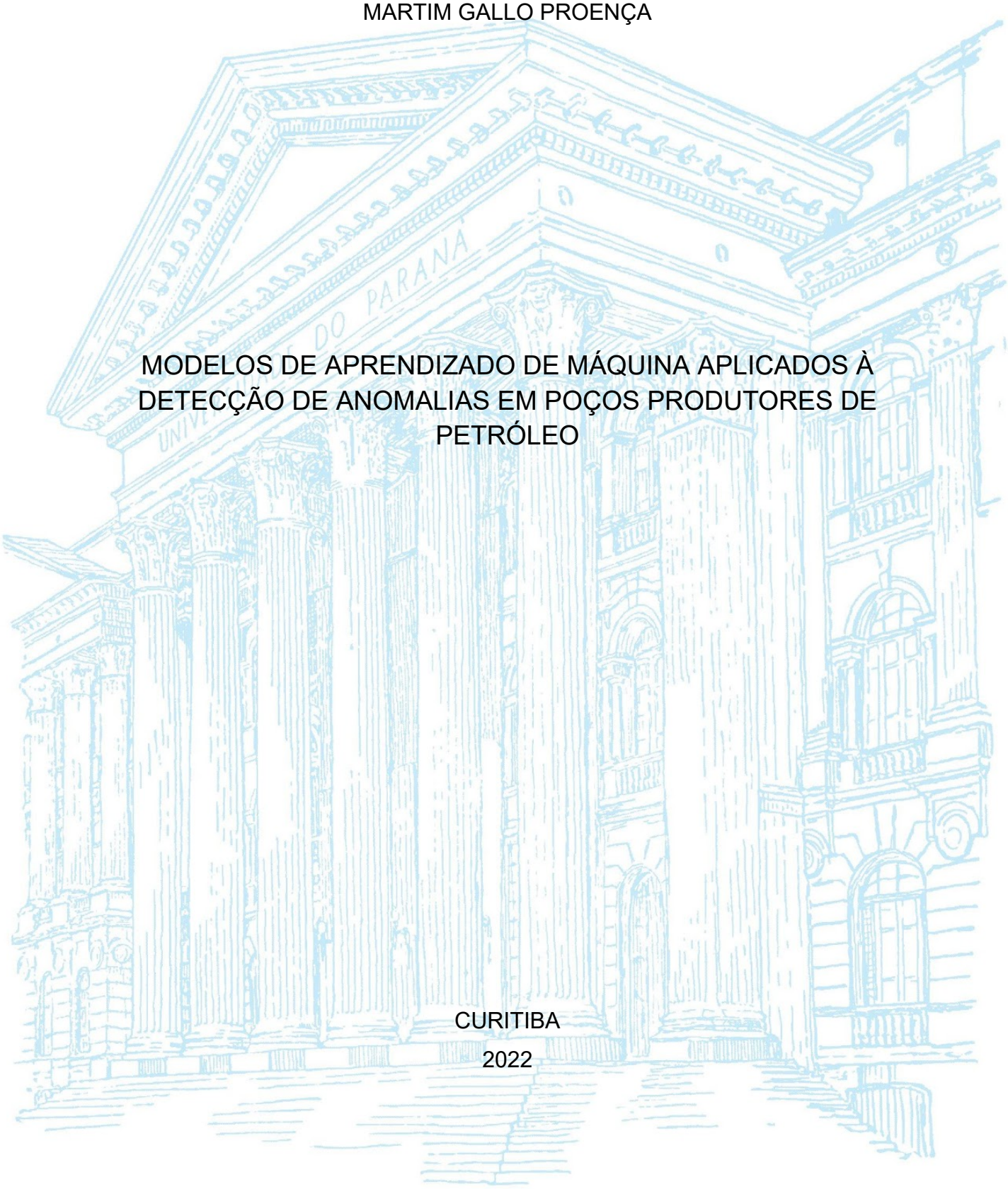
UNIVERSIDADE FEDERAL DO PARANÁ

MARTIM GALLO PROENÇA



MODELOS DE APRENDIZADO DE MÁQUINA APLICADOS À DETECÇÃO DE ANOMALIAS EM POÇOS PRODUTORES DE PETRÓLEO

CURITIBA

2022

MARTIM GALLO PROENÇA

MODELOS DE APRENDIZADO DE MÁQUINA APLICADOS À DETECÇÃO DE ANOMALIAS EM POÇOS PRODUTORES DE PETRÓLEO

TCC apresentado ao curso de Graduação em Engenharia Elétrica, Setor de Tecnologia, Universidade Federal do Paraná, como requisito parcial à obtenção do título de Bacharel em Engenharia Elétrica.

Orientador: Prof. Dr. Leandro Coelho

CURITIBA

2022

TERMO DE APROVAÇÃO

MARTIM GALLO PROENÇA

MODELOS DE APRENDIZADO DE MÁQUINA APLICADOS À DETECÇÃO DE ANOMALIAS EM POÇOS PRODUTORES DE PETRÓLEO

TCC apresentado ao curso de Graduação em Engenharia Elétrica, Setor de Tecnologia, Universidade Federal do Paraná, como requisito parcial à obtenção do título de Bacharel em Engenharia Elétrica.

_____

Prof. Dr. Leandro Coelho
Orientador – Departamento de Engenharia Elétrica, UFPR

_____

Prof. Dr. Luis Henrique Assumpção Lolis
Departamento de Engenharia Elétrica, UFPR

_____

Prof. Dr. Ândrei Camponogara
Departamento de Engenharia Elétrica, UFPR

Curitiba, 22 de setembro de 2022.

AGRADECIMENTOS

Gostaria de agradecer ao meu professor orientador Dr. Leandro Coelho pela instrução durante este período final de minha graduação.

Ao meu pai (Luis Antonio de Oliveira Proença), mãe (Clarissa Weber Gallo) assim como meu irmão (Miguel Gallo Proença) por terem oferecido suporte durante essa etapa importante da minha jornada.

E por fim, aos meu colegas e amigos que me acompanharam durante esta caminhada e tornaram o tempo que passei na UFPR ainda mais memorável.

RESUMO

O presente trabalho oferece uma análise de possíveis soluções para o problema de detecção de anomalias em poços de petróleo utilizando técnicas de aprendizado de máquina e dados provenientes do *3W Dataset*, extraído de poços de petróleo da empresa Petrobras em parceria com a UFES e disponibilizado por (Vargas, 2019). Este dataset é composto por diversas séries temporais multivariadas (*multivariate time series*, MTS) contendo medidas de temperatura e pressão em diferentes pontos de plataformas utilizadas para extração de petróleo. Inicialmente são encontradas as colaborações de demais pesquisadores que já trabalharam no mesmo conjunto de dados aqui utilizado, anotando-se as técnicas de pré-processamento, assim como modelos de classificação e demais abordagens para atacar o problema de detecção de anomalias. Dois algoritmos de classificação não supervisionada já presentes na literatura são escolhidos para serem trabalhados, Floresta de Isolamento (*Isolation Forest*, IF) e Fator de Isolamento Local (*Local Outlier Factor*, LOC). São testadas duas abordagens de janela deslizante para amostrar o conjunto de dados, assim como a adição de colunas de atrasos nas séries temporais. Como contribuição principal, este trabalho também propõe a utilização de diversas combinações (*ensembles*) dos modelos mencionados com o objetivo de melhorar a performance de classificação, aqui medida em F1 *score*. Foi encontrado que a utilização da técnica alternativa de amostragem melhorou consideravelmente o desempenho de um dos classificadores quando comparado à resultados na literatura. No entanto, as combinações de diferentes amostragens, número de modelos nos ensembles e adição de atrasos não resultaram em melhora no F1 *Score* final.

Palavras-chave: Aprendizado de Máquina. Detecção de Anomalias. Floresta de Isolamento. Fator de Isolamento Local.

ABSTRACT

The present thesis offers an analysis of possible solutions to the problem of anomaly detection in oil extraction wells using machine learning techniques and data from the 3W Dataset, provided by (Vargas, 2019) in collaboration with PETROBRAS and the Federal University of Espírito Santo (UFES). This dataset is composed of multivariate time series (MTS) containing measures of temperature and pressure from different points in platforms during the petroleum extraction process. In this thesis, the collaborations of other authors regarding this same dataset and problem are gathered, the pre-processing techniques, classification algorithms and other approaches are summarized. Two algorithms for unsupervised classification found in the literature were chosen to be implemented in this thesis, Isolation Forest (IF) and Local Outlier Factor (LOC). Two approaches based on rolling windows for sampling the dataset are tested, as well as the addition of lag columns. The main collaboration proposed in this work is the additional testing of a variety of model ensembles by Feature Bagging with the intent of improving the performance of the classification, here measured by F1 Score. It was found that the newly proposed sampling increased significantly the performance of one of the models. The combination of the different sampling strategies, number of models used in the ensembles and the addition of lags did not result in an increase in the maximum final F1 score.

Key words: Machine Learning. Anomaly detection. Isolation Forest. Local Outlier Factor.

SUMÁRIO

1 INTRODUÇÃO

Anomalias ou Outliers são muitas vezes tidos por sinônimos*,* como observado por (Garcia et al, 2021)*, e* podem ser definidos como observações que desviam tanto das demais no conjunto de dados que estão inseridas ao ponto de levantar suspeitas de que foram geradas por um mecanismo diferente. Detecção de anomalias, então, diz respeito a técnicas capazes de identificar esses padrões destoantes.

Em diversas aplicações industriais, sensores são usados para controlar e monitorar processos a fim de evitar ou prevenir falhas e outros eventos indesejáveis. Dependendo da aplicação, os volumes de dados podem ser tamanhos, que abordagens na maior parte das vezes não supervisionadas para detecção destes eventos tornam-se necessárias.

Dentro deste contexto, uma equipe de pesquisadores da Universidade Federal do Espírito Santo (UFES) em conjunto com a Petrobras desenvolveu o dataset (conjunto de dados) 3W, composto por diversas séries temporais com medidas reais de sensores de temperatura e pressão em poços de extração de petróleo. (Vargas, 2019). O dataset tem como objetivo fornecer exemplos de anomalias encontradas no contexto industrial de extração petrolífera, e servir como base para o desenvolvimento de ferramentas de detecção antecipada das mesmas a fim de prevenir ou mitigar seus impactos.

1.2 OBJETIVOS

O presente trabalho tem como objetivo avaliar diferentes modelos de aprendizado de máquina, assim como técnicas de processamento, para realizar detecção de anomalias encontradas nas séries temporais do dataset 3W.

1.2.2 Objetivos Específicos

- Revisar a literatura pertinente ao tema de detecção de anomalias e aprendizado de máquina.
- Análise exploratória de dados do dataset 3W
- Treinar, testar e otimizar os classificadores já propostos e disponibilizados por Vargas, (2019)
- Desenvolver um modelo de detecção de anomalias que supere as medidas de desempenho propostas no segundo *Benchmark* proposto em Vargas, (2019)

1.3 Motivação

O presente trabalho visa explorar um dataset real, com benchmarks projetados para simular situações reais com condições não idealizadas a fim de melhorar o desempenho na detecção de anomalias em poços extratores de petróleo. As poucas referências encontradas na literatura que abordam os desafios diretamente obtiveram resultados que ainda tem espaço para serem otimizados.

2 REVISÃO DE LITERATURA

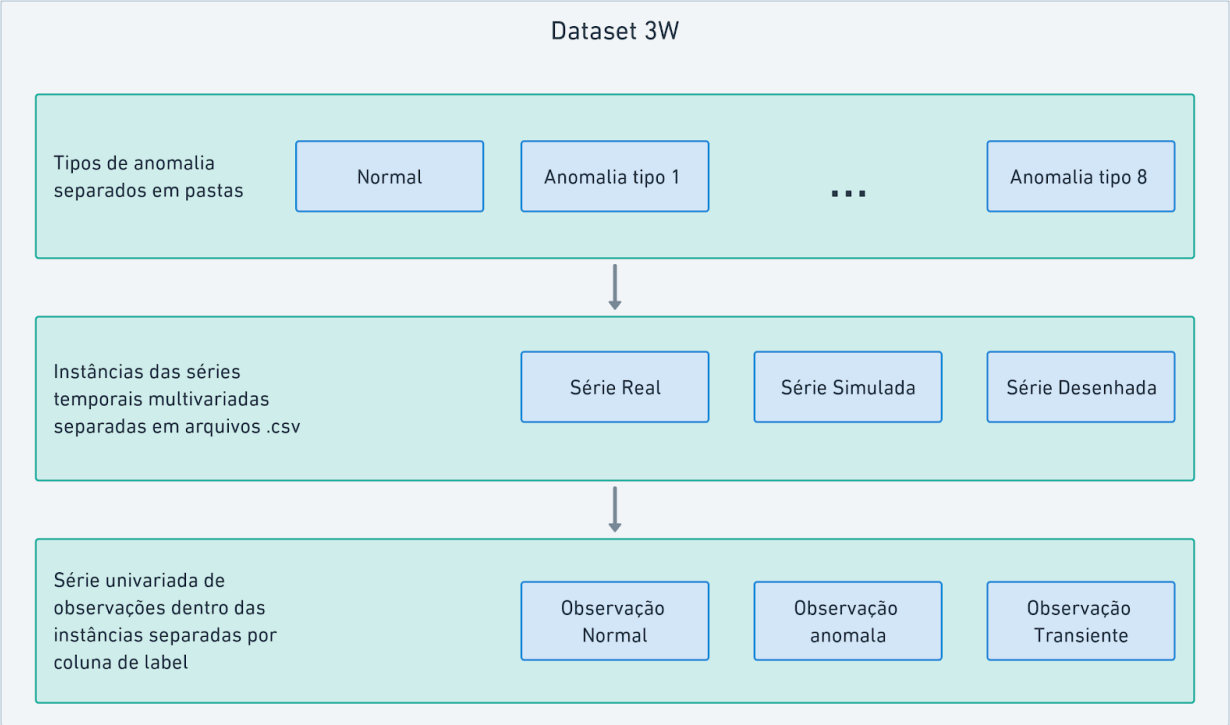
Nesta seção, é analisada a estrutura do *Dataset 3W* e do benchmark proposto por (Vargas, 2019), assim como uma breve descrição das categorias de anomalias a serem detectadas. São explicados conceitos relevantes sobre técnicas de aprendizado de máquina utilizadas durante este trabalho. Por fim, também é realizado um resumo dos trabalhos prévios encontrados na literatura que se utilizaram do *Dataset 3W*.

2.1 DATASET 3W

Como detalhado em (Vargas, 2019), o dataset é estruturado da seguinte forma.

- Separação das MTS por tipo de anomalia
    - Pastas rotuladas com números de 0 a 8, correspondendo ao tipo único de anomalia (ou ausência de, no caso do 0) encontrada durante um período de tempo dentro de todas as MTS da pasta
- Separação das MTS por origem
    - Arquivos .csv são nomeados com as palavras *real, simulated* ou *drawn*, indicando a origem das informações. As MTS reais são dados provenientes dos sensores instalados nos poços petrolíferos, simulações são séries simuladas por profissionais da área em ferramenta computacional, e as desenhadas foram criadas a mão por especialistas da área e digitalizadas em ferramenta desenvolvida pelos próprios pesquisadores. O poço de origem também é identificado pelo nome do arquivo .csv
- Separação das observações na MTS por coluna de rótulo
    - Cada série multivariada contém uma série univariada na coluna de rótulo que separa cada observação em Normal, Transiente e Estado Estável de Anomalia.

FIGURA 1 - Estrutura do Dataset 3W



FONTE: O autor (2022)

2.2 BENCHMARK PROPOSTO

O benchmark estabelecido em (Vargas, 2019) determina as seguintes regras:

- Apenas instâncias reais com anomalias de tipos que têm períodos normais (1, 2, 5, 6,7 e 8) maiores ou iguais a vinte minutos devem ser utilizadas. Aquelas com rótulos diferentes não podem ser utilizadas. Em outras palavras, apenas arquivos com extensão CSV salvos em diretório cujo nome é um desses tipos podem ser utilizados.
- Múltiplas rodadas de treinamento e validação devem ser realizadas. O número de rodadas deve ser igual ao número de instâncias. Em cada rodada, o seguinte cenário deve ser implementado. As amostras utilizadas para treinamento ou validação devem ser extraídas de apenas uma instância. Parte das amostras negativas devem ser utilizadas no treinamento e a outra parte na validação. Todas as amostras positivas devem ser utilizadas apenas na validação. Portanto, uma técnica de aprendizagem de classe única deve ser utilizada. O

conjunto de validação deve ser composto pelo mesmo número de amostras de cada classe (positiva e negativa).

- Em cada rodada, precisão, sensibilidade e *F1 score* devem ser computadas, mas outras métricas também podem ser consideradas. Valor médio e desvio padrão de cada métrica entre todas as rodadas devem ser apresentados. Valor médio da medida F deve ser considerado a principal métrica de desempenho, por estabelecer uma relação de compromisso entre precisão e sensibilidade.

Cabe aqui uma breve definição das métricas principais usadas pelo benchmark.

- Precisão (P): Razão entre o número de amostras anômalas estimadas corretamente e o número total de amostras anômalas estimadas
- Sensibilidade (S): Razão entre o número de amostras anômalas estimadas corretamente e o total de amostras anômalas
- Medida F (F1 *Score*): Média harmônica entre a precisão e a sensibilidade

$$F1 = \frac{2*P*S}{P+S}$$

2.3 ESPECIFICAÇÃO DAS ANOMALIAS

Aqui são brevemente explicadas as anomalias contidas no dataset e seus respectivos códigos. Mais detalhes, assim como exemplos ilustrados com as séries temporais podem ser encontrados em (Vargas, 2019)

Tabela 1 - Códigos das anomalias no dataset 3W e suas descrições

Código	Descrição
0	Operação Normal
1	Aumento Abrupto de BSW (*Basic Sediment and Water*)
2	Fechamento Espúrio de DSHV (*Downhole Safety Valve*)
3	Intermitência Severa
4	Instabilidade de Fluxo
5	Perda Rápida de Produtividade
6	Restrição Rápida na CKP (*Choke* de Produção)
7	Incrustação em CKP (*Choke* de Produção)
8	Hidrato na Linha de Produção

Fonte: Vargas, 2019

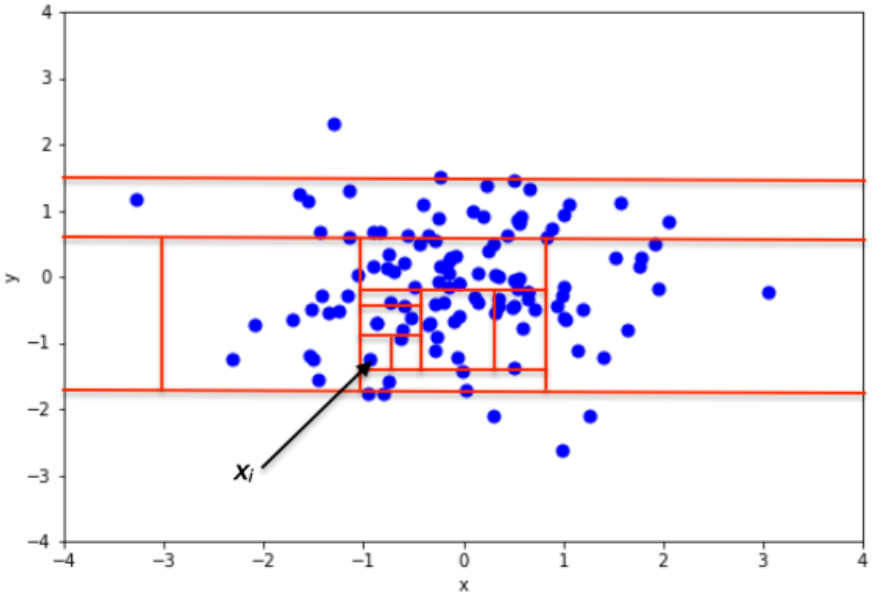
2.4 TÉCNICAS UTILIZADAS

Nesta seção são explicados os fundamentos dos classificadores e técnicas de aprendizado de máquina utilizados durante este trabalho.

2.4.1 Isolation Forest

Isolation Forest é um algoritmo usado para detecção de anomalias. O modelo se baseia no conceito de que pontos anômalos são mais fáceis de serem isolados dos demais, e diferentemente de outras abordagens, ele tenta encontrar estes pontos isolados ao invés de modelar as observações normais. O algoritmo recursivamente faz várias partições aleatórias do dataset até isolar algum ponto, no fim do processo os pontos que necessitam de menos partições são considerados anômalos. A figura 2 demonstra o algoritmo isolando um ponto Xi não anômalo em um espaço 2D por meio de diversas partições, já na figura 3 o mesmo algoritmo necessita de significantemente menos partições para encontrar um ponto anômalo distante dos demais.

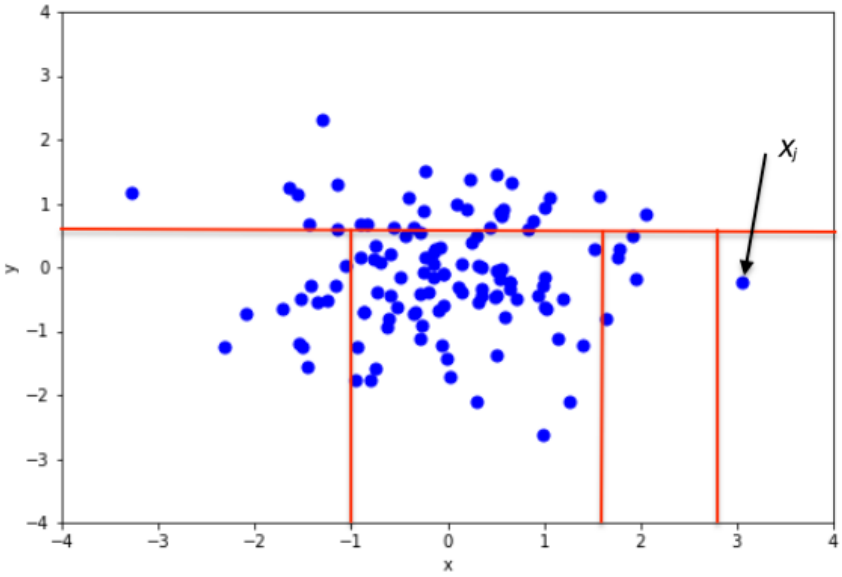
FIGURA 2 - Isolamento de ponto não anômalo por isolation Forest



FONTE: *Isolation Forest. In: WIKIPÉDIA: a enciclopédia livre.*[1]

[1] *Disponível em: https://en.wikipedia.org/wiki/Isolation_Forest*

FIGURA 3 - Isolando ponto anômalo por isolation forest.



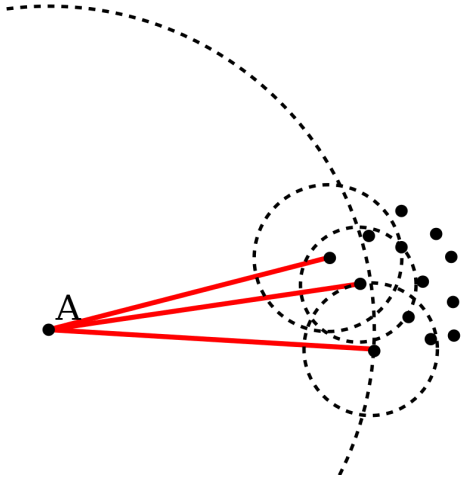
FONTE: *Isolation Forest. In: WIKIPÉDIA: a enciclopédia livre.*[2]

2.4.2 Local Outlier Factor

Esse algoritmo de detecção de anomalias se baseia no conceito de que pontos anômalos estão situados em áreas de baixa densidade quando comparados aos normais. Para encontrar estas densidades, o algoritmo calcula as distâncias de um ponto A aos seus vizinhos, e as distâncias destes vizinhos aos demais. Dessa maneira, se o ponto A está relativamente distante de um grupo de pontos mais próximos entre si, é possível concluir que ele é uma anomalia. A figura 4 exemplifica este processo.

[2] *Disponível em: https://en.wikipedia.org/wiki/Isolation_Forest*

FIGURA 4 - Ideia básica de comparação de densidade local entre pontos.



FONTE: Local Outlier Factor. *In: WIKIPÉDIA: a enciclopédia livre.*[3]

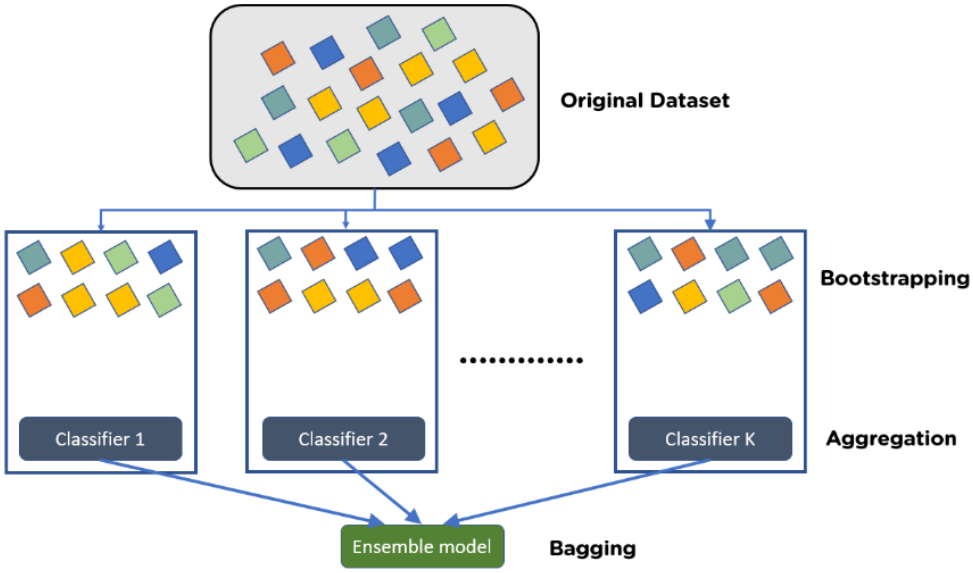
2.4.3 Feature Bagging

No contexto de aprendizado de máquina, um *ensemble* é basicamente uma combinação de modelos base diferentes entre si com o objetivo de criar um modelo final com melhor performance. *Bagging*, também conhecido como *bootstrap aggregating*, é uma técnica usada para criar *ensembles* de modelos de aprendizado de máquina.

Bootstrapping é um método estatístico de amostragem de dados por seleção aleatória para estimar o comportamento de uma população. No contexto de aprendizado de máquina, são amostradas aleatoriamente sem substituição um número aleatório X de colunas (*features*) N vezes, gerando assim N datasets diferentes entre si. São então treinados modelos de aprendizado de máquina em cada *dataset*, o que resulta em N modelos distintos. Aggregation corresponde ao processo de combinação destes N modelos para gerar uma única saída, que neste trabalho, cada amostra recebe N classificações de anomalia (uma por modelo base) e a classificação final é obtida por voto majoritário. Um exemplo deste processo é encontrado na figura 5.

[3] *Disponível em: https://en.wikipedia.org/wiki/Local_outlier_factor.*

FIGURA 5 - Esquemático de um ensemble por feature bagging.



FONTE: O autor (2022)

2.4.4 Grid Searching

Grid searching diz respeito a uma técnica de otimização de hiperparâmetros de modelos de aprendizado de máquina. A técnica consiste em testar extensivamente combinações de parâmetros para encontrar aquela que entrega os melhores resultados, utilizada neste trabalho primariamente para encontrar o número ideal de modelos usados em ensembles por feature bagging.

2.5 TRABALHOS PRÉVIOS

Desde sua concepção, o *dataset* 3W foi usado por diversos pesquisadores. A maioria das aplicações em detecção de anomalias encontradas não realizaram os benchmarks propostos em (Vargas, 2019) o que torna difícil uma comparação direta, porém ainda é possível a identificação de abordagens e técnicas promissoras. Aqui serão ressaltadas algumas destas aplicações.

2.5.1 Trabalhos que realizaram o benchmark proposto

Os autores em (Júnior et al., 2020) realizaram a amostragem das instâncias com janela deslizante com geração de até 15 amostras com 180 observações cada. Além das demais etapas de pré-processamento especificadas no benchmark. Após isso, as mesmas features usadas no benchmark são extraídas. Para classificação foram usados 4 modelos distintos, Local Outlier Factor (LOF), *Isolation Forest* (IF), Envelope Elíptico (EE) e *One-Class Support Vector Machine (One-class SVM)*. Foi então realizada a otimização dos classificadores usando a função *ParameterGrid* da biblioteca *scikit-learn.* O melhor resultado médio obtido foi o do classificador LOF (F1 score = 0.882 +- 0.126), seguido pelo FI (F1 score = 0.743 +- 0.179). Ambos melhores que os obtidos em (Vargas, 2019)

A primeira customização realizada por (Nascimento, 2021) é a de inserção de 100 atrasos em cada observação. Os autores também reduzem a dimensionalidade dos dados utilizando *autoencoders* em cascata e *principal component analysis (PCA)*. Modelos de *one-class SVM* e IF são alimentados com os dados reduzidos, com e sem atrasos. Os hiperparâmetros utilizados foram os já otimizados em (Júnior et al., 2020). Foi obtido o melhor desempenho médio de F1 Score (0.8345+- 0.1329) no classificador de IF com os atrasos e sem a redução de dimensionalidade. e foi observado que na maioria dos casos testados, o classificador IF desempenhou melhor que o *one-class SVM* .

2.5.2 Trabalhos que não realizaram o benchmark proposto

Em (Sobrinho et al., 2020), propõe-se um sistema para monitoramento em tempo real de anomalias. Os autores utilizaram uma abordagem conjunta de modelagem analítica e de aprendizado de máquina. Inicialmente são estabelecidas regras de comportamento esperado dos sensores, essas regras são então implementadas por um modelo de árvore de decisão para selecionar os dados a serem alimentados em uma rede neural, que quando treinada estabelece um score crítico de anomalia. Novos dados são então introduzidos à rede, que retorna um score de anomalia, a detecção acontece quando este retorno é igual ou superior ao score crítico estabelecido.

Em (Oliveira, 2020) utilizaram-se 3 séries temporais contidas no dataset 3W para treinar modelos Auto Regressivos de Média Móvel e (ARMA) e Auto Regressivos Integrados de Média Móvel (ARIMA) para prever valores futuros. Os hiperparâmetros são determinados a partir de dados iniciais, previsões futuras são realizadas e então

calcula-se um score de erro. Caso o erro ultrapasse um valor pré-determinado é detectada uma anomalia, as previsões são então realimentadas no treinamento a fim de absorver a nova tendência. Caso contrário, os novos dados são incorporados à série e repete-se o processo.

Em (Turan & Jaschke, 2021) foram excluídos os dados das instâncias desenhadas. As features de P-CKGL e T-CKGL do dataset foram eliminadas. Downsample das séries de uma observação por segundo para uma observação a cada 10 segundos foi realizada. Janelamento foi utilizado para amostragem, e sobre estas amostras, as medidas normalizadas de Média, Variância, distorção e curtose, assim como suas transformadas de Fourier foram utilizadas. Máximos, mínimos, medianas, quantis, coeficiente de variação, variação média e média da segunda derivada também foram incluídos. Algoritmos de seleção de *features* foram utilizados. Os modelos testados foram Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), Linear Support Vector Classifier (SVC), Logistic Regression, Decision Trees, Random Forest (RF) e AdaBoost (ADA). Um *grid search* para otimização de hiperparâmetros foi realizado. A métrica de classificação principal utilizada foi o de valor F1, calculado usando validação cruzada. Foi obtido desempenho máximo de 91% com o classificador RF, e foi constatado que o uso de *feature selection* não melhorou de maneira significativa os classificadores.

Marins et al. (2020) apresentaram uma abordagem de extração de medidas da média, desvio padrão, distorção e curtose por janelamento. Redução de dimensionalidade foi realizada utilizando PCA. 3 Experimentos foram propostos, todos utilizando *Random Forest* (RF): classificação de classe única para discriminar qualquer tipo de evento anômalo do normal, multiplos classificadores binários especializados para discriminar individualmente cada tipo de anomalia no dataset do normal, e um único classificador multiclasse para discriminar em conjunto todos os tipos de anomalias. Diferentes métricas de acurácia foram calculadas utilizando validação cruzada. No primeiro experimento foi obtida acurácia de cerca de 98%, no segundo, todas as classes obtiveram acurácias acima de 97%, e no terceiro o classificador geral obteve acurácia de 97%.

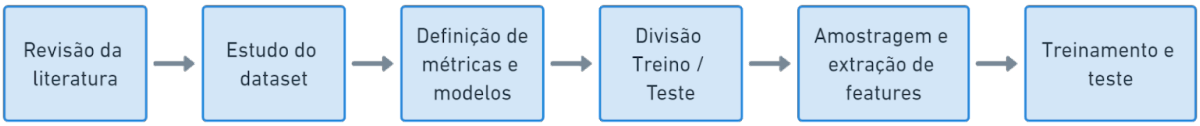
Em Figueiredo et al, (2021), foi utilizado aprendizado não supervisionado para detecção. Os autores selecionaram 5 instâncias específicas e utilizaram os modelos C-AMDATS, *Luminol Bitmap*, SAX-REPEAT, K Nearest Neighbors (k-NN), *Bootstrap* e RRCF otimizados via grid search para a classificação. O melhor *score* F1 médio entre os 5 casos foi 98,9%, obtido pelo classificador C-AMDATS, que desempenhou consideravelmente melhor que os demais.

Em sua tese, Carvalho, (2021) focou em classificar somente a classificar anomalias do tipo 4 (instabilidade de fluxo). O autor propõe uma nova técnica de split nos dados para realizar a validação cruzada considerando o poço de origem das observações. Um *grid search* é realizado para otimização dos classificadores *Adaptive Boosting*, *Extreme Learning Machine, Gaussian Naive Bayes,* k-NN, QDA, LDA, *random forest*, e SVM. Para seleção de características foram usados *Sequential feature selection, hybrid ranking wrapper e genetic algorithm.* Utilizando a técnica de *split* padrão foi obtido o melhor *score* F1 de 99% com o classificador RF, semelhante ao obtido por (Marins et al., 2020). O novo split resultou em piores performances em todos os classificadores, o melhor destes sendo QDA com 67%.

3 MATERIAL E MÉTODOS

O projeto foi dividido nas seguintes etapas

FIGURA 6 - Etapas da Metodologia e Desenvolvimento



FONTE: O autor (2022)

3.1 REVISÃO DA LITERATURA

Nesta etapa, identificou-se as principais topologias dos problemas detecção de anomalias encontradas na literatura e suas respectivas soluções. Quais as vantagens e desvantagens de cada método, assim como os possíveis cenários de aplicação.

Foram então coletadas todas as referências que continham citações à (Vargas, 2019). Suas metodologias e resultados foram analisados individualmente, e as principais técnicas de modelagem, processamento e os classificadores utilizados foram categorizados. Isso permitiu a criação de um portfólio para embasar futuras decisões.

Feita a categorização inicial, as referências estudadas foram reclassificadas entre as que utilizaram ou não o benchmark proposto por (Vargas, 2019). Isso permitiu focar nos métodos mais aplicáveis ao problema principal, e ao mesmo tempo não descartar possíveis abordagens efetivas em contextos distintos.

3.2 ESTUDO DO DATASET 3W

Para começar com os estudos foi necessário importar o dataset para um ambiente onde ele poderia ser manipulado pelos programas em python. A ferramenta de desenvolvimento escolhida para o projeto foi o Google *Collaboratory,* devido a facilidade de criar ambientes python e as funções de compartilhamento de código. O google drive foi escolhido como plataforma para hospedar o dataset, pois ele integra bem com o *Collaboratory.* Modificações iniciais na importação de bibliotecas e nos métodos construtores de certos objetos tiveram de ser feitas devido a alterações em

versões mais novas nas bibliotecas *sci-kit Learn* e *TSFresh* para rodar o código disponibilizado por (Vargas, 2019).

Durante esta etapa a estrutura do dataset 3W foi mapeada, um esquemático pode ser encontrado na figura 1. Análises exploratórias foram realizadas a fim de compreender o comportamento dos dados. Tópicos analisados incluíram: Tipos de anomalias presentes, distribuição dos tipos, uso de casos simulados e desenhados e quantas instâncias se qualificam para os benchmarks.

3.3 DEFINIÇÃO DE MÉTRICAS E MODELOS

A escolha das métricas de desempenho em um projeto envolvendo aprendizado de máquina são essenciais para definir tanto os objetivos a serem atingidos quanto uma metodologia adequada. Tendo isso em mente, (Vargas, 2019) em seu benchmark propõe a utilização do F1 *Score* como métrica principal para avaliação dos classificadores, permitindo o uso da Sensibilidade e Precisão como métricas secundárias para fim de outras comparações.

Devido a natureza do problema de detecção de anomalias e a necessidade de uso de classificadores de classe única proposta pelo benchmark descrito em 2.2, os modelos escolhidos foram limitados a essa topologia. Os modelos selecionados foram os já testados na literatura e que apresentaram os melhores desempenhos, Isolation Forest e Local Outlier Factor. Como diferencial, nesta dissertação também é proposto o teste de diversas combinações de ensembles por Feature Bagging dos classificadores mencionados, assim como o uso de novas técnicas de amostragem a serem descritas na seção 3.5

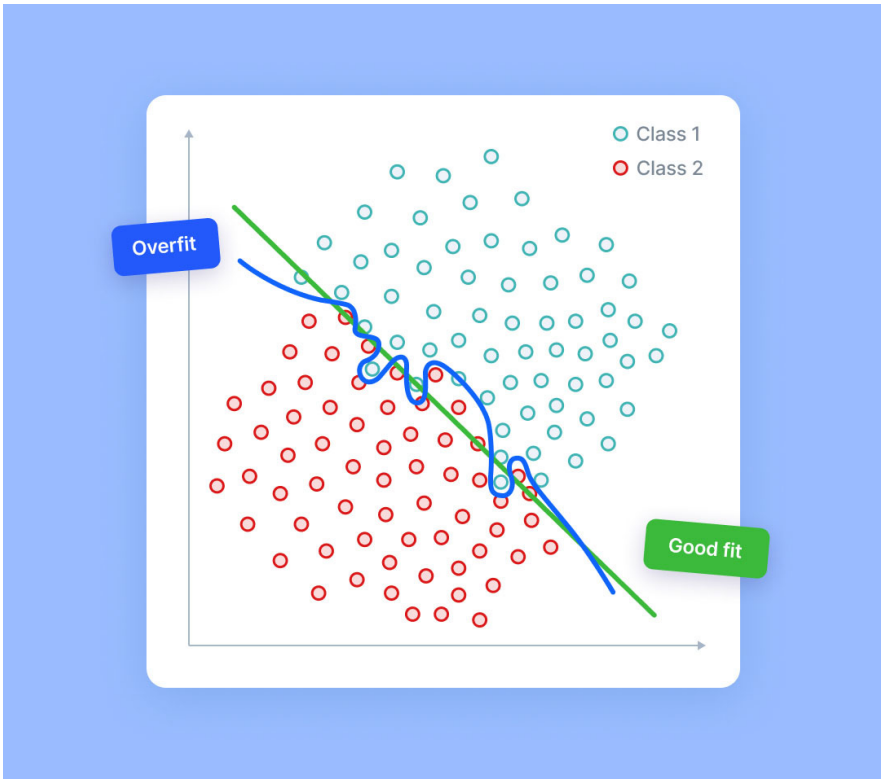
Os hiperparâmetros utilizados para os modelos individuais foram os mesmos já otimizados por (Júnior et al., 2020). Para determinar o número de classificadores utilizados no ensemble por feature bagging foi realizado um grid search, onde foram testados de 1 a 25 modelos.

3.4 DIVISÃO TREINO/TESTE

Em aplicações de modelos de aprendizado de máquina é comum a divisão dos dados em sets, um dedicado ao treinamento dos modelos e outro somente para testar o desempenho das classificações. Isso é feito para validar os resultados finais em dados não vistos durante o treino a fim de imitar uma situação real e evitar *overfitting,*

um problema que pode ocorrer quando são encontrados padrões muito específicos nos dados que não podem ser generalizados, distorcendo as métricas de desempenho. Um exemplo de overfitting pode ser visto na figura 7 Neste trabalho, foi utilizada a mesma divisão de dados de 0.6 (60% treino 40% teste), como proposta por (Vargas, 2019)

FIGURA 7 - Exemplo de *overfitting*



FONTE: Ilustração por Pragati Baheti em Blog V7 Labs[4]

3.5 AMOSTRAGEM E EXTRAÇÃO DE FEATURES

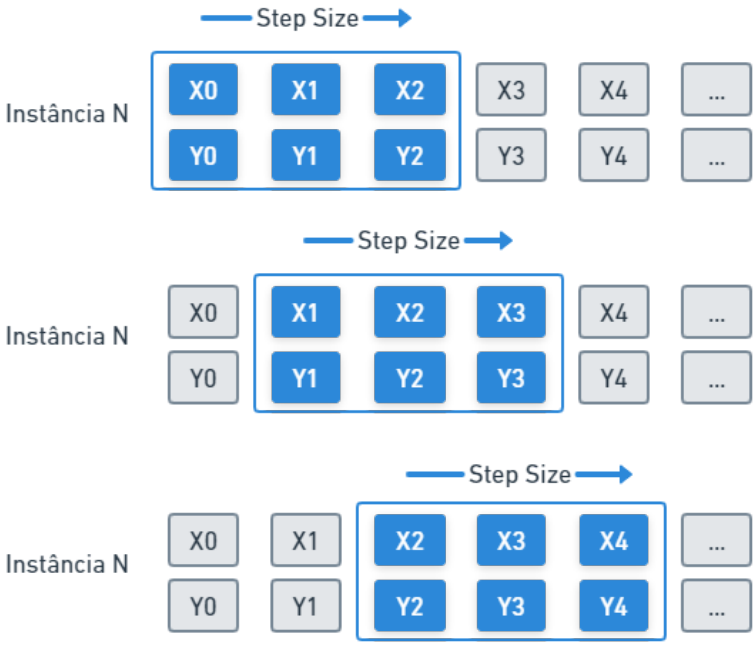
Devido a redundâncias no código original durante a lógica de checagem das regras propostas na seção 2.2 os dados originais não são usados em totalidade, o que resulta em poucas amostras de treino para alimentar os classificadores. Como alternativa, foi desenvolvida uma nova amostragem com o uso da função *roll_time_series* da biblioteca *tsfresh* para aumentar a quantidade de pontos de treinamento e teste dos modelos. O desempenho dos classificadores foi medido usando ambas as técnicas.

Em ambos os casos uma abordagem de janela deslizante (exemplificado na figura 8) foi utilizada para realizar a amostragem das instâncias (séries multivariadas),

[4] Disponível em: https://www.v7labs.com/blog/overfitting

com cada amostra contendo somente observações normais ou anômalas. Em estados estáveis de anomalia, as primeiras observações foram privilegiadas na amostragem para os sets de validação.

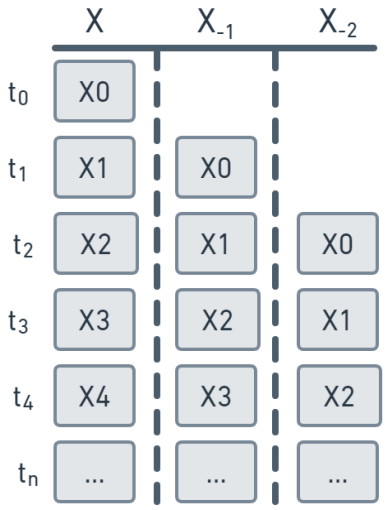
FIGURA 8 - Exemplo de janela deslizante



FONTE: O autor (2022)

Em (Nascimento, 2021) foi constatado que a criação de 100 colunas de atraso sobre cada variável, melhorou o desempenho da classificação, esse processo é ilustrado pela figura 9. Neste trabalho foram testadas e comparadas ambas as implementações, com e sem atrasos.

FIGURA 9 - Exemplo de colunas de atrasos



FONTE: O autor (2022)

Uma vez calculados os atrasos, sobre cada janela das séries temporais, foram calculadas as seguintes medidas: Mediana, média, desvio padrão, variância, raiz quadrada média, máximo e mínimo. Os dados foram então normalizados, passando a ter média zero e variância unitária, diminuindo assim o efeito da diferença das escalas das variáveis na classificação. Amostras cujas variáveis continham mais que 10% de valores ausentes ou que estavam estáticas (desvio padrão menor que 1%) foram descartadas. Valores ausentes restantes foram substituídos por 0.

3.6 TREINAMENTO E TESTE

Em cada rodada, modelos são alimentados com os dados de teste já pré-processados e amostrados, de acordo com as regras descritas na seção 2.2 e os processos descritos em 3.4 e 3.5, após isso, são feitas as previsões sobre os dados de teste. As previsões são comparadas às categorias de anomalias reais anotadas no dataset, obtendo-se assim as métricas de F1 score médio e desvio padrão (descritas na seção 3.3) de cada modelo.

Ambos modelos base e ensembles foram treinados utilizando tanto a amostragem original como alternativa. Em ambos os casos também foi testado o impacto da adição de atrasos no desempenho dos classificadores.

4 APRESENTAÇÃO DE RESULTADOS

Nesta seção são apresentados os valores médios de F1 *Score*, assim como desvio padrão entre as instâncias para os testes com amostragem padrão e amostragem alternativa para os diversos *ensembles* de modelos por *feature bagging* descritos em 2.4.3.

4.1 AMOSTRAGEM PADRÃO

Utilizando o método de sampling padrão descrito na seção 3.5 o classificador LOC obteve o mesmo desempenho encontrado por (Júnior et al., 2020) como esperado. Devido a alterações no funcionamento dos hiperparâmetros na biblioteca SKlearn utilizada para implementação do algoritmo Isolation Forest, este obteve um F1 score de 0.699, um pouco abaixo do encontrado por (Vargas, 2019) de 0.727, porém o desvio padrão permaneceu constante em 0.182.

Os resultados do grid search para o número de modelos do ensemble (n) utilizando LOC como classificador base podem ser encontrados na tabela 3. O melhor resultado de F1 (0.856) foi obtido com n=2, com os scores do top 5 em torno de 0.850, e desvio padrão de 0.185. Os resultados do grid para o modelo base Isolation Forest se encontram na tabela 2, onde percebe-se que os F1 scores do top 5 ficaram em torno de 0.715 e desvio padrão de 0.19, com melhor desempenho no ensemble com n=2 de 0.720. Em ambas as tabelas são apresentados os 15 primeiros ordenados por F1 score.

Para os ensembles por feature bagging com LOC houve uma piora tanto nos scores de F1 médios quanto no desvio padrão quando comparado ao modelo original utilizando todas as *features* ao mesmo tempo. Isso indica que um maior número de colunas agrega informação útil para a classificação, o que é corroborado pelos resultados de (Júnior et al., 2020), onde a utilização de seleção de features não resultou em melhoras no desempenho. Nos ensembles utilizando Isolation Forest, houve uma pequena melhora nos valores médios de F1 e o desvio padrão permaneceu constante.

Número de Classificadores LOC	F1	STD
n = 18	0.889	0.129
n = 10	0.889	0.131
n = 6	0.887	0.129
n = 14	0.887	0.129
n = 17	0.887	0.140
n = 9	0.884	0.125
n = 5	0.884	0.123
n = 16	0.884	0.128
n = 2	0.884	0.125
n = 21	0.884	0.125
n = 23	0.882	0.127
n = 11	0.882	0.128
n = 13	0.882	0.136
n = 4	0.882	0.128

Tabela 2 - Resultados do Ensemble de modelos IF com sampling original

Número de Classificadores IF	F1	STD
n = 2	0.720	0.189
n = 6	0.715	0.201
n = 8	0.715	0.186
n = 7	0.713	0.187
n = 13	0.711	0.199
n = 9	0.706	0.218
n = 11	0.699	0.185
n = 14	0.699	0.215
n = 1	0.699	0.183
n = 4	0.697	0.183
n = 5	0.694	0.137
n = 10	0.688	0.201
n = 3	0.688	0.201
n = 12	0.669	0.219

Tabela 3 - Resultados do ensemble de modelos LOC com sampling original

4.2 AMOSTRAGEM ALTERNATIVA

Os mesmos testes foram refeitos, porém com a amostragem alternativa proposta em 3.5. Percebe-se que ocorreu uma pequena piora nos scores F1 médios e desvio padrão do modelo LOC, porém utilizando o algoritmo de *Isolation Forest* obteve-se uma melhora de 13% no score F1, sem impacto significativo no desvio padrão.

Os resultados do *grid search* para o número de modelos do *ensemble* (n) utilizando IF como classificador base podem ser encontrados na tabela 5. Percebe-se que não ocorreram melhoras nos *scores* F1 dos *ensembles* quando comparados ao modelo base usando o mesmo método de amostragem. Os resultados do grid para o modelo base LOC são encontrados na tabela 6, onde percebe-se que o melhor ensemble (F1 = 0.856 e DP = 0.183) foi obtido com n = 2. Os resultados para o uso de grid.

Para o teste de adição de colunas de atraso, percebe se que houveram pioras tanto nos valores médios de F1 score, quando nos desvios padrões entre as instâncias

Número de Classificadores LOC	F1	STD
n = 2	0.856	0.183
n = 7	0.851	0.189
n = 3	0.850	0.188
n = 4	0.849	0.190
n = 10	0.848	0.186
n = 11	0.848	0.189
n = 15	0.848	0.189
n = 12	0.847	0.188
n = 13	0.847	0.190
n = 5	0.846	0.189
n = 24	0.846	0.190
n = 20	0.845	0.191
n = 9	0.845	0.188
n = 8	0.845	0.186

Tabela 4 - Resultados do ensemble de modelos LOC com amostragem alternativa

Número de Classificadores IF	F1	STD
n = 12	0.856	0.185
n = 4	0.856	0.184
n = 2	0.854	0.191
n = 5	0.854	0.186
n = 8	0.854	0.195
n = 14	0.853	0.194
n = 16	0.853	0.196
n = 1	0.852	0.187
n = 20	0.852	0.195
n = 13	0.852	0.197
n = 23	0.852	0.197
n = 24	0.851	0.197
n = 18	0.851	0.198
n = 7	0.851	0.196

Tabela 5 - Resultados do ensemble de modelos IF com amostragem alternativa

Número de Classificadores IF	F1	STD
n = 6	0.842	0.233
n = 22	0.842	0.241
n = 11	0.840	0.231
n = 17	0.838	0.239
n = 16	0.838	0.235
n = 1	0.837	0.237
n = 12	0.836	0.245
n = 8	0.836	0.228
n = 15	0.835	0.230
n = 3	0.834	0.241
n = 13	0.834	0.235
n = 9	0.833	0.245
n = 23	0.832	0.243
n = 2	0.832	0.246

Tabela 6 - Resultados do ensemble de modelos LOC com amostragem alternativa e adição de 100 atrasos

Número de Classificadores LOC	F1	STD
n = 15	0.803	0.271
n = 7	0.802	0.267
n = 17	0.802	0.267
n = 27	0.802	0.267
n = 25	0.802	0.267
n = 29	0.802	0.272
n = 40	0.802	0.272
n = 14	0.802	0.269
n = 11	0.801	0.270
n = 9	0.801	0.271
n = 5	0.801	0.266
n = 35	0.801	0.269
n = 21	0.801	0.269
n = 18	0.801	0.269

Tabela 7 - Resultados do ensemble de modelos IF com amostragem alternativa e adição de 100 atrasos

5 CONCLUSÕES E RECOMENDAÇÕES

Neste trabalho, observou-se que a implementação de ensembles por feature bagging de modelos *Isolation Forest* e *Local Outlier Factor* não resultou em melhoras no valor máximo atingível de *F1 score* para as previsões de anomalias no *Dataset3W* utilizando de 2 à 25 modelos base. A introdução de uma nova metodologia de amostragem melhorou consideravelmente o desempenho dos classificadores *Isolation Forest*, tanto individualmente quanto em *ensemble*. A nova amostragem eliminou a diferença de desempenho, previamente observada na literatura por (Júnior et al., 2020) e durante este trabalho, entre as duas técnicas de classificação (LOC e IF), enquanto também causando uma piora na classificação por LOC. A utilização de atrasos combinada com a nova amostragem proposta resultou em pioras tanto nos valores médios de F1 score como desvio padrão entre as instâncias em ambos os classificadores. Isto demonstra que o menor número de amostras e features favorece o algoritmo *Local Outlier Factor*, o que pode indicar que as diferenças de densidade local são mais claras com menos pontos nas vizinhanças e em um espaço com dimensões reduzidas.

Neste trabalho foi explorada somente a técnica de ensemble de modelos de classificação por *feature bagging*, na literatura também são encontradas melhoras em resultados de problemas de detecção de anomalias utilizando outras técnicas de combinação de modelos, como Stacking. Uma possibilidade a ser testada em trabalhos futuros é a aplicação destas técnicas no contexto do dataset 3W.

REFERÊNCIAS

Carvalho, B. G. (2021). *Evaluating machine learning techniques for detection of flow instability events in offshore oil wells*. Universidade Federal do Espírito Santo.

Chandola, V., Banerjee, A., & Kumar, V. (n.d.). Anomaly Detection: A Survey. *ACM Computing Surveys*, *41*(3), 58. 10.1145/1541880.1541882

Figueiredo, I. S., Carvalho, T. F., Silva, W. S. D., Guarieiro, L. L. N., & Nascimento, E. G. S. (2021). *Detecting Interesting and Anomalous Patterns In Multivariate Time-Series Data in an Offshore Platform Using Unsupervised Learning*. Offshore Technology Conference.

Garcia, A. B., Conde, A., Mori, U., & Lozano, J. A. (2021). A Review on Outlier/Anomaly Detection in Time Series Data. *ACM Computing Surveys*, *54*(3), 33. 10.1145/3444690

Júnior, W. F., Vargas, R. E. V., Komati, K. S., & Gazolli, K. A. d. S. (2020). Detecção de anomalias em poços produtores de petróleo usando aprendizado de máquina. 10.48011/asba.v2i1.1405

Marins, M. A., Barros, B. D., Santos, I. H., Barrionuevo, D. C., Vargas, R. E.V., Prego, T. d. M., de Lima, A. A., de Campos, M. L.R., da Silva, E. A.B., & Netto, S. L. (2020, 9 5). Fault detection and classification in oil wells and production/service lines using random forest. *Journal of Petroleum Science and Engineering*, 197.

Nascimento, R. S. (2021). *DETECÇÃO DE ANOMALIAS EM POÇOS DE PRODUÇÃO DE PETRÓLEO OFFSHORE COM A UTILIZAÇÃO DE AUTOENCODERS E TÉCNICAS DE RECONHECIMENTO DE PADRÕES*. Universidade Federal de Lavras.

Oliveira, I. d. M. N. (2020, 12). *TÉCNICAS DE INFERÊNCIA E PREVISÃO DE DADOS COMO SUPORTE À ANÁLISE DE INTEGRIDADE DE REVESTIMENTOS*. Universidade Federal de Alagoas – UFAL.

Sobrinho, E. d. S. P., de Oliveira, F. L., dos Anjos, J. L. R., Gonçalves, C., Ferreira, M. V. D., Lopes, L. G. O., Lira, W. W. M., de Araújo, J. P. N., da Silva, T. B., & de Golveia, L. P.

(2020, 12 1). Uma ferramenta para detectar anomalias de produção utilizando aprendizagem profunda e árvore de decisão. *Rio Oil & Gas Expo and Conference, Rio de Janeiro*, 437.

Turan, E. M., & Jaschke, J. (2021, 7 1). Classification of undesirable events in oil well operation. *International Conference on Process Control (PC)*, 23.

Vargas, R. (2019). *ricardovvargas/3w_dataset: The first realistic and public dataset with rare undesirable real events in oil wells.* GitHub. Retrieved January 25, 2022, from https://github.com/ricardovvargas/3w_dataset

Vargas, R. E. V. (2019). *BASE DE DADOS E BENCHMARKS PARA PROGNÓSTICO DE ANOMALIAS EM SISTEMAS DE ELEVAÇÃO DE PETRÓLEO*. Universidade Federal do Espírito Santo.