com estatística clássica e estatística robusta, em que todos os elementos internos à figura geométrica são normais e todos os elementos externos à elipse são anormalidades



Fonte: adaptado de Hubert e Debruyne (2010).

A Equação 8 ilustra o estimador de distância robusto cujo cálculo minimiza o determinante da matriz de covariâncias. Esse método utiliza um subconjunto de observações  $h < n$  que permite a obtenção de uma estimativa resistente a *outliers* no conjunto de dados (HUBERT; DEBRUYNE, 2010). A seleção do valor de  $h$  é feita conforme o algoritmo “fastmcd” proposto por Rousseeuw e Driessen (1999), que encontra o valor que minimiza o determinante da matriz de covariância.

$$RD(x) = \sqrt{(x - \hat{\mu}_{MCD})^T \hat{\Sigma}_{MCD}^{-1} (x - \hat{\mu}_{MCD})} \quad (8)$$

onde  $\hat{\mu}_{MCD}$  e  $\hat{\Sigma}_{MCD}$  representam a média e matriz de covariância que minimizam o

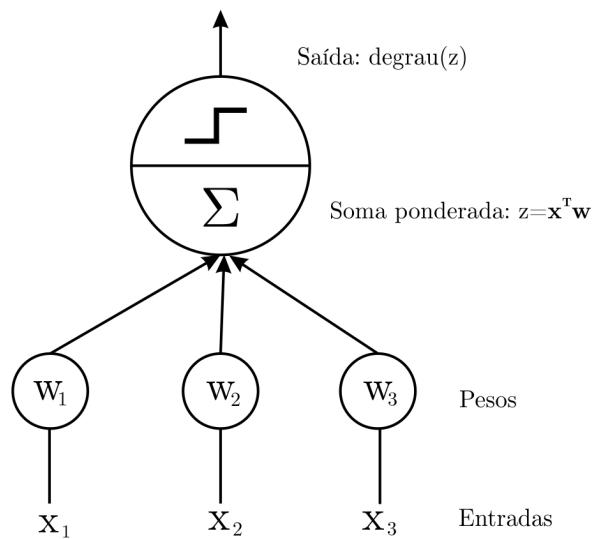
determinante da matriz de covariância.

Por serem menos influenciados pelo efeito de mascaramento, os estimadores MCD podem ser utilizados para detecção de anomalias. O algoritmo implementado na biblioteca Scikit-Learn por Pedregosa et al. (2011b) permite o ajuste de parâmetros que influenciam no cálculo da distância robusta com MCD: centralização, fração de suporte e contaminação no conjunto de dados.

### 3.4.5 Redes Neurais Artificiais

A origem das redes neurais artificiais remonta à década de 40, quando McCulloch e Pitts (1943) propuseram o primeiro modelo de redes neurais utilizando lógica proposicional (valores binários) como entradas e saídas. Na década de 50 uma arquitetura denominada Perceptron foi inventada por Rosenblatt (1958) que permitia a utilização de números e na qual cada conexão é associada a pesos. Conforme ilustrado na Figura 12, essas entradas passavam por uma unidade lógica de gatilho (ou *threshold logic unit*, TLU), que computava uma soma ponderada pelos pesos e então aplicava um função degrau para fornecer uma saída. O Perceptron é alimentado com um exemplo por vez e, para cada instância com resposta incorreta, ele reforça os pesos de conexão das entradas que teriam contribuído para a previsão correta (GÉRON, 2019).

Figura 12 – Unidade básica da rede neural Perceptron.



Fonte: adaptado de Géron (2019).

Marvin e Seymour (1969) destacaram limitações do Perceptron, entre elas sua separabilidade linear que impedia o aprendizado de padrões complexos. Era sabido que as redes neurais multicamadas não possuíam essa restrição de separabilidade linear, porém, somente na

década de 80 Rumelhart, Hinton e Williams (1986) demonstraram com experimentos computacionais que a aplicação de uma técnica denominada *backpropagation* seria uma solução para treinamento de redes multicamadas.

A rede neural é denominada *feedforward* quando cada camada recebe entrada apenas da camada anterior e fornece uma entrada apenas para a camada subsequente. Quando as redes neurais são estendidas para incluir conexões de retroalimentação, as redes neurais são chamadas de recorrentes. Dado um par entrada-saída  $(x, y), x \in X, y \in Y$ , o objetivo é aprender as relações entre as entradas e saídas usando as camadas ocultas (GOODFELLOW; BENGIO; COURVILLE, 2016).

As representações de redes neurais profundas podem ter diferentes arquiteturas. Ao longo das décadas foram desenvolvidas diversas outras arquiteturas especializadas tais como redes *autoencoder*, redes convolucionais e redes recorrentes (RANJAN, 2020).

O uso de técnicas baseadas em redes neurais de múltiplas camadas (também chamado *deep learning* ou aprendizado profundo) para classificação de séries temporais é considerado uma solução interessante, mas desafiadora na área de mineração de dados (FAWAZ et al., 2019).

#### 3.4.5.1 Autoencoder

Um *autoencoder* é uma estrutura de rede na qual o número de nós na camada de saída é o mesmo que o da camada de entrada e a arquitetura é simétrica. A Figura 13 ilustra um autoencoder totalmente conectado.

Para detecção de anomalias a técnica de autoencoder é referenciada por Chen et al. (2017) e Elsayed et al. (2020). Um dos primeiros estudos que envolveu o autoencoder para detecção de anomalias foi proposto por Hawkins et al. (2002).

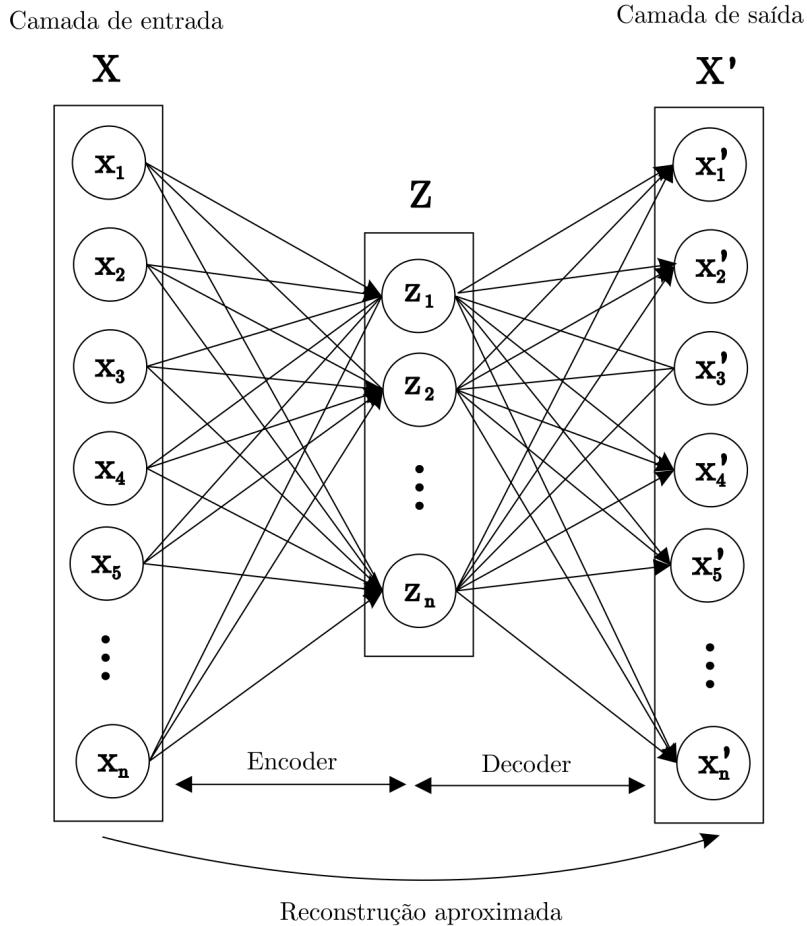
O objetivo é treinar a saída para reconstruir a entrada o mais próximo possível. Os nós nas camadas intermediárias são menores em número e, portanto, a única maneira de reconstruir a entrada é aprender pesos para que as saídas intermediárias dos nós nas camadas intermediárias sejam representações reduzidas dos dados de entrada.

Na etapa de codificação (*encoder*) é realizada a redução das dimensões dos dados da entrada  $X$  de acordo com a Equação 9a.

$$Z = \sigma(WX + b) \tag{9a}$$

$$X' = \sigma'(W'Z + b') \tag{9b}$$

Figura 13 – Exemplo de uma rede neural do tipo Autoencoder.



Fonte: adaptado de Kwon et al. (2019) e Elsayed et al. (2020)

onde  $Z$  é a dimensão reduzida ou latente,  $\sigma$  é a função de ativação,  $W$  é a matriz de pesos e  $b$  é o vetor de *bias*. Da mesma maneira, na etapa de decodificação (*decoder*), treinada de acordo com a Equação 9b, os pesos são calibrados para que os dados de saída sejam o mais semelhante possível aos dados originais.

O objetivo principal desse conceito é que ambos, *encoder* e *decoder*, sejam treinados juntos e que a discrepância entre os dados originais e sua reconstrução, com base em alguma função de custo, seja minimizada (KWON et al., 2019). Como exemplo é ilustrada na Equação 10 a função de erro absoluto médio (ou MAE, *mean absolute error*, em inglês).

$$MAE = \frac{\sum_{i=1}^n |x'_i - x_i|}{n} \quad (10)$$

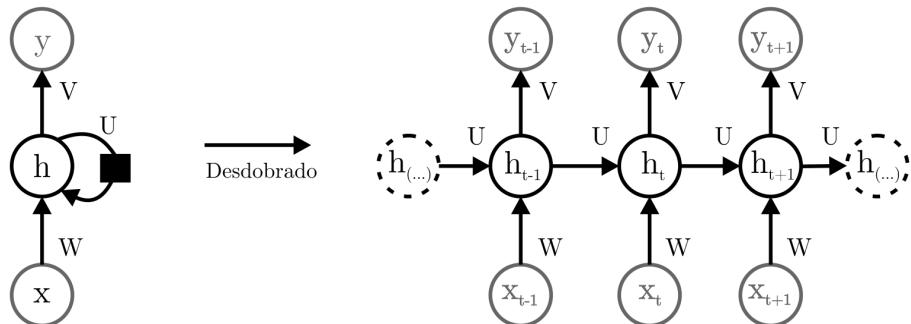
onde  $x_i$  representa a entrada,  $x'_i$  a representação reconstruída da entrada e  $n$  a dimensão do vetor  $x$ .

A ideia básica desse modelo é que as anomalias serão mais difíceis de serem reconstruídas que as condições de normalidade, e em consequência apresentarão maiores erros de construção ao serem submetidas à rede neural.

### 3.4.5.2 Long short-term memory (LSTM)

Quando trabalha-se com dados sequenciais as observações anteriores podem ter um efeito nas observações futuras. Existe uma classe de rede neural chamada rede neural recorrente (RNN), desenvolvida inicialmente por Rumelhart, Hinton e Williams (1986), que permite a utilização dessa memória sequencial para classificar padrões temporais, pois os dados da saída são realimentados em suas entradas. A Figura 14 ilustra a estrutura básica de uma rede neural recorrente, com destaque para o nó  $h$  no qual ocorre a realimentação e sua representação desdobrada. A ideia básica dessa realimentação é permitir o compartilhamento de parâmetros (GOODFELLOW; BENGIO; COURVILLE, 2016).

Figura 14 – Estrutura básica de uma rede neural recorrente (RNN).



Fonte: adaptado de Goodfellow, Bengio e Courville (2016).

A Equação 11 representa as expressões para uma sequência de valores  $x$  onde a variável  $h$  representa o estado das unidades ocultas da rede,  $b$  e  $c$  são os vetores de *bias* e  $W$ ,  $V$  e  $U$ , as matrizes de peso da entrada para a camada oculta, camada oculta para saída e camada oculta para camada oculta, respectivamente. Diferentes tipos de funções de ativação são possíveis para as camadas camadas ocultas e de saída, sendo que na Equação foram representadas as funções  $\tanh$  e  $\text{softmax}$  para a camada oculta e de saída, respectivamente.

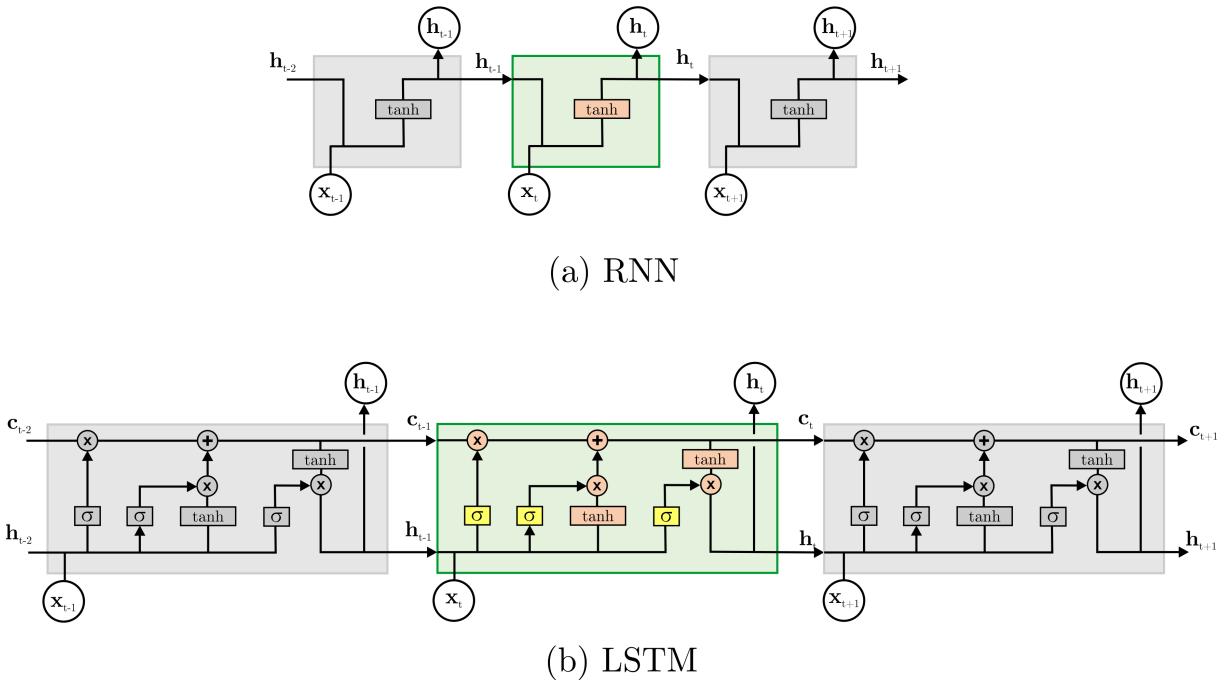
$$h_t = \tanh(Wx_t + Uh_{t-1} + b) \quad (11a)$$

$$y_t = \text{softmax}(Vh_t + c) \quad (11b)$$

As RNN são uma classe poderosa de modelos computacionais capazes de aprender dinâmicas arbitrárias, pois conseguem manter essa memória de padrões em ordens sequenciais, porém têm como principal limitação a incapacidade de aprender memórias de longo prazo devido ao problema de *vanishing gradient*. O gradiente é usado para atualizar os valores de peso do modelo aprendido. No entanto, caso o gradiente seja muito pequeno (*vanishing gradient*), o modelo não aprende com eficiência. Esse problema ocorre nas redes RNN, pois as operações de *backpropagation* são realizadas sobre as entradas e unidades recorrentes  $h$ .

Para minimizar essa problema foi desenvolvida a rede *Long short-term memory* (LSTM), originalmente descrita em Hochreiter e Schmidhuber (1997), que introduziram o conceito de estados de células que guardam as memórias de longo e de curto prazo. A Figura 15 ilustra o fluxo de informações ao longo das redes RNN e LSTM.

Figura 15 – Fluxo de informações nas estruturas internas das redes RNN e LSTM.

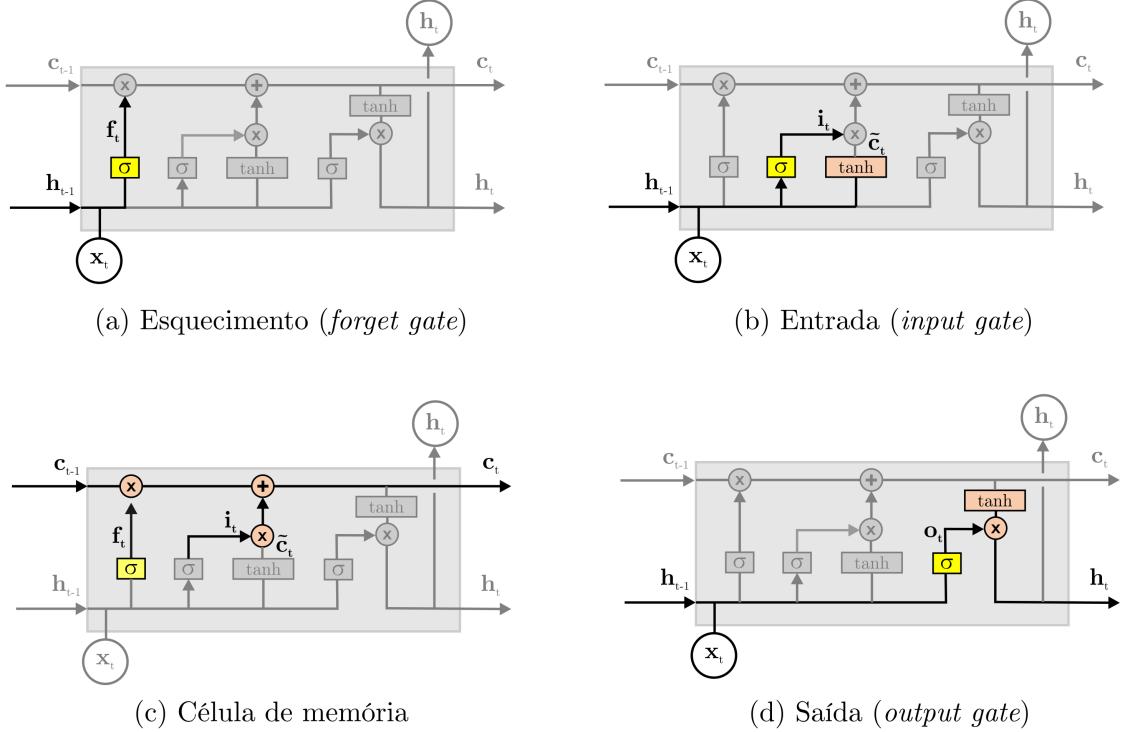


Fonte: adaptado de Ranjan (2020)

A LSTM é um tipo especial de rede neural recorrente (RNN) que é estável, poderosa o suficiente para ser capaz de modelar dependências de tempo de longo alcance e superar o problema do *vanishing gradient*. Sua célula adicional de estado permite que os pesos possam fluir por longos períodos de tempo sem tenderem a zero (sem efeito de *vanishing gradient*). Um mecanismo de portas é utilizado para regular o fluxo de informações conforme ilustrado na Figura 16.

A célula consiste em três portas, entrada (i), saída (o) e esquecimento (f), com ativação sigmóide ( $\sigma$ ) mostrada nas caixas amarelas. A célula obtém informações relevantes por

Figura 16 – Detalhamento das partes presentes nas estruturas internas da célula LSTM.



Fonte: traduzido e adaptado de Ranjan (2020)

meio da funções de ativações tanh mostradas em caixas laranja. A célula pega os estados anteriores ( $c_{t-1}, h_{t-1}$ ), executa-os através das portas e extrai informações para produzir o estado atualizado ( $c_t, h_t$ ) (RANJAN, 2020). As expressões matemáticas envolvidas estão descritas na Equação 12.

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \quad (12a)$$

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \quad (12b)$$

$$\tilde{c}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c) \quad (12c)$$

$$c_t = f_t c_{t-1} + i_t \tilde{c}_t \quad (12d)$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \quad (12e)$$

$$h_t = o_t \tanh(c_t) \quad (12f)$$

onde:  $i_t$ ,  $o_t$ , e  $f_t$  são portas de entrada, saída e esquecimento;  $\sigma$  é a função sigmóide;  $\tilde{c}_t$  é uma variável temporária que contém as informações relevantes no *timestep* atual  $t$ ;  $c_t$  e  $h_t$  são o estado da célula e as saídas; e  $W_{i,o,f,c}$ ,  $U_{i,o,f,c}$  e  $b_{i,o,f,c}$  são os parâmetros de pesos e *bias* das portas de entrada (i), saída (o), esquecimento (f) e memória da célula (c), respectivamente.

Inicialmente, a porta de esquecimento (*forget gate*), calculada conforme a Equação 12a e representada na Figura 16a, determina quais dados serão descartados e quais serão mantidos do estado da célula. Por meio de uma função de ativação sigmóide  $\sigma$  o *forget gate* mantém uma fração das informações de estados anteriores (GREFF et al., 2016).

Paralelamente, conforme ilustrado na Figura 16b, a decisão de quais dados serão armazenados no estado da célula é feita em duas etapas: pela porta de entrada (*input gate*), que é responsável por obter dados do *timestep* atual e do estado anterior, e pela determinação do valor ( $\tilde{c}_t$ ), que avalia os valores a serem acrescentados ao estado (candidatos a células de estado). As expressões estão representadas nas Equações 12b e 12c.

Em seguida é realizada a atualização da célula de memória (estado atual da célula), ilustrada na Figura 16c. Esse cálculo é feito com base no estado da célula anterior, no *forget gate*, no *input gate* e no candidato a célula de estado ( $\tilde{c}_t$ ), conforme a Equação 12d.

Por fim, é calculada a saída  $o_t$  (*output gate*) da célula e o novo estado oculto  $h_t$ , de acordo com as Equações 12e e 12f e conforme ilustrado na Figura 16d.

Os algoritmos implementados na biblioteca Tensorflow desenvolvida por Abadi et al. (2015) permitem a realização de experimentos com diversas variações de parâmetros nas redes neurais, tais como: tipo, número e tamanho das camadas; funções de ativação, parâmetros de inicialização das matrizes de pesos, taxa de aprendizado, entre outros.

O *autoencoder* também pode ser usado como ferramenta para extração de características em classificação de dados de alta dimensionalidade, na qual a saída da camada de *encoder* (representação reduzida) é utilizada como entrada para outro classificador (AGGARWAL, 2016). Um dos experimentos realizados nesse trabalho utilizou essa abordagem, utilizando conjuntamente um *autoencoder* com camadas LSTM seguido do LOF.

### 3.5 AVALIAÇÃO DE DESEMPENHO

Na última etapa do processo de classificação, é necessário avaliar o desempenho obtido. De acordo com Kowsari et al. (2019), essa avaliação é geralmente feita por meio de métricas, tais como acurácia, revogação, precisão ou por meio do indicador medida-F1. Estas métricas são obtidas a partir da matriz de confusão, ilustrada na Tabela 4, na qual são listados os valores verdadeiros positivos (ou *True Positive* - TP), falsos positivos (ou *False Positive* FP), verdadeiros negativos (ou *True Negative* - TN) e falsos negativos (ou *False Negative* - FN) para cada classe.

A acurácia ( $Acc_i$ ), representada na Equação 13, indica a porcentagem de amostras classificadas adequadamente em toda a base de dados, ou seja, o quanto frequente o classificador está correto. A revogação ( $R_i$ ), representada na Equação 14, é definida

Tabela 4 – Matriz de confusão para avaliação de desempenho.

|             |                     | Classe prevista   |                   |
|-------------|---------------------|-------------------|-------------------|
|             |                     | Previsto Positivo | Previsto Negativo |
| Classe real | Verdadeiro Positivo | <i>TP</i>         | <i>FN</i>         |
|             | Verdadeiro Negativo | <i>FP</i>         | <i>TN</i>         |

Fonte: Adaptado de Kadhim (2019).

como a porcentagem de amostras classificadas adequadamente entre as pertencentes a determinada classe, ou seja, quando realmente é da classe  $K$ , o quanto frequente o classificador acerta nesta classe. A precisão ( $P_i$ ) representa, dentre aquelas classificadas como corretos, quantas efetivamente estavam corretas, e está representado na Equação 15. Por fim, a medida F1 representa a média harmônica entre a revogação e a precisão (KADHIM, 2019).

$$Acc_i = \frac{TP_i + TN_i}{TP_i + TN_i + FP_i + FN_i} \quad (13)$$

$$R_i = \frac{TP_i}{TP_i + FN_i} \quad (14)$$

$$P_i = \frac{TP_i}{TP_i + FP_i} \quad (15)$$

$$F1 = \frac{2 \times P_i \times R_i}{P_i + R_i} \quad (16)$$

Testes estatísticos podem ser utilizados para verificar se múltiplos classificadores podem gerar métricas de desempenho cujas médias sejam iguais entre si. O teste de Friedman é um teste estatístico que é utilizado para detectar diferenças entre vários experimentos. O teste de Wilcoxon é um teste de hipóteses utilizado quando se deseja comparar duas amostras relacionadas para avaliar se as médias diferem entre si (DEMŠAR, 2006).

Nesse trabalho, foi utilizada a métrica de medida F1 para análise de desempenho. Os testes estatísticos de Friedman e Wilcoxon também foram aplicados aos resultados dos experimentos de forma a permitir comparações entre os classificadores entre si e com os resultados do trabalho de Vargas (2019).

## 4 RESULTADOS E DISCUSSÕES

Esse capítulo se divide em duas seções. Na primeira seção são apresentados os resultados dos experimentos realizados conforme o *benchmark* para detecção de anomalias proposto por Vargas (2019). Esses experimentos foram realizados no nível de instância de duas diferentes formas: com e sem a etapa de extração de características. Na segunda seção são apresentados os resultados de experimentos realizados com o agrupamento de instâncias e utilização do classificador *Autoencoder* com camadas LSTM.

### 4.1 EXPERIMENTOS COM REGRAS DO *BENCHMARK*

Na primeira etapa deste trabalho, foram realizados experimentos seguindo-se as seguintes regras estabelecidas no *benchmark* para detecção de anomalias proposto por Vargas (2019):

- Apenas instâncias reais com anomalias de tipos que têm períodos normais (1, 2, 5, 6, 7 e 8) maiores ou iguais a vinte minutos foram utilizadas;
- Múltiplas rodadas de treinamento e teste foram realizadas, sendo o número de rodadas igual ao número de instâncias. Em cada rodada, as amostras utilizadas para treinamento ou teste foram extraídas de apenas uma instância. Parte das amostras de normalidade foram utilizadas no treinamento e a outra parte, no teste. Todas as amostras de anormalidades foram utilizadas apenas no teste (técnica de aprendizagem de classe única). O conjunto de teste foi composto pelo mesmo número de amostras de cada classe (normalidade e anormalidade);
- Em cada rodada, precisão, revocação e medida F1 foram computadas (valor médio e desvio padrão de cada métrica), sendo o valor médio da medida F1 apresentada nesta seção para comparação com trabalho anterior (VARGAS, 2019).

#### 4.1.1 Experimentos com extração de características

Aqui, são descritos os experimentos nos quais foi realizada a etapa de extração de características conforme descrito no item 3.2. A partir de cada amostra de série temporal, foram extraídas e utilizadas como características a mediana, média, desvio padrão, variância, máximo e mínimo para cada variável.

Previamente à aplicação dos algoritmos com as regras *benchmark* nas instâncias reais, foi realizada a calibração dos classificadores (implementado pela função *ParameterGrid* do scikit-learn) em 426 instâncias simuladas por meio de rodadas em diferentes combinações entre classificadores e hiperparâmetros. Na Tabela 5 constam os valores de médias F1 e

desvio padrão (entre parênteses), por algoritmo (com melhores parâmetros destacados em negrito), para o melhor caso nas instâncias simuladas e nas instâncias reais.

Tabela 5 – Médias da medida F1 e desvio padrão (entre parênteses) do experimento com extração de características, por algoritmo (com melhores parâmetros destacados em negrito), para o melhor caso nas instâncias simuladas e nas instâncias reais.

| Classificador                      | Combinações de hiperparâmetros  | Melhor média F1 e desvio padrão nas instâncias simuladas | Média F1 e desvio padrão nas instâncias reais |
|------------------------------------|---|--|---|
| Local Outlier Factor               | 'n_neighbors': [5, 10, 15, <b>20</b> ],<br>'metric': ['hamming', 'euclidean', 'manhattan', 'minkowski'],<br>'contamination': ['auto', 0.01, 0.05, 0.10],  | 0,915 (0,035)  | 0,870 (0,140)                                 |
| Floresta de Isolamento             | 'n_estimators': [50, 100, 150, <b>200</b> ],<br>'max_samples': ['auto', 0.50, <b>0.75</b> , 1.0],<br>'contamination': ['auto', 0, 0.05, <b>0.10</b> ],<br>'bootstrap': [True, <b>False</b> ],<br>'max_features': [0.50, 0.75, <b>1.0</b> ], | 0,777 (0,187)  | 0,701 (0,176)                                 |
| Autoencoder ( <i>feedforward</i> ) | 'hidden_neurons': [8, 4, 4, 8], [ <b>4</b> , 2, 2, 4]<br>'dropout': [0, 0.2],<br>'batch_size': [4, 8]   | 0,721 (0,184)  | 0,590 (0,169)                                 |
| Envelope Elíptico                  | 'contamination': [1e-4, 1e-3, <b>0.01</b> , 0.05, 0.10, 0.50],<br>'assume_centered': [True, <b>False</b> ],<br>'support_fraction': [0.95, 0.975, <b>0.99</b> ],   | 0,577 (0,132)  | 0,586 (0,141)                                 |
| One-class SVM                      | 'kernel': ['linear', 'rbf', 'poly', 'sigmoid'],<br>'gamma': ['auto', 'scale', 1e-4, 1e-3, <b>1e-2</b> , 0.1, 0.50, 1.0, 5.0, 10.0],<br>'nu': [1e-4, <b>1e-3</b> , 1e-2, 0.10, 0.50, 1.0]  | 0,572 (0,208)  | 0,477 (0,221)                                 |

Fonte: elaborado pelo autor.

O classificador que obteve a melhor medida F1 nos testes foi o LOF com medida F1 de 0,870, seguido da Floresta de Isolamento com F1 de 0,701. Uma conjectura sobre o fato do LOF ter apresentado melhor resultado é de que a definição de fronteiras dos casos normais em uma única classe, como no OCSVM e no Envelope Elíptico que apresentaram os menores valores, não é bem definida. Desse modo, mesmo os casos normais são melhor representados por vários grupamentos, e por isso a medida F1 foi maior para a Floresta de Isolamento e para o LOF.

Testes estatísticos (não-paramétricos) foram utilizados neste trabalho e os seus resultados foram analisados considerando-se significância de 5%. A verificação se múltiplos classificadores podem gerar métricas F1 cujas médias sejam iguais entre si foi feita com o Teste de Friedman. Como o resultado desse teste ( $p = 2,432 \times 10^{-20}$ ) rejeitou a hipótese nula, pode-se concluir que ao menos um dos classificadores testados gera métricas F1 cuja média é diferente em relação às demais com alta probabilidade.

Na sequência, a verificação de quais classificadores podem gerar métricas F1 cujas médias

sejam iguais à média obtida pelos demais classificadores foi feita com o Teste de Wilcoxon. Nesse teste, utilizou-se as mesmas métricas F1 submetidas ao Teste de Friedman e a correção de Bonferroni. Os resultados dos testes pareados obtidos são apresentados na Tabela 6.

Tabela 6 – Valores  $p$  dos testes estatísticos de Wilcoxon com ajuste de Bonferroni dos experimentos com extração de características.

| <i>Local Outlier Factor</i>                  | 1,000000        |                 |                 |          |                 |          |          |          |  |
|--|-----------------|-----------------|-----------------|----------|-----------------|----------|----------|----------|--|
| Floresta de Isolamento<br>(benchmark Vargas) | <b>0,006755</b> | 1,000000        |                 |          |                 |          |          |          |  |
| Floresta de Isolamento                       | <b>0,000306</b> | 1,000000        | 1,000000        |          |                 |          |          |          |  |
| <i>Autoencoder (feedforward)</i>             | <b>0,000006</b> | <b>0,007872</b> | 0,065352        | 1,000000 |                 |          |          |          |  |
| Envelope Elíptico                            | <b>0,000020</b> | <b>0,010399</b> | 0,056257        | 1,000000 | 1,000000        |          |          |          |  |
| <i>Dummy</i>                                 | <b>0,000005</b> | <b>0,000127</b> | 0,000218        | 0,112747 | <b>0,039093</b> | 1,000000 |          |          |  |
| <i>One-Class SVM</i><br>(benchmark Vargas)   | <b>0,000018</b> | 0,006883        | <b>0,010718</b> | 1,000000 | 1,000000        | 1,000000 | 1,000000 |          |  |
| <i>One-Class SVM</i>                         | <b>0,000024</b> | <b>0,010787</b> | <b>0,007326</b> | 1,000000 | 1,000000        | 1,000000 | 1,000000 | 1,000000 |  |

Local Outlier Factor  
Floresta de Isolamento  
(benchmark Vargas)  
Floresta de Isolamento  
*Autoencoder (feedforward)*  
Envelope Elíptico  
*Dummy*  
*One-Class SVM*  
(benchmark Vargas)  
*One-Class SVM*

Fonte: elaborado pelo autor.

Os dados da tabela mostram que a hipótese nula pode ser rejeitada para o classificador LOF em todas as comparações.

Os resultados obtidos para os classificadores Floresta de Isolamento e One-class SVM não apresentaram melhorias estatisticamente significativas em relação aos resultados obtidos por Vargas (2019), no qual a medida F1 foi 0,727 para Floresta de Isolamento e 0,470 para *One-class SVM* (kernel sigmóide), respectivamente. Os classificadores *Local Outlier Factor*, Envelope Elíptico e *Autoencoder (feedforward)* não foram utilizados no trabalho de referência.

Portanto, em função dos resultados apresentados na Tabela 6, pode-se concluir que os classificadores baseados em *Local Outlier Factor* geram, com alta probabilidade, métricas F1 cujas médias são diferentes e maiores em relação à média de métricas F1 obtidas com os demais classificadores.

#### 4.1.2 Experimentos sem extração de características

Com o objetivo de permitir comparações na experimentação com redes neurais *autoencoder* com camadas LSTM, que demandam séries temporais como entrada, foi realizada uma rodada de simulação de todos os classificadores sem a realização da etapa de extração de características. Foram simuladas as mesmas combinações do item anterior e também foi incluído um novo classificador baseado em redes neurais. A Tabela 7 mostra os resultados obtidos de médias da medida F1 e desvio padrão (entre parênteses) para cada algoritmo (com melhores parâmetros destacados em negrito), para o melhor caso nas instâncias simuladas e nas instâncias reais.

Tabela 7 – Médias da medida F1 e desvio padrão (entre parênteses) do experimento sem extração de características, por algoritmo (com melhores parâmetros destacados em negrito), para o melhor caso nas instâncias simuladas e nas instâncias reais.

| Classificador                    | Combinações de hiperparâmetros  | Melhor média F1 e desvio padrão nas instâncias simuladas | Média F1 e desvio padrão nas instâncias reais |
|----------------------------------|---|--|---|
| <i>Local Outlier Factor</i>      | 'n_neighbors': [5, 10, 15, <b>20</b> ],<br>'metric': ['hamming', 'euclidean', ' <b>manhattan</b> ', 'minkowski'],<br>'contamination': [' <b>auto</b> ', 0.01, 0.05, 0.10],  | 0,920 (0,027)  | 0,859 (0,129)                                 |
| Envelope Elíptico                | 'contamination': [ <b>1e-4</b> , 1e-3, 0.01, 0.05, 0.10, 0.50],<br>'assume_centered': [ <b>True</b> , False],<br>'support_fraction': [0.95, 0.975, <b>0.99</b> ],   | 0,665 (0,160)  | 0,650 (0,145)                                 |
| <i>Autoencoder</i> (LSTM)        | 'lstm_units': [ <b>16</b> , 8, 8, <b>16</b> ], [8, 4, 4, 8]<br>'dropout': [0, 0.2],<br>'batch_size': [4, 8]   | 0,668 (0,153)  | 0,627 (0,179)                                 |
| Floresta de Isolamento           | 'n_estimators': [ <b>50</b> , 100, 150, 200],<br>'max_samples': ['auto', 0.50, <b>0.75</b> , 1.0],<br>'contamination': ['auto', 0, 0.05, <b>0.10</b> ],<br>'bootstrap': [ <b>True</b> , False],<br>'max_features': [0.50, <b>0.75</b> , 1.0], | 0,680 (0,186)  | 0,616 (0,183)                                 |
| <i>Autoencoder</i> (feedforward) | 'hidden_neurons': [ <b>16</b> , 8, 8, <b>16</b> ], [8, 4, 4, 8]<br>'dropout': [0, 0.2],<br>'batch_size': [4, 8]   | 0,751 (0,192)  | 0,579 (0,169)                                 |
| <i>One-class SVM</i>             | 'kernel': ['linear', 'rbf', 'poly', ' <b>sigmoid</b> '],<br>'gamma': ['auto', 'scale', 1e-4, 1e-3, <b>1e-2</b> , 0.1, 0.50, 1.0, 5.0, 10.0],<br>'nu': [ <b>1e-4</b> , 1e-3, 1e-2, 0.10, 0.50, 1.0]  | 0,569 (0,178)  | 0,551 (0,191)                                 |

Fonte: elaborado pelo autor.

O classificador que obteve a melhor medida F1 nos testes foi o LOF com medida F1 de 0,859, seguido do Envelope Elíptico com F1 de 0,650 e do *Autoencoder* (LSTM) com F1 de 0,627.

Testes estatísticos (não-paramétricos) também foram utilizados nesta etapa, e os seus resultados foram analisados considerando-se significância de 5%. A verificação se múltiplos

classificadores podem gerar métricas F1 cujas médias sejam iguais entre si foi feita com o Teste de Friedman. Como o resultado desse teste (valor  $p = 3,812 \times 10^{-19}$ ) rejeitou a hipótese nula, pode-se concluir que ao menos um dos classificadores testados gera métricas F1 cuja média é diferente em relação às demais com alta probabilidade.

Na sequência, a verificação de quais classificadores podem gerar métricas F1 cujas médias sejam iguais à média obtida pelos demais classificadores foi feita com o Teste de Wilcoxon. Nesse teste, utilizou-se as mesmas métricas F1 submetidas ao Teste de Friedman e a correção de Bonferroni. Os resultados dos testes pareados obtidos são apresentados na Tabela 6.

Tabela 8 – Valores  $p$  dos testes estatísticos de Wilcoxon com ajuste de Bonferroni dos experimentos sem extração de características.

|  |  | <i>Local Outlier Factor</i>      |                          |                           |                               |                                  |                      |              |
|--|--|----------------------------------|--------------------------|---------------------------|-------------------------------|----------------------------------|----------------------|--------------|
|  |  | 1,000000                         |                          |                           |                               |                                  |                      |              |
|  |  | Envelope Elíptico                | 0.000016                 | 1,000000                  |                               |                                  |                      |              |
|  |  | <i>Autoencoder</i> (LSTM)        | 0.000007                 | 1,000000                  | 1,000000                      |                                  |                      |              |
|  |  | Floresta de Isolamento           | 0.000013                 | 1.000000                  | 1.000000                      | 1,000000                         |                      |              |
|  |  | <i>Autoencoder</i> (feedforward) | 0.000007                 | 0.629214                  | 0.734375                      | 1,000000                         | 1,000000             |              |
|  |  | <i>One-Class SVM</i>             | 0.000010                 | 0.296349                  | 1.000000                      | 1.000000                         | 1.000000             | 1,000000     |
|  |  | <i>Dummy</i>                     | 0.000003                 | 0.000480                  | 0.018093                      | 0.007500                         | 0.100479             | 1,000000     |
|  |  |                                  |                          |                           |                               |                                  |                      | 1,000000     |
|  |  | <i>Local Outlier Factor</i>      | <i>Envelope Elíptico</i> | <i>Autoencoder</i> (LSTM) | <i>Floresta de Isolamento</i> | <i>Autoencoder</i> (feedforward) | <i>One-Class SVM</i> | <i>Dummy</i> |

Fonte: elaborado pelo autor.

Os dados da tabela mostram que a hipótese nula pode ser rejeitada para o classificador LOF em todas as comparações.

Os resultados obtidos para os classificadores Envelope Elíptico, Autoencoder (LSTM) e Floresta de Isolamento somente foram significativamente superiores ao classificador ingênuo (*Dummy*). Já os classificadores Autoencoder (feedforward) e *One-class SVM* não foram estatisticamente melhores que classificador ingênuo.

Em função dos resultados apresentados na Tabela 6, pode-se concluir que os classificadores baseados em *Local Outlier Factor* gera, com alta probabilidade, métricas F1 cujas médias

são diferentes e maiores em relação à média de métricas F1 obtidos com os demais classificadores.

Em relação aos resultados obtidos com as redes neurais, presume-se que esses poderiam ser diferentes com o uso de uma maior quantidade de dados, já que os experimentos do *benchmark* são realizados no nível de instância (um classificador para cada série temporal). Assim, a fim de verificar o desempenho das redes neurais em um cenário com maior quantidade de dados, foram realizadas experimentações com o agrupamento de instâncias. Tais experimentos são apresentados na próxima seção.

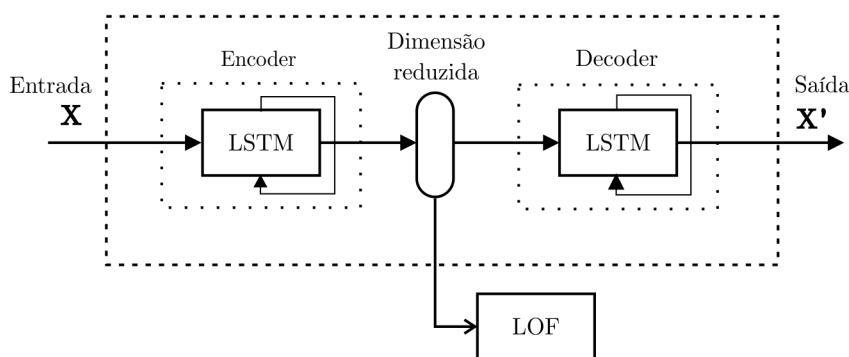
## 4.2 EXPERIMENTOS COM AGRUPAMENTO DE INSTÂNCIAS

Para uma melhor análise do desempenho do classificador *Autoencoder* (LSTM) foi realizado um experimento utilizando as amostras das instâncias de forma conjunta. Todas as amostras das instâncias com anomalias de tipos que têm períodos normais maiores ou iguais a vinte minutos (1, 2, 5, 6, 7 e 8) foram agrupadas. O pré-processamento foi realizado da mesma forma das seções anteriores e conforme descrito na Seção 3.1. Das amostras normais, 60% foram utilizadas para treinamento 40% para validação. As amostras rotuladas como anomalias foram utilizadas apenas para validação. Também foi realizado um outro experimento utilizando-se as características reduzidas da saída do *encoder* como entrada para o classificador LOF, a fim de verificar seu desempenho nesse cenário.

### 4.2.1 Uso conjunto de Autoencoder LSTM e Local Outlier Factor

Conforme ilustrado na Figura 17, as características reduzidas da saída do *encoder* foram utilizadas como entrada para o classificador LOF, a fim de verificar seu desempenho nesse cenário.

Figura 17 – Combinação entre LOF e características reduzidas geradas pelo *Autoencoder* LSTM.



Fonte: adaptado de Elsayed et al. (2020).

A Tabela 9 mostra os resultados de medida F1 do experimento com agrupamento de instâncias e sem extração de características, por algoritmo (com melhores parâmetros destacados em negrito), para o melhor caso nas instâncias simuladas e nas instâncias reais.

Houve um aumento significativo do desempenho numérico da medida F1 (0,815) para a rede neural *Autoencoder* (LSTM) em comparação aos resultados obtidos pela medida F1 média obtida com classificadores individuais por instância (0,627). Por outro lado, percebe-se que houve um diminuição numérica dos valores de medida F1 do classificador LOF quando aplicado às características reduzidas da rede neural, tanto para as instância simuladas quanto para as instâncias reais.

Tabela 9 – Medida F1 do experimento com agrupamento de instâncias e sem extração de características, por algoritmo (com melhores parâmetros destacados em negrito), para o melhor caso nas instâncias simuladas e nas instâncias reais.

| Classificador  | Combinações de hiperparâmetros  | Melhor média F1 nas instâncias simuladas | Média F1 nas instâncias reais |
|--|---|--|-------------------------------|
| Autoencoder (LSTM)                                     | 'hidden_neurons': [32, <b>16, 16, 32</b> ],[64, 32, 32, 64]<br>'learning_rate': [0.0001, 0.001, 0.01]<br>'dropout': [0, 0.1, 0.2],<br>'batch_size': [32, 64]      | 0,904                                    | 0,815                         |
| Local Outlier Factor (dados obtidos da camada encoder) | 'n_neighbors': [5, 10, 15, <b>20</b> ],<br>'metric': ['hamming', 'euclidean', 'manhattan' , 'minkowski'],<br>'contamination': [ <b>auto</b> ', 0.01, 0.05, 0.10], | 0,820                                    | 0,500                         |

Fonte: elaborado pelo autor.

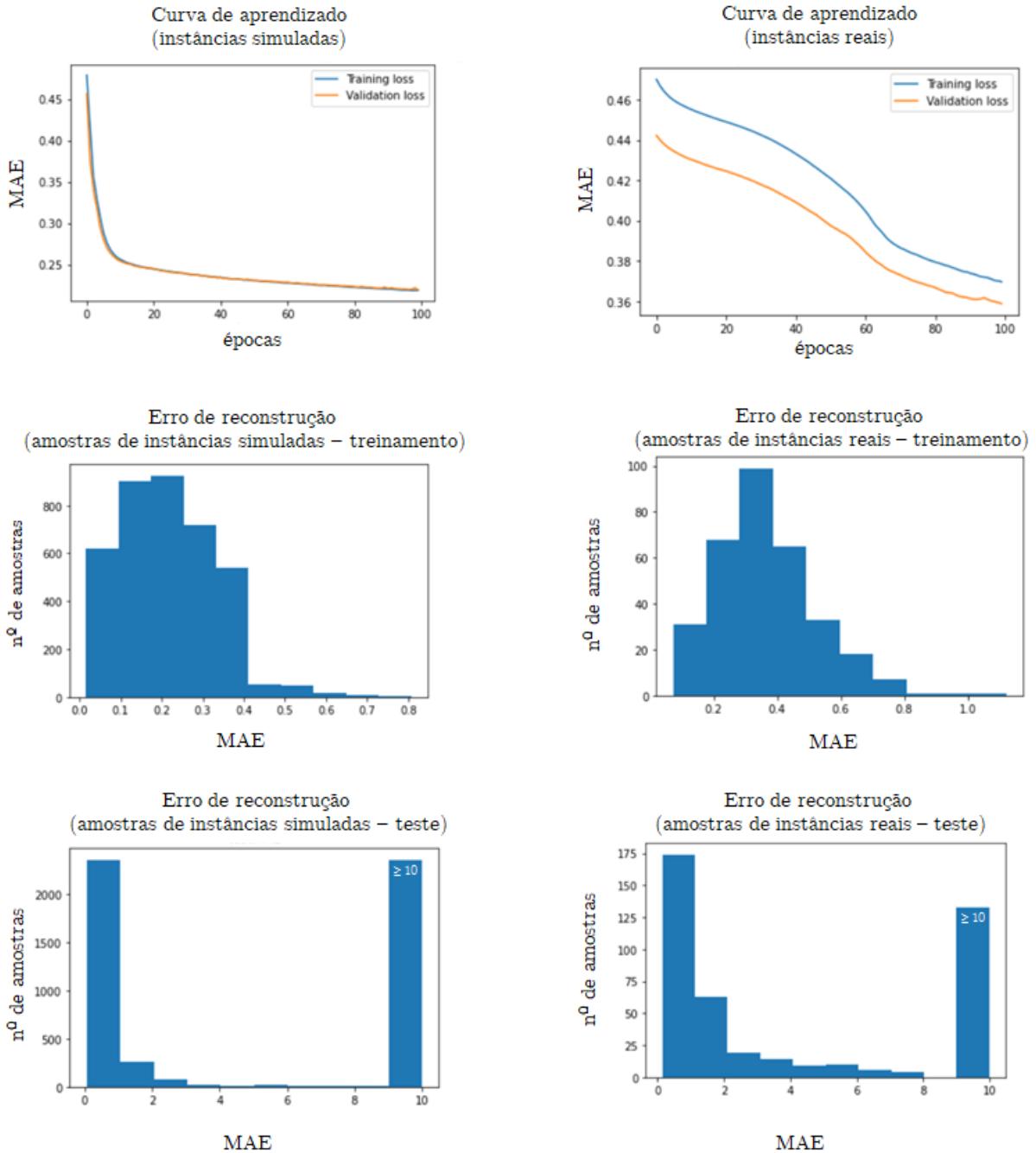
As curvas de aprendizado e a distribuição do erro absoluto médio de reconstrução das instâncias está na Figura 18. Nessa figura observa-se duas colunas, uma para instâncias simuladas e outra para instâncias reais, com os seguintes gráficos: curva de aprendizado, erro de construção em treinamento e erro de reconstrução em teste.

Dos gráficos de erros de reconstrução percebe-se que nas amostras de treinamento os erros de reconstrução foram baixos (de 0 a 1) e que nas amostras de teste existe uma maior diferenciação. Esse comportamento corrobora o que era esperado tendo em vista que em treinamento apenas de amostras classificadas como normais foram utilizadas.

Com relação às curvas de aprendizado, observa-se um estabilização de erro ao longo das épocas ao longo do experimento nas instâncias simuladas, porém o mesmo comportamento não foi observado nas instâncias reais.

Esse ausência de estabilização no treinamento da rede neural utilizando apenas as instâncias reais com os parâmetros obtidos nas instâncias simuladas motivou a realização de um

Figura 18 – Curvas de aprendizado e distribuição do erro absoluto médio de reconstrução para as amostras das instâncias.



Fonte: elaborado pelo próprio autor.

experimento adicional, descrito na próxima seção, na qual foi realizada a combinação de instâncias reais e simuladas de forma gradual.

#### 4.2.2 Combinação de instâncias reais e simuladas

Para permitir o experimento com a combinação de instâncias reais e simuladas, o primeiro passo foi a separação das instâncias simuladas em duas partes, sendo a primeira metade utilizada para calibração de parâmetros e a segunda metade incluída conjuntamente com

instâncias reais de forma gradativa de 10% em 10% até 50%, com o objetivo de avaliar a curva de aprendizado da rede neural.

Os resultados obtidos na calibração da rede utilizando 50% dos dados simulados para treinamento são apresentados na Tabela 10. Observa-se que os parâmetros selecionados foram os mesmos de quando foi utilizado 100% dos dados, porém com o resultado de medida F1 (0,884) menor do que no caso que utilizou todas as instâncias simuladas (0,904).

Tabela 10 – Medida F1 do experimento com agrupamento de instâncias e sem extração de características, utilizando 50% das instâncias simuladas.

| Classificador               | Combinações de parâmetros   | Melhor média F1 nas instâncias simuladas (50% delas) |
|-----------------------------|---|--|
| Autoencoder ( <i>LSTM</i> ) | 'hidden_neurons': [32, 16, 16, 32], [64, 32, 32, 64]<br>'learning_rate': [0.0001, 0.001, 0.01]<br>'dropout': [0, 0.1, 0.2],<br>'batch_size': [32, 64] | 0,884  |

Fonte: elaborado pelo autor.

Em seguida foi realizada o experimento em combinações de instâncias reais com as demais 50% das instâncias simuladas, com a inclusão gradativa dessas demais instâncias simuladas de 10% em 10% até 50%. Os resultados obtidos estão na Tabela 11.

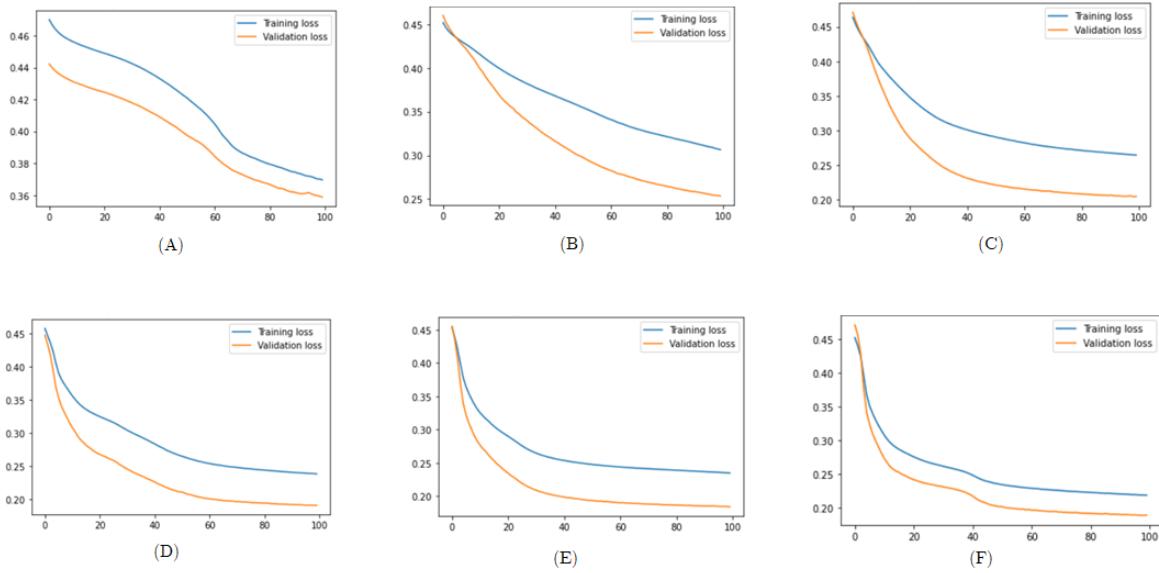
Tabela 11 – Medida F1 do experimento com agrupamento de instâncias e sem extração de características, utilizando instâncias reais e inclusão gradativa das demais instâncias simuladas de 10% em 10% até 50%.

| Classificador               | Dados utilizados                                     | Média F1 nas instâncias reais + simuladas |
|-----------------------------|--|---|
| Autoencoder ( <i>LSTM</i> ) | (A) Instâncias reais (100%)                          | 0,815                                     |
|                             | (B) Instâncias reais (100%) + demais simuladas (10%) | 0,852                                     |
|                             | (C) Instâncias reais (100%) + demais simuladas (20%) | 0,868                                     |
|                             | (D) Instâncias reais (100%) + demais simuladas (30%) | 0,836                                     |
|                             | (E) Instâncias reais (100%) + demais simuladas (40%) | 0,890                                     |
|                             | (F) Instâncias reais (100%) + demais simuladas (50%) | 0,866                                     |

Fonte: elaborado pelo autor.

Os resultados mostram que houve melhoria numérica no valor de medida F1. Com relação à curvas de aprendizado, também pode-se observar uma maior estabilização das mesmas, conforme ilustrado na Figura 19.

Figura 19 – Curvas de aprendizado do experimento com agrupamento de instâncias e sem extração de características, utilizando instâncias reais e inclusão gradativa das demais instâncias simuladas de 10% em 10% até 50%.



Fonte: elaborado pelo próprio autor.

Embora a utilização de instâncias agrupadas não esteja previsto no *benchmark* de referência, o resultado obtido mostra que a utilização agrupada das instâncias reais e simuladas auxiliou na estabilização dos erros de treinamento e nos valores de desempenho obtidos pela rede neural *Autoencoder* (LSTM) auferidos pela métrica de medida F1.

## 5 CONCLUSÃO

O presente trabalho aplicou e comparou quantitativamente técnicas de detecção de anomalias em poços marítimos produtores surgentes de petróleo utilizando a base de dados pública 3W *dataset* e o *benchmark* para detecção de anomalias elaborado por Vargas (2019).

Classificadores de classe única citados pela literatura e que apresentam diferentes características em seus modelos preditivos foram selecionados para experimentação: Floresta de Isolamento, *One-class Support Vector Machine* (OCSVM), *Local Outlier Factor* (LOF), Envelope Elíptico e *Autoencoder* com camadas *feedforward* e LSTM (*Long short-term memory*).

Experimentos foram realizados para detecção de anomalias com a aplicação dos algoritmos à base de dados, tendo sido apuradas as métricas de medida F1 e desvios padrão. A partir dos testes estatísticos aplicados, é possível verificar que o classificador *Local Outlier Factor* (LOF) apresenta desempenho superior aos demais (medida F1 de 0,870 e 0,859 para o cenário com e sem extração de características, respectivamente) ao ser aplicado conforme as regras do *benchmark*. Uma análise sobre esse resultado pressupõe que a definição de fronteiras dos casos normais em uma única classe não é bem definida e, desse modo, mesmo os casos normais são melhor representados por vários agrupamentos, o que explica o maior valor de medida F1 para o classificador LOF.

Também foram realizados experimentos complementares com redes neurais *Autoencoder* com camadas LSTM, que mostraram que o desempenho das redes neurais (medida F1 de 0,815) comparou-se numericamente ao LOF somente quando as instâncias foram utilizadas de forma conjunta. Esse cenário com agrupamento das amostras mostrou que a maior disponibilidade de dados aumentou o desempenho das redes neurais em termos de medida F1. Adicionalmente, a separação de parte das instâncias simuladas para uso em validação propiciou maior estabilidade para as curvas de aprendizado e maior desempenho numérico (medida F1 de 0,866 na aplicação nas instâncias reais com 50% das instâncias simuladas).

Tendo em vista os resultados numéricos similares, a escolha da técnica mais adequada para aplicação em ambiente profissional depende também dos recursos disponíveis para o processo de monitoramento e detecção de anomalias. O uso de classificadores individuais por instância como o LOF e conforme as regras do *benchmark* se mostra uma alternativa interessante por permitir configurações de hiperparâmetros específicos para cada poço marítimo. Porém, a quantidade de modelos a serem treinados pode ser grande a depender da quantidade de poços a serem monitorados.

Por outro lado, devido ao fenômeno de *concept drift*, que são mudanças nos valores das variáveis ao longo do tempo que deterioram o desempenho dos modelos aprendidos e que

pode ocorrer em séries temporais de processos industriais (KRAWCZYK et al., 2017), é possível que o uso das redes neurais utilizadas de forma agrupada ajude a minimizar esse efeito, além de reduzir a quantidade de modelos a serem treinados e monitorados.

## 5.1 TRABALHOS FUTUROS

Embora os objetivos propostos no trabalho tenham sido alcançados, algumas propostas de continuação do estudo são possíveis visando trabalhos futuros:

- Implantar os classificadores no sistema informatizado de monitoramento de poços da empresa.
- Realizar a etapa de extração de características utilizando diferentes técnicas.
- Experimentar os algoritmos baseados em redes neurais com mais variações de estruturas e camadas.
- Realizar experimentos com algoritmos multiclasse.
- Aumentar a base de dados com instâncias de outros tipos de poços além do tipo marítimo surgente.

## REFERÊNCIAS

- ABADI, Martín et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. 2015. Software available from tensorflow.org. Disponível em: <<https://www.tensorflow.org/>>.
- AGGARWAL, Charu C. *Outlier Analysis*. 2nd. ed. [S.l.]: Springer, 2016.
- AGGARWAL, Charu C; SATHE, Saket. *Outlier ensembles: An introduction*. [S.l.]: Springer, 2017.
- ANDREOLLI, Ivanildo. *Introdução à Elevação e Escoamento Monofásico e Multifásico de Petróleo*. [S.l.]: Interciência, 2016.
- ANP. *Boletim Mensal da Produção de Petróleo e Gás Natural*. 2020. [acessado em 25-03-2020]. Disponível em: <<http://www.anp.gov.br/>>.
- BAI, Qiang; BAI, Yong. *Sistemas marítimos de produção de petróleo: processos, tecnologias e equipamentos offshore*. [S.l.]: Elsevier Brasil, 2015.
- BARBARIOL, Tommaso; FELTRESI, Enrico; SUSTO, Gian Antonio. Machine learning approaches for anomaly detection in multiphase flow meters. *IFAC-PapersOnLine*, Elsevier, v. 52, n. 11, p. 212–217, 2019.
- BP. *British Petroleum Energy Outlook*: 2019 edition. London, United Kingdom: [s.n.], 2019. [acessado em 20-04-2020]. Disponível em: <<https://www.bp.com/content/dam/bp/business-sites/en/global/corporate/pdfs/energy-economics/energy-outlook/bp-energy-outlook-2019.pdf>>.
- BRANCO, Luiz Henrique Castelo. Maniac: uma metodologia para o monitoramento automatizado das condições dos pavimentos utilizando vants. Biblioteca Digital de Teses e Dissertações da USP, 2017.
- BREUNIG, Markus M et al. Lof: identifying density-based local outliers. In: *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*. [S.l.: s.n.], 2000. p. 93–104.
- CHAN, Chun-Fai et al. Detecting anomalies in programmable logic controllers using unsupervised machine learning. In: SPRINGER. *IFIP International Conference on Digital Forensics*. [S.l.], 2019. p. 119–130.
- CHANDOLA, Varun; BANERJEE, Arindam; KUMAR, Vipin. Anomaly detection: A survey. *ACM Computing Surveys*, ACM, v. 41, n. 3, p. 1–58, jul 2009. ISSN 03600300.
- CHEN, Jinghui et al. Outlier detection with autoencoder ensembles. In: SIAM. *Proceedings of the 2017 SIAM international conference on data mining*. [S.l.], 2017. p. 90–98.
- CHEN, Wo-Ruo et al. Representative subset selection and outlier detection via isolation forest. *Analytical methods*, Royal Society of Chemistry, v. 8, n. 39, p. 7225–7231, 2016.
- CHRIST, Maximilian et al. Time series feature extraction on basis of scalable hypothesis tests (tsfresh—a python package). *Neurocomputing*, Elsevier, v. 307, p. 72–77, 2018.
- DEMŠAR, Janez. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine learning research*, v. 7, n. Jan, p. 1–30, 2006.

- DUDA, Richard O; HART, Peter E; STORK, David G. *Pattern classification*. [S.l.]: John Wiley & Sons, 2012.
- EL SAYED, Mahmoud Said et al. Network anomaly detection using lstm based autoencoder. In: *Proceedings of the 16th ACM Symposium on QoS and Security for Wireless and Mobile Networks*. [S.l.: s.n.], 2020. p. 37–45.
- FAWAZ, Hassan Ismail et al. Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery*, Springer, v. 33, n. 4, p. 917–963, 2019.
- FILHO, José Edson de Albuquerque et al. Detecção de anomalia nas eleições de 2018 com isolation forest. *Revista de Engenharia e Pesquisa Aplicada*, v. 5, n. 1, p. 104–109, 2020.
- GAUTO, Marcelo Antunes et al. *Petróleo e gás: princípios de exploração, produção e refino*. [S.l.]: Bookman Editora, 2016.
- GÉRON, Aurélien. *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow*. [S.l.]: "O'Reilly Media, Inc.", 2019.
- GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron. *Deep learning*. [S.l.]: MIT press, 2016.
- GRASHORN, Philipp; HANSEN, Jonas; RUMMENS, Marcel. *How Airbus Detects Anomalies in ISS Telemetry Data Using TFX*. 2020. [acessado em 22-11-2021]. Disponível em: <<https://blog.tensorflow.org/2020/04/how-airbus-detects-anomalies-iss-telemetry-data-tfx.html>>.
- GREFF, Klaus et al. Lstm: A search space odyssey. *IEEE transactions on neural networks and learning systems*, IEEE, v. 28, n. 10, p. 2222–2232, 2016.
- GUO, Boyun. *Petroleum production engineering, a computer-assisted approach*. [S.l.]: Elsevier, 2011.
- HARDIN, Johanna; ROCKE, David M. Outlier detection in the multiple cluster setting using the minimum covariance determinant estimator. *Computational Statistics & Data Analysis*, Elsevier, v. 44, n. 4, p. 625–638, 2004.
- HAWKINS, Simon et al. Outlier detection using replicator neural networks. In: SPRINGER. *International Conference on Data Warehousing and Knowledge Discovery*. [S.l.], 2002. p. 170–180.
- HOCHREITER, Sepp; SCHMIDHUBER, Jürgen. Long short-term memory. *Neural computation*, MIT Press, v. 9, n. 8, p. 1735–1780, 1997.
- HUBERT, Mia; DEBRUYNE, Michiel. Minimum covariance determinant. *Wiley interdisciplinary reviews: Computational statistics*, Wiley Online Library, v. 2, n. 1, p. 36–43, 2010.
- JÚNIOR, Wander Fernandes et al. Detecção de anomalias em poços produtores de petróleo usando aprendizado de máquina. In: *Congresso Brasileiro de Automática-CBA*. [S.l.: s.n.], 2020. v. 2, n. 1.
- KADHIM, Ammar Ismael. Survey on supervised machine learning techniques for automatic text classification. *Artificial Intelligence Review*, Springer, p. 1–20, 2019.

- KHAN, Samir et al. Unsupervised anomaly detection in unmanned aerial vehicles. *Applied Soft Computing*, Elsevier, v. 83, p. 105650, 2019.
- KHAN, Shehroz S; MADDEN, Michael G. One-class classification: taxonomy of study and review of techniques. *The Knowledge Engineering Review*, Cambridge University Press, v. 29, n. 3, p. 345–374, 2014.
- KOHONEN, Teuvo. Essentials of the self-organizing map. *Neural networks*, Elsevier, v. 37, p. 52–65, 2013.
- KOWSARI, Kamran et al. Text classification algorithms: A survey. *Information*, Multidisciplinary Digital Publishing Institute, v. 10, n. 4, p. 150, 2019.
- KRAWCZYK, Bartosz et al. Ensemble learning for data stream analysis: A survey. *Information Fusion*, Elsevier, v. 37, p. 132–156, 2017.
- KWON, Donghwoon et al. A survey of deep learning-based network anomaly detection. *Cluster Computing*, Springer, v. 22, n. 1, p. 949–961, 2019.
- LIU, Fei Tony; TING, Kai Ming; ZHOU, Zhi-Hua. Isolation-based anomaly detection. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, ACM New York, NY, USA, v. 6, n. 1, p. 1–39, 2012.
- MACROTRENDS. *Brent Crude Oil Prices - 10 Year Daily Chart*. 2020. [acessado em 25-03-2020]. Disponível em: <<https://www.macrotrends.net/>>.
- MARVIN, Minsky; SEYMOUR, A Papert. *Perceptrons*. [S.l.]: MIT Press, 1969.
- MCCULLOCH, Warren S; PITTS, Walter. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, Springer, v. 5, n. 4, p. 115–133, 1943.
- MISRA, Siddharth; LI, Hao; HE, Jiabo. *Machine Learning for Subsurface Characterization*. [S.l.]: Elsevier, 2020.
- PAL, Avishek; PRAKASH, PKS. *Practical Time Series Analysis: Master Time Series Data Processing, Visualization, and Modeling using Python*. [S.l.]: Packt Publishing Ltd, 2017.
- PEDREGOSA, F. et al. *Novelty and Outlier Detection*. 2011. [acessado em 22-04-2021]. Disponível em: <[https://scikit-learn.org/stable/modules/outlier\\_detection.html](https://scikit-learn.org/stable/modules/outlier_detection.html)>.
- PEDREGOSA, F. et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, v. 12, p. 2825–2830, 2011.
- PORTELA, Gerardo. *Gerenciamento de riscos na indústria de petróleo e gás*. [S.l.]: Elsevier, 2015.
- QIN, S Joe. Survey on data-driven industrial process monitoring and diagnosis. *Annual reviews in control*, Elsevier, v. 36, n. 2, p. 220–234, 2012.
- RANJAN, Chitta. *Understanding Deep Learning: Application in Rare Event Prediction*. [S.l.]: Connaissance Publishing, 2020. URL: <[www.understandingdeeplearning.com](http://www.understandingdeeplearning.com)>.

- ROSENBLATT, Frank. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, American Psychological Association, v. 65, n. 6, p. 386, 1958.
- ROUSSEEUW, Peter J; DRIESSEN, Katrien Van. A fast algorithm for the minimum covariance determinant estimator. *Technometrics*, Taylor & Francis Group, v. 41, n. 3, p. 212–223, 1999.
- RUMELHART, David E; HINTON, Geoffrey E; WILLIAMS, Ronald J. Learning representations by back-propagating errors. *nature*, Nature Publishing Group, v. 323, n. 6088, p. 533–536, 1986.
- SANTOS, Tiago; KERN, Roman. A literature survey of early time series classification and deep learning. In: *SamI40 workshop at i-KNOW'16*. [S.l.: s.n.], 2016.
- SCHÖLKOPF, Bernhard et al. Estimating the support of a high-dimensional distribution. *Neural computation*, MIT Press, v. 13, n. 7, p. 1443–1471, 2001.
- TAN, Yanghui et al. A comparative investigation of data-driven approaches based on one-class classifiers for condition monitoring of marine machinery system. *Ocean Engineering*, Elsevier, v. 201, p. 107174, 2020.
- VARGAS, Ricardo Emanuel Vaz. *Base de Dados e Benchmarks para Prognóstico de Anomalias em Sistemas de Elevação de Petróleo*. 2019. Tese (Doutorado) — Universidade Federal do Espírito Santo, 2019.
- VARGAS, Ricardo Emanuel Vaz et al. A realistic and public dataset with rare undesirable real events in oil wells. *Journal of Petroleum Science and Engineering*, v. 181, p. 106223, 2019. ISSN 0920-4105.
- YU, Hongyang et al. Self-organizing map based fault diagnosis technique for non-gaussian processes. *Industrial & Engineering Chemistry Research*, ACS Publications, v. 53, n. 21, p. 8831–8843, 2014.