INSTITUTO FEDERAL DO ESPÍRITO SANTO
ENGENHARIA DE CONTROLE E AUTOMAÇÃO

HIGOR LUIZ SOARES

DETECÇÃO DE ANOMALIAS NO SISTEMA APS DE CAMINHÕES DE CARGA UTILIZANDO ALGORITMOS DO TIPO ONE-CLASS

SERRA
2022

HIGOR LUIZ SOARES

DETECÇÃO DE ANOMALIAS NO SISTEMA APS DE CAMINHÕES DE CARGA UTILIZANDO ALGORITMOS DO TIPO ONE-CLASS

Trabalho de Conclusão de Curso apresentado à Coordenadoria do Curso de Engenharia de Controle e Automação do Instituto Federal do Espírito Santo como requisito parcial para a obtenção do título de Engenheiro de Controle e Automação.

Orientador: Prof. Dr. Daniel Cruz Cavalieri.

SERRA

2022

Dados Internacionais de Catalogação na Publicação (CIP)

S676d 2022

Soares, Higor Luiz
Detecção de anomalias no sistema APS de caminhões de carga utilizando algoritmos do tipo One Class / Higor Luiz Soares. - 2022.
40 f.; il.; 30 cm

Orientador: Prof. Dr. Daniel Cruz Cavalieri.
Monografia (graduação) - Instituto Federal do Espírito Santo, Coordenadoria de Automação, Curso de Graduação em Engenharia de Controle de Automação, 2022.

1. Inteligência artificial. 2. Aprendizado do computador. 3. Python (Linguagem de programação de computador). 4. Caminhões - Manutenção e reparos. I. Cavalieri, Daniel Cruz. II. Instituto Federal do Espírito Santo. III. Título.

CDD 006.3

Bibliotecária Rogeria Gomes Belchior - CRB6/ES 417

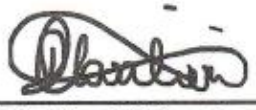
HIGOR LUIZ SOARES

DETECÇÃO DE ANOMALIAS NO SISTEMA APS DE CAMINHÕES DE CARGA UTILIZANDO ALGORITMOS DO TIPO ONE-CLASS

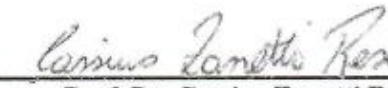
Trabalho de Conclusão de Curso apresentado como parte das atividades para obtenção do título de Bacharel em Engenharia de Controle e Automação, do curso de Engenharia de Controle e Automação do Instituto Federal do Espírito Santo.

Aprovado em 20 de OUTUBRO de 2022.

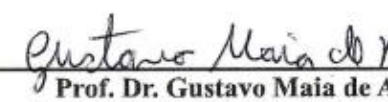
COMISSÃO EXAMINADORA



Prof. Dr. Daniel Cruz Cavalieri
Instituto Federal do Espírito Santo
Campus Serra



Prof. Dr. Cassius Zanetti Resende
Instituto Federal do Espírito Santo
Campus Serra



Prof. Dr. Gustavo Maia de Almeida
Instituto Federal do Espírito Santo
Campus Serra

AGRADECIMENTOS

Agradeço primeiramente a Deus por ter me dado a oportunidade conquistar esse sonho. Agradeço a minha mãe, Valdênia que me apoiou e nunca desistiu de mim. Também agradeço minha família e amigos por apoiarem nesta etapa da minha vida acadêmica.

Agradeço aos meus professores por todos os ensinamentos e conhecimentos compartilhados, pois este momento só é real devido a vocês. Um obrigado especial ao meu orientador, Daniel Cavalieri.

Agradeço ao Ifes por ser um excelente instituto de aprendizado e um lugar que me proporcionou muitas amizades, conhecimento e crescimento.

Muito obrigado!

Porque sou eu que conheço os planos que tenho para vocês, diz o Senhor, planos de fazê-los prosperar e não de causar dano, planos de dar a vocês esperança e um futuro.

Jeremias 29:11

RESUMO

Para os processos industriais ter controle das *outliers* ou anomalias é de extrema importância, pois assim é possível reduzir custos e diminuir o número de manutenções corretivas. Neste contexto, o cerne deste trabalho é avaliar o desempenho de diferentes algoritmos de aprendizado de máquina do tipo *one-class* (*One-Class SVM, Isolation Forest, Cluster-Based Local Outlier Factor e Empirical-Cumulative-Distribution-Based Outlier Detection*) para detecção de anomalias no sistema APS dos caminhões da marca Scania. Para tal, a empresa disponibilizou um conjunto de dados com 60.000 amostras para treinamento e 16.000 para testes. Os algoritmos serão responsáveis por detectar se a falha está relacionada com o sistema APS (*outliers)* ou não (*inliers)*. Com a metodologia utilizada, sem ajuste dos parâmetros, foi possível alcançar um custo de manutenção de $16.490 utilizando a técnica *Cluster-Based Local Outlier Factor*, detectando ainda 98,1% dos *outliers* e 91,7% dos *inliers*.

Palavras-chave: Sistema APS. Detecção de anomalias. Algoritmos *one-class.*

ABSTRACT

The approach studied in this work refers to the use of four techniques for detecting outliers, all being learning techniques of machine not supervised. For the industrial processes, the control of outliers or anomies is extremely important, as it is also possible to reduce costs and reduce the number of corrective maintenances. As part of this work, it is to assess or perform one-class algorithms (One-Class SVM, Isolation Forest, Cluster-Based Local Outlier Factor and Empirical-Cumulative-Distribution-Based Outlier Detection) for the detection of anomalies in the Scania brand trucks APS system. For that, the company made available a set of data with 60,000 samples for training and 16,000 for tests. The algorithms will be responsible for detecting if the fault is related to the APS system (outliers) or not (inliers). With the methodology used, without hyperparameter tuning, it was possible to reach a maintenance cost of $16,490 using the Cluster-Based Local Outlier Factor technique, which detected 98.1% two outliers and 91.7% two inliers.

Keywords: APS system. Anomaly detection. *One-class* algorithms.

LISTA DE ABREVIATURAS, SIGLAS

APS – *Air Pressure System*

FN – Falso negativo

FP – Falso positivo

IA – Inteligência Artificial

IoT - Internet of Things

LOF - *Local Outlier Factor*

PCA - *Principal component analysis*

SMOTE - *Synthetic Minority Over-sampling Technique*

SVM - *Support vector machine*

TI – Tecnologia da Informação

TPR - *True positive rate*

VN – Verdadeiro negativo

VP – Verdadeiro positivo

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

A primeira revolução industrial, que teve início no final do século XVII, teve como marco a mudança dos métodos artesanais para processos de fabricação mecanizados. Essas transformações revolucionaram não só a economia, mas também o dia a dia das pessoas. A partir de então a indústria tem evoluído de forma contínua.

A segunda revolução Industrial, que começou na segunda metade do século XIX, marca o início de uma nova era de industrialização que começou na Inglaterra e acabou se espalhando para outras nações como Estados Unidos, França, Rússia, Japão e Alemanha.

A segunda revolução Industrial marca uma nova etapa da civilização no que diz respeito aos avanços tecnológicos. A primeira fase da Revolução Industrial foi caracterizada pelo ferro, carvão e energia vaporizada. A segunda fase da Revolução Industrial é agora representada pelo aço, eletricidade e petróleo. A introdução de tecnologias durante esse período possibilitou a produção em massa, a automação do trabalho e o surgimento de várias indústrias.

A terceira revolução Industrial ocorreu em meados do século XX, a partir da década de 1950. Nesse momento, diversos campos do conhecimento começaram a sofrer mudanças em consequência do avanço tecnológico vivido nesse período e jamais visto anteriormente. Destacam-se nesse período as áreas de robótica, genética, informática, telecomunicações e eletrônica.

Além do desenvolvimento alcançado no setor industrial aliado ao desenvolvimento do campo científico, a terceira revolução Industrial mudou também as relações sociais e as relações entre o homem e o meio. As novas tecnologias desenvolvidas, nessa fase, possibilitaram que as informações fossem transmitidas cada vez mais rápido e estimularam a interação entre as pessoas do mundo todo.

A evolução das tecnologias da informação (TI) e sua integração nos processos de fabricação resultaram em benefícios em todos os setores. O avanço da capacidade tecnológica impulsionou a produtividade industrial, reduzindo custos de produção. Essa evolução proporcionou a criação de novos métodos de produção no setor

manufatureiro, tais inovações como, robótica, inteligência artificial, internet das coisas (do inglês, *Internet of Things* - *IoT*) e inteligência de dados, computação em nuvem, entre outras. A Indústria 4.0 - também conhecida como manufatura avançada - é a aplicação dessas tecnologias em um ambiente industrial com o objetivo de garantir competitividade ao negócio, otimizar a eficiência do processo de fabricação, agregar valor ao produto, racionalização do uso de recursos e customização de soluções tecnológicas.

> A indústria 4.0 assenta-se na integração de tecnologias de informação e comunicação que permitem alcançar novos patamares de produtividade, flexibilidade, qualidade e gerenciamento, possibilitando a geração de novas estratégias e modelos de negócio para a indústria (SACOMANO et al., 2018, p. 28).

Neste contexto, a inteligência artificial (IA) se tornou uma ferramenta poderosa e com grande aplicabilidade na indústria, como em análises preditivas, realidade aumentada, diagnóstico de desempenho, detecção de anomalias, entre outras. Sistemas baseados em IA além de aumentarem o desempenho, segurança e produtividade trazem recursos como disponibilidade de análises em tempo real. Assim, técnicas de inteligência artificial como redes neurais cada vez ganham mais espaço no contexto industrial devido a sua capacidade de gerar modelos baseados em conhecimentos prévios e sua capacidade de generalização.

O objetivo da IA é utilizar dispositivos ou métodos computacionais de forma similar à capacidade de raciocínio do ser humano (SACOMANO et al., 2018), e dessa forma, segundo Sellitto (2002), fornece modelos que auxiliam na tomada de decisão e controle, cujo embasamento é fundamentado em fatos do mundo real, conhecimento empírico e teórico mesmo que amparados por dados incompletos.

> Um processo de aprendizagem inclui a aquisição de novas formas de conhecimento: o desenvolvimento motor e a habilidade cognitiva (através de instruções ou prática), a organização do novo conhecimento (representações efetivas) e as descobertas de novos fatos e teorias através da observação e experimentação. Desde o início da era dos computadores, tem sido realizadas pesquisas para implantar algumas destas capacidades em computadores. Resolver este problema tem sido o maior desafio para os pesquisadores de inteligência artificial (IA). O estudo e a modelagem de processos de aprendizagem em computadores e suas múltiplas manifestações constituem o objetivo principal do estudo de aprendizado de máquinas (SANTOS, 2005, p. 10).

Na área de manutenção de equipamentos e predição de falhas, o uso de métodos de aprendizado de máquina tem se potencializado, pois permitem a criação de modelos

que descrevem o funcionamento dos equipamentos em condições específicas. Como resultado, os modelos desenvolvidos podem realizar previsões, classificações e até mesmo descobrir anomalias em um conjunto de dados de operação da máquina. Neste contexto, a abordagem a ser tratada neste trabalho é a detecção de anomalias no sistema APS de caminhões de carga utilizando *One-Class SVM, Isolation Forest, Cluster-Based Local Outlier Factor e Empirical-Cumulative-Distribution-Based Outlier Detection.*

O conjunto de dados utilizado, disponibilizado para um desafio no simpósio internacional organizado pelo IDA (*Intelligent Data Analysis*) (COSTA et al., 2016), cujo objetivo é desenvolver um bom modelo de previsão para julgar se um veículo enfrenta ou não uma falha iminente do componente específico ou não. Além disso, os participantes também deverão escrever um artigo descrevendo os métodos que usaram ao criar seu modelo preditivo. Os trabalhos com melhor performance foram publicados nos anais da IDA 2016. A Scania que forneceu os dados, também patrocinou um prémio para os melhores modelos.

O sistema em questão é o sistema de Pressão de Ar (APS) que gera ar pressurizado e é utilizado em diversas funções em um caminhão, como frenagem e troca de marchas. O *dataset* fornecido possui 170 atributos e é dividido em treinamento, com 60 mil dados, e teste, contendo 16 mil dados. O conjunto é dividido em duas classes: positiva, que significa que a falha ocorreu devido a componentes relacionados com o APS; e negativa que significa que a falha é não relacionada com os componentes APS. Os dados consistem em um subconjunto de todos os dados disponíveis que foram escolhidos por especialistas.

1.1 PROBLEMA DE PESQUISA

A detecção de *outliers* é uma etapa crucial na mineração de dados. Seu objetivo é encontrar amostras que se desviem significativamente dos padrões apresentados em um conjunto de dados. Neste sentido, este trabalho tem como objetivo responder o seguinte questionamento:

Qual método de previsão de anomalias (*One-Class SVM, Isolation Forest, Cluster-Based Local Outlier Factor e Empirical-Cumulative-Distribution-Based Outlier Detection.*) obterá o melhor desempenho e o menor custo de manutenção aplicado na classificação do sistema APS?

Neste sentido, por se tratar de um problema cuja base de dados é muito desbalanceada, é possível utilizar estratégias de detecção de anomalias do tipo *one-class* para identificar as classes corretamente, reduzindo o custo de manutenção dos veículos.

1.2 OBJETIVOS

1.2.1 Objetivo Geral

O objetivo geral deste trabalho de conclusão de curso é desenvolver algoritmos do tipo *one-class* para detecção de anomalias no sistema APS de caminhões de carga da Scania.

1.2.2 Objetivos Específicos

- Analisar o desempenho (sem ajuste dos parâmetros) dos algoritmos *One-Class SVM, Isolation Forest, Cluster-Based Local Outlier Factor e Empirical-Cumulative-Distribution-Based Outlier Detection*.
- Avaliar o custo de manutenção para resultado obtido nas previsões de cada um dos algoritmos propostos, sendo que a empresa Scania tem os seguintes custos de manutenção:
    - **Custo 1 ($10)**. Este Custo referente a uma vistoria desnecessária feita no caminhão em uma oficina;
    - **Custo 2 ($500)**. Custo de uma manutenção não prevista, visto que ocorrerá uma falha inesperada no caminhão.

- **Custo de Manutenção**. $Custo = custo1 * FN + Custo2 * FP$, onde FP é falso positivo e FN é falso negativo.

- Avaliar o tempo de processamento de cada modelo.

1.3 ESTRUTURA DO TRABALHO

O presente trabalho está organizado como mostrado a seguir:

Capítulo 1 (Introdução): onde foram expostos contextualização do tema, problemas de pesquisa, hipóteses, objetivos gerais e específicos e é apresentada uma revisão da literatura acerca dos trabalhos relacionados ao tema de detecção de anomalias em sistema APS de caminhões de carga da empresa Scania; Capítulo 2 (Fundamentação Teórica): conceitos relacionados aos tipos de técnicas abordadas no trabalho e além destes conceitos, fundamentos do processo de detecção de anomalias também são apresentados; Capítulo 3 (Metodologia): neste capítulo é realizada uma descrição detalhada do caminho percorrido para executar o processo de pesquisa do trabalho; Capítulo 4 (Resultados e discussões): este capítulo tem como objetivo interpretar os resultados encontrados no estudo e Capítulo 5 (Conclusão): por fim, este capítulo será usado para responder o problema central do trabalho e verificar se o trabalho cumpriu os objetivos de pesquisa.

1.4 ESTADO DA ARTE

No ano de 2016 foi publicado um desafio de previsão de aprendizado de máquina, cuja tarefa era desenvolver um modelo de predição de falhas relacionadas ao sistema APS. O conjunto de dados disponibilizados foi coletado por caminhões pesados no seu uso diário. Neste contexto, alguns trabalhos usando diferentes técnicas foram desenvolvidos, como apresentado a seguir.

No trabalho de Rawat (2020) foram utilizadas cinco técnicas para detecção de *outliers*: *Naivebayes*, que é um classificador que usa o método de aprendizado Bayesiano com independência condicional sobre o conjunto de dados de treinamento (MITCHELL,

1997, p. 177); *Logistic Regression,* ferramenta utilizada para classificação quando a variável dependente é binária, categórica ordenada ou mesmo categórica desordenada; *Random Forest*; *Support Vector Machine* e *k-Nearest Neighbours.* A primeira etapa desenvolvida pelos autores foi o pré-processamento dos dados ausentes e normalização da escala. Os valores faltantes foram tratados com a imputação da mediana.

Antes de realizar o aprendizado de máquina e resolver o desequilibro de classe, Rawat (2020) aplicaram uma técnica de redução de dimensionalidade denominada Análise de Componentes Principais (do inglês, *Principal Component Analysis* - PCA), reduzindo o número de características de 170 para 81. Com relação ao desequilibro de classe, o autor optou por usar a técnica de *oversampling* denominada SMOTE (*Synthetic Minority Oversampling Technique*), que cria dados sintéticos com base em um modelo gerado a partir da distância entre amostras. Novamente, foi usado o PCA e obtido uma ligeira melhoria nos resultados. Por fim o autor aplica os algoritmos de aprendizagem de máquinas, cujo melhor desempenho foi da técnica *Logistic Regression* que obteve um custo total de $15.940,00.

No trabalho de Cerqueira et al. (2016) foram utilizadas as técnicas de *Gradiente Boost* e *Random Forest,* com e sem o auxílio de *Metafeature Engineering*. A primeira tarefa realizada pelos autores foi solucionar o problema dos valores ausentes no conjunto dados. Para isso é empregado um filtro que exclui as características e exemplos que apresentam maior quantidade de valores ausentes, fazendo com que se reduza o número de dados e remova algumas informações ruidosas. Na segunda etapa é feita uma análise dos *outliers* para verificar o quão longe estes estão dos restantes dos dados, utilizando para isso três técnicas: *Boxplot Analysis*, LOF (*Local Outlier Factor*); e *Clustering-Based Outlier Ranking*. Em seguida é usado o *SMOTE (SyntheticMinority Over-samplingTechnique)*. Por fim, é gerado o conjunto de árvores de decisão impulsionadas, cujo melhor resultado obtido foi a técnica *Gradient Boost*em conjunto com *meta feature engineering* que teve um custo médio de $4.560,00.

Já os autores Gondek et al. (2016) em seu trabalho utilizaram a técnica de *Random Forest* para detecção de *outliers.* A primeira parte abordada no trabalho foi a limpeza dos dados, uma vez que no conjunto de dados existem muitos valores ausente de atributos. Neste caso os autores optaram pela substituição dos valores omissos pela

mediana. A normalização dos dados não foi necessária devido ao uso da técnica *Random Forest*. O resultado obtido foi a redução da média dos custos de $ 9,83 para $ 0,60 por caminhão, totalizando um custo de manutenção de $36.000,00.

No trabalho desenvolvido por Leão (2019) foi utilizado a técnica *one-class* SVM. Primeiramente a autora realizou o pré-processamento dos dados, dessa forma fez a substituição dos dados faltosos pelo valor médio de cada atributo. Em seguida foi feita a padronização usando a biblioteca “scikit-learn” do python. Posteriormente, foi aplicado o PCA para reduzir as dimensões por agregação e assim diminuir a dimensionalidade do conjunto de dados eliminando possíveis redundâncias. Por fim foi aplicado a técnica *one-class SVM* que obteve um custo total de manutenção de $15.290,00.

2 FUNDAMENTAÇÃO TÉORICA

2.1 MACHINE LEARNING

O ser humano ao longo de sua evolução vem utilizando inúmeras ferramentas para tornar suas atividades mais simples, precisa e que gaste menos energia e tempo. A criatividade levou a humanidade a desenvolver diversas máquinas, das quais hoje seria difícil imaginar um mundo sem elas, um exemplo é o computador. Com advento do computador, outras ferramentas foram desenvolvidas ou aprimoradas, como o aprendizado de máquina.

Para Géron (2019), Aprendizado de Máquina é a ciência (e a arte) da programação de computadores para que eles possam aprender com os dados. Em outra literatura tem-se a seguinte definição:

> Aprendizado de máquina é um ramo da Inteligência Artificial que envolve o projeto e desenvolvimento de sistemas capazes de mostrar uma melhoria de desempenho com base em suas experiências anteriores. Isso significa que, ao reagir à mesma situação, uma máquina deve apresentar uma melhoria de tempos em tempos. Com o *Machine Learning*, os sistemas de *software* são capazes de prever com precisão sem precisar ser programados explicitamente. O objetivo do *Machine Learning* é construir algoritmos que possam receber dados de entrada e usar a análise estatística para prever o valor de saída em um intervalo aceitável (CHAMBERLIN, 2020, p. 10).

Segundo Géron (2019), o aprendizado de máquina pode ser dividido de acordo com a quantidade e o tipo de supervisão que é recebido no decorrer do treinamento. O autor considera ter quatro categorias principais de aprendizado, são elas: supervisionado, não supervisionado, semissupervisionado e por reforço.

2.2 APRENDIZADO SUPERVISIONADO

No aprendizado supervisionado o conjunto de dados a serem treinados pelo algoritmo possuem as soluções desejadas, nomeadas de rótulos. O autor usa exemplo de classificação de *spam* na caixa de *e-mail.* A classificação é considerada uma tarefa típica do aprendizado supervisionado. No caso da imagem da figura 1 o filtro é treinado com uma quantidade considerável de exemplos de *e-mails* junto às classes (*spam* ou não *spam)* o algoritmo se encarregará de aprender a classificar novos *e-mails.*

Figura 1 – Aprendizado Supervisionado.

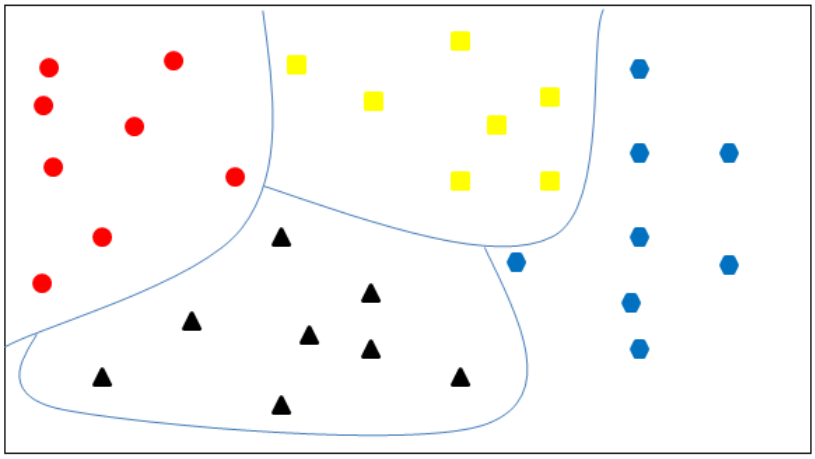
Fonte: Géron (2019)

2.3 APRENDIZADO NÃO SUPERVISIONADO

No aprendizado não supervisionado os dados de treinamentos não possuem rótulos (ou classes), ou seja, o sistema terá que aprender sem um instrutor. Segundo Guerón (2019) o aprendizado não-supervisionado pode ser dividido em três métodos: *clustering*, visualização e redução da dimensionalidade.

O *Clustering* é um método para a análise exploratória de dados utilizado para auxiliar a resolução de problemas de classificação (Backer, 1995). O objetivo da clusterização é definir agrupamentos entre dados similares. Por exemplo, para detectar grupos de usuários de uma determinada loja de livros *online,* o algoritmo fará todas as conexões sem auxílio do rótulo de saída. Ele poderá agrupar os usuários por suas características como: sexo, faixa etária, preferências, entre outras características. A figura 2 ilustra um exemplo de agrupamento.

Figura 2 – Aprendizado Não Supervisionado.



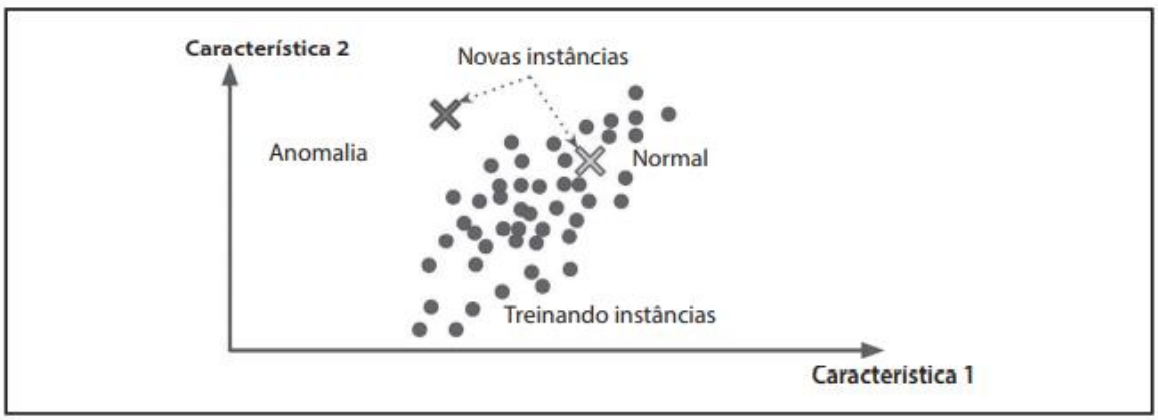
Fonte: Autoria Própria

Já os algoritmos de visualização são nutridos com uma vasta quantidade de informações complexas e não rotuladas e como produto exibem gráficos em 2D e 3D dos dados. A redução da dimensionalidade tem como objetivo principal combinar características correlacionadas para reduzir a complexidade do problema. No trabalho de Silva et al. (2018) foi utilizada a técnica de importância de característica (do inglês, *feature importance*) (BIAU, 2012), baseado no algoritmo de árvores de decisão, para aferir a influência das variáveis do processo no valor de saída, escolhendo assim as que tem maior influência na mesma. Desta forma o problema se torna mais simples, uma que foi reduzido o número de variáveis.

A detecção de anomalias é outra tarefa não supervisionada e tem sido amplamente utilizada em diversas áreas como, por exemplo, nos trabalhos de Morais et al. (2021), onde foi proposta uma técnica de detecção de anomalias na vibração de motores elétricos, baseando-se no algoritmo de classificação One-Class Support Vector Machine (OCSVM) e Silva, cujo objetivo é a criação de modelo de detecção de anomalias para termômetro *IoT* usado em refrigeradores hospitalares. Pode-se definir como objeto da detecção de anomalias, identificar eventos que estão em não conformidade com um padrão esperado ou com um conjunto de dados. Geralmente, são acontecimentos que estão relacionados com algum tipo de problema em um certo sistema, por exemplo: uma fraude bancária, falha em componentes mecânicos e elétricos, problemas médicos, entre outros.

A figura 3 ilustra um conjunto de dados que contém anomalia.

Figura 3 – Conjunto de dados com Anomalia.



Fonte: Géron (2019)

2.4 APRENDIZADO SEMI-SUPERVISIONADO

São algoritmos com capacidade de aprender com exemplos rotulados e não rotulados. Esse tipo de aprendizado é bastante útil pois em muitas tarefas de aprendizado, há uma grande quantidade de dados não rotulados e os dados rotulados são insuficientes, pois a geração de dados rotulados é frequentemente custosa (AMINI; GALLINARI, 2003; BASU; BANJEREE; MOONEY, 2004). O aprendizado semi-supervisionado pode ser utilizado em tarefas de classificação e *clustering.*

2.5 APRENDIZADO POR REFORÇO

O aprendizado por reforço é o treinamento de um modelo de aprendizado de máquina para tomar uma série de decisões. Os agentes aprendem a atingir objetivos em ambientes incertos e potencialmente complexos. No aprendizado por reforço, os sistemas de inteligência artificial enfrentam uma situação e os computadores encontram soluções para os problemas por meio de tentativa e erro. Para que as máquinas façam o que os programadores querem, a IA é recompensada ou punida pelas ações que realiza. Seu objetivo é maximizar o retorno total.

Mesmo o programador definindo as regras e recompensas, ele não dá ao modelo nenhuma sugestão de como alcançar seu objetivo. Cabendo ao modelo descobrir como executar.

O principal desafio do aprendizado por reforço está na preparação do ambiente de simulação, que depende muito da tarefa a ser executada. Escalar e ajustar a rede neural que controla o agente é outro desafio. Não há como se comunicar com a rede a não ser através do sistema de recompensas e penalidades. Isso pode levar a um esquecimento catastrófico, em que a aquisição de novos conhecimentos faz com que alguns dos antigos sejam apagados da rede. Ou seja, é necessário guardar o aprendizado na “memória” do agente.

Outro desafio é alcançar um ótimo local – ou seja, o agente executa a tarefa como está, mas não da maneira ideal ou necessária. Por fim, existem agentes que otimizarão o prêmio sem executar a tarefa para a qual foram projetados.

2.6 DETEÇÃO DE ANOMALIAS

Anomalia é definida como uma alteração dentro de um padrão de dados, ou seja, é um valor atípico ou um evento que esteja fora de uma tendência padrão.

O processo de detecção de anomalias é o nome dado à tarefa de identificar ocorrências incomuns em um conjunto de dados. A anomalia é definida como uma observação que difere tanto de outras observações.

Os algoritmos de detecção de anomalias podem ser utilizados em diversas áreas, como: detecção de fraudes em cartão de crédito e em seguros automotivos.

2.6.1 *One-class* SVM

Em seu trabalho, Schölkopf et al. (1999) definiram um método de adequação da metodologia SVM para a problemática de classificação de uma classe, que até então era limitado a regras de decisão lineares no espaço de entrada e não havia como lidar com *outliers*. No trabalho foi proposto um algoritmo que computa uma função binária, que deve capturar regiões no espaço de entrada onde reside a densidade de probabilidade, ou seja, uma função tal que a maioria dos dados resida na região onde a função tem valor diferente de zero**.**

O *One-Classe* SVM é um algoritmo adaptado do *Support Vector Machine* para o caso de classificação de uma classe. O OCSVM tem como objetivo maximizar a distância de um hiperplano, usando a origem como referência, até um conjunto de dados, esta etapa é estabelecida a partir do uso funções denominadas kernel, o núcleo de uma transformação linear.

De acordo com Pedregosa et al. (2011), o OCSVM se baseia em fronteiras lineares para a separação de diferentes conjuntos de dados. Ainda que também seja uma técnica de aprendizado baseada em processos estatísticos, as SVMs não fornecem estimativas de probabilidade diretamente, uma vez que estas são computadas usando validação cruzada.

2.6.2 *Isolation Forest*

Para Aggarwal (2017), *isolation forest* é uma combinação de um conjunto de árvores de isolamento. De acordo com o autor, os dados em uma árvore de isolamento são particionados recursivamente com cortes paralelos ao eixo em pontos de partição escolhidos de forma randômica em atributos também selecionados randomicamente, de tal modo que as instâncias estejam isoladas em nós *singleton*. Como esses pontos de dados estão localizados em locais esparsos, os galhos da árvore contendo valores discrepantes são notavelmente menos profundos nessas circunstâncias. Como resultado, a pontuação *outlier* é determinada pela distância entre a folha e a raiz. Os comprimentos de caminho dos pontos de dados nas árvores individuais da floresta de isolamento são calculados na etapa final de combinação.

Liu et al. (2008) divide o processo de detecção de anomalias usando *isolation forest* em duas etapas, sendo: treinamento, onde são construídas as árvores de isolamento, utilizando subamostras do conjunto de treinamento fornecido. A segunda etapa passa as instâncias de teste por meio de árvores de isolamento com finalidade de obter uma pontuação de anomalia para cada instância.

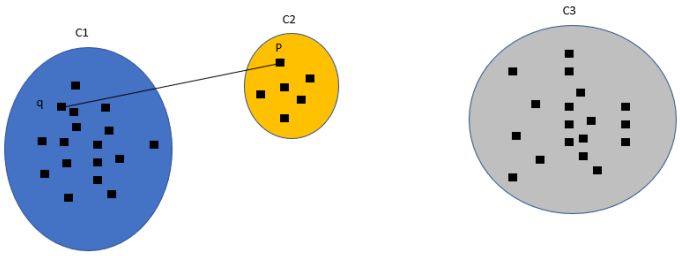
Ainda segundo Liu et al. (2008), o *isolation forest* é um algoritmo com baixa complexidade, tempo linear e um pequeno requisito de memória. É possível construir um modelo cujo desempenho seja bom com uma quantidade pequena de árvores e usando pequenas subamostras de tamanho fixo, independentemente do tamanho do conjunto de dados.

2.6.3 *Cluster-Based Local Outlier Factor*

Em seu trabalho He et al. (2003), apresentam uma nova definição para outlier, CBLOF: *Cluster-based local outlier factor*, uma medida para identificar a significância física de um outlier.

Para He et al. (2003), a pontuação de uma anomalia é igual à distância até o cluster de maior tamanho mais próximo multiplicado pelo tamanho do cluster ao qual o objeto pertence. A Figura 4 ilustra esse conceito.

Figura 4 – Ilustração da distância de cluster menor até o cluster maior mais próximo.



Fonte: Autoria Própria

O ponto P está no pequeno cluster C2 e, portanto, a pontuação seria igual à distância para C1, que é o grande cluster mais próximo multiplicado por 6, que é o tamanho de C2.

2.6.4 *Empirical-Cumulative-Distribution-Basedoutlier Detection*

Li et al. (2022) descreve que o *Empirical-Cumulative-Distribution-Based Outilier Detection* (ECOD) primeiramente estima a distribuição subjacente do conjunto de dados de entrada de forma não paramétrica, computando a distribuição cumulativa empírica por dimensões dos dados. Em seguida, o algoritmo usa essas distribuições empíricas para estimar as probabilidades de cauda por dimensão para cada ponto de dados. Finalmente, o ECOD calcula uma pontuação atípica de cada ponto de dados agregando as probabilidades estimadas entre as dimensões.

Li et al. (2022) destaca ainda em seu trabalho, por meio de experimentos, que o ECOD tem bons resultados de eficácia, eficiência, escalabilidade e interoperabilidade em relação a outros métodos de detecção de *outliers*. O autor também mostra que como desvantagem o ECOD não tem ajuste de hiper parâmetros e tem complexidade de tempo que escala linearmente no tamanho do conjunto de dados e no número de dimensões.

3 METODOLOGIA

Esta seção tem como objetivo descrever os modelos e dados utilizados no presente trabalho.

3.1 BASE DE DADOS

O conjunto de treinamento contém 60.000 exemplos no total, dos quais 59.000 pertencem à classe negativa e 1.000 à classe positiva, como mostra a figura 5. O conjunto de teste contém 16.000 exemplos, sendo 15.625 negativos e 375 positivos. Todos os dados possuem 171 atributos por registro. Os nomes dos atributos dos dados foram anonimizados por motivos de propriedade. A primeira coluna denominada de *class* é onde está inserido a informação de positivo e negativo, ou seja, se é relacionado ao sistema APS ou não. Os valores ausentes são indicados por "na".

Figura 5 – Dados de treinamento.

Fonte: Autoria Própria

3.2 SOFTWARE E COMPUTADOR

Foi utilizado a linguagem *Python* para desenvolvimento deste trabalho, é uma linguagem de alto nível e bastante utilizada em aprendizado de máquinas.

Python também é bastante utilizado no desenvolvimento *web*, ou seja, para a construção de sites, aplicativos, softwares, bancos de dados e para o desenvolvimento de jogos.

A interface utilizada para o desenvolvimento dos algoritmos foi a *Google Collaboratory*, que é um serviço de armazenamento em nuvem de notebooks voltados à criação e execução de códigos em Python. Seu uso se dá diretamente em um navegador, sem ter a necessidade de instalação de software em uma máquina. Neste ambiente é possível desenvolver e rodar códigos, compartilhá-los, modificá-los e mantê-los salvos de maneira totalmente online. O poder computacional utilizado para executar os programas é fornecido pela nuvem de computadores da Google.

3.3 PRÉ-PROCESSAMENTO DOS DADOS

Para Faceli et al. (2011) as técnicas de pré-processamento de dados são frequentemente utilizadas para melhorar a qualidade dos dados por meio da eliminação ou minimização de problemas como: valores duplicados, ruídos, imperfeições, valores incorretos e valores inconsistentes. Essa melhora pode facilitar o uso de técnicas de aprendizado de máquina, levar à construção de modelos mais fiéis à distribuição real dos dados, reduzindo sua complexidade computacional, tornar mais fáceis e rápidos o ajuste de parâmetros do modelo e seu posterior uso.

Dessa forma o conjunto de dados obtido no *site kaggle* passou pelas seguintes etapas de processando de dados:

1. Substituição dos dados ausentes codificados como 'na' por NaN;
2. Mapeamento da coluna de valor binário 'class' para {-1,1} para torná-la adequada para os modelos de rede profunda que usam tanh() como a função de ativação;

3. Atribuição de valores ausentes usando a estimativa de K-vizinhos mais próximos do pacote *pythonfancyimpute*. Foi feito separadamente para dados de treinamento e teste;

4. Normalização das colunas de recursos para média zero, variação de unidade nos dados de treinamento. Aplicação de coeficientes de normalização calculados para dados de treinamento aos dados de teste;

5. Padronizar as colunas de recursos para [-1, 1] para atenuar o efeito de valores discrepantes;

6. Diminuir o número de casas de precisão dos dados para que cada valor caiba em um único byte.

3.4 TREINAMENTO E TESTES

Os treinamentos dos estimadores foram realizados em sua configuração *default*, ou seja, sem alteração dos hiperparâmetros dos algoritmos. Para a construção dos modelos foi utilizado a biblioteca PyOD do *Python*.

É importante ressaltar que o treinamento dos estimadores foi realizado somente com os dados da classe majoritária, ou seja, os *inliers*. Os estimadores retornam como *inlier*s valor de 1 e para os *outliers* o valor de -1.

Os resultados serão exibidos em gráficos de matriz de confusão (tabela 1), que é uma tabela que mostra as frequências de classificação para cada classe do modelo.

Segundo Guerón (2019), a matriz de confusão é uma forma de avaliar melhor o desempenho de um classificador. A ideia geral é contar o número de vezes que as instâncias da classe A são classificadas como classe B. Para calcular a matriz de confusão é necessário primeiramente ter um conjunto de previsões para que possa ser comparado com os alvos reais. Cada linha na matriz de confusão representa uma classe real, enquanto cada coluna representa a classe prevista.

Tabela 1 - Matriz de confusão

		CLASSE PREDITIVA	
		+	-
CLASSE VERDADEIRA	+	VP	FN
	-	FP	VN

Fonte: Faceli et al. (2011, p. 164)

Onde:

- Verdadeiro positivo (VP) ocorre quando valor real e o valor previsto são iguais e positivos;
- Falso positivo (FP) ocorre quando a classe verdadeira é negativa e o algoritmo classificou como positiva
- Verdadeiro Negativo (VN) os valores reais e previstos são os mesmos, sendo que o valor previsto do modelo é negativo, juntamente com um valor negativo real;
- Falso negativo (FN): ocorre quando a classe real é positiva e o algoritmo classifica como negativo.

Para Géron (2019), a matriz de confusão fornece muitas informações, mas às vezes pode ser necessária uma métrica mais concisa. Uma métrica interessante a ser observada é a precisão das previsões positivas; que é chamada de precisão do classificador. A equação 1 mostra como é calculada a precisão:

$$Precisão = \frac{VP}{VF+VP} \quad (1)$$

Em seu trabalho, Géron (2019) explica que uma maneira trivial de ter uma acurácia perfeita é fazer uma única previsão positiva e garantir que ela seja correta (acurácia de 100%). Entretanto, não seria muito útil uma vez que o classificador ignoraria todas, exceto uma instância positiva. Portanto, a precisão é utilizada em conjunto com outra métrica chamada *revocation*, também conhecida como sensibilidade ou taxa de verdadeiros positivos (*True Positive Rate* - TPR, do inglês): esta é a taxa de instâncias

positivas que são corretamente detectadas pelo classificador. A equação 2 mostra como é calculada a *revocation*:

$$Revocation = \frac{VP}{VP+FN} \quad (2)$$

4 RESULTADOS E DISCUSSÕES

Os testes e consequentes análises, necessários para responder o problema de pesquisa apresentado na introdução, são apresentados neste capítulo. Tem-se como ponto central o teste de diferentes modelos de detecção de anomalias baseados em estratégias de *one-class*.

Este capítulo está divido da seguinte maneira:

- na seção 4.1 é apresentada a metodologia experimental, onde serão descritos os dados de treinamento, parâmetros e procedimentos de avaliação para os algoritmos escolhidos;
- nas seções 4.2 a 4.5 são apresentados e discutidos os resultados provenientes da aplicação da metodologia delineada no capítulo 3 na base de dados de APS de caminhões da marca Scania;
- na seção 4.6 é apresentada uma comparação dos resultados obtidos.

4.1 EXPERIMENTOS

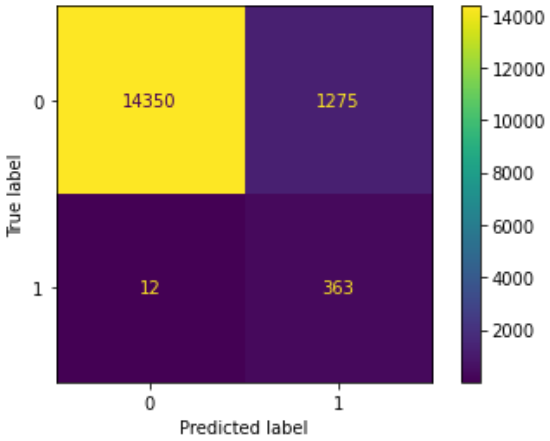
O conjunto de treinamento contém 60.000 exemplos no total, dos quais 59.000 pertencem à classe negativa e 1.000 à classe positiva, como mostra a figura 5. O conjunto de teste contém 16.000 exemplos, sendo 15.625 negativos e 375 positivos. Todos os dados possuem 171 atributos por registro. Os algoritmos foram usados com a taxa de contaminação de 10% e os demais parâmetros em sua configuração padrão. O custo de manutenção é estimado conforme a equação 3, onde $custo1$ é igual a \$500,00 e $custo2$ é igual a \$10,00:

$$CustoManutenção = FN * custo1 + FP * custo2 \quad (3)$$

4.2 ONE CLASS SVM

Primeiramente, será analisada a matriz de confusão gerada pelo modelo *One-class SVM*, conforme mostrado na tabela 2.

Tabela 2 – Matriz de confusão do algoritmo *One-Class SVM.*



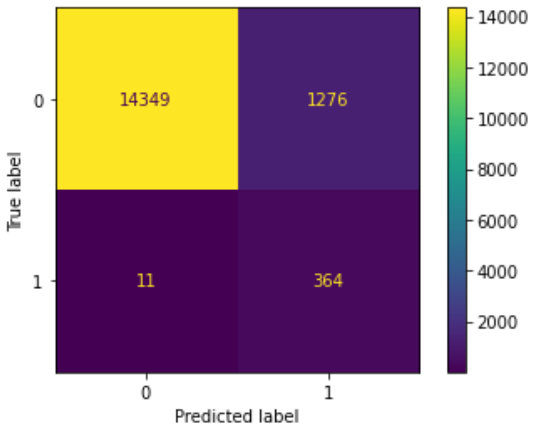
Fonte: Autoria Própria

Analisando a matriz confusão da tabela 2, é observado que para os 16 mil dados destinados para testes sendo 15.325 para *inliers* e 375 para classe de *outliers*, o algoritmo acertou um total de 14.713 dados (*inliers* + *outliers)*, apresentando 91,8% de acerto da classe majoritária (*inliers*) e 96,8% de acerto na classe minoritária (*outliers*). A seguir, é utilizado a equação 3 para calcular o custo de manutenção com a utilização do *One-Class SVM*, alcançando um custo total de $18.750,00. O tempo de processamento do treinamento do modelo foi de 1.251,85 s.

4.3 ISOLATION FOREST

Ao utilizar o algoritmo do *isolation forest* obteve-se a seguinte matriz de confusão, mostrada na tabela 3.

Tabela 3 – Matriz de confusão do algoritmo *Isolation Forest.*



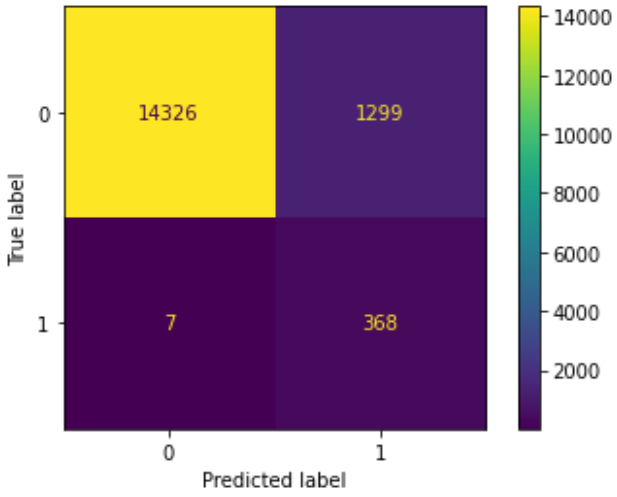
Fonte: Autoria Própria

Os resultados mostrados na matriz de confusão acima são de acerto de 14.713 (*outliers + inliers)* dados de um total de 16 mil. Sendo que para classe majoritária o algoritmo teve uma taxa de acerto de 91,8% (*inliers)* e para classe minoritária (*outliers)* 97%. Utilizando a equação 3, o modelo obteve um custo total de manutenção de $18.260,00. Nota-se que o resultado obtido pelo algoritmo é bem próximo do obtido pelo algoritmo *One-class SVM*, com uma pequena melhora no FN. O tempo de processamento do treinamento do modelo foi de 16,87 s.

4.4 CLUSTER-BASED LOCAL OUTLIER FACTOR

Inicialmente será exibido o resultado do algoritmo *CBLOF,* conforme tabela 4.

Tabela 4 – Matriz de confusão do algoritmo *CBLOF.*



Fonte: Autoria Própria

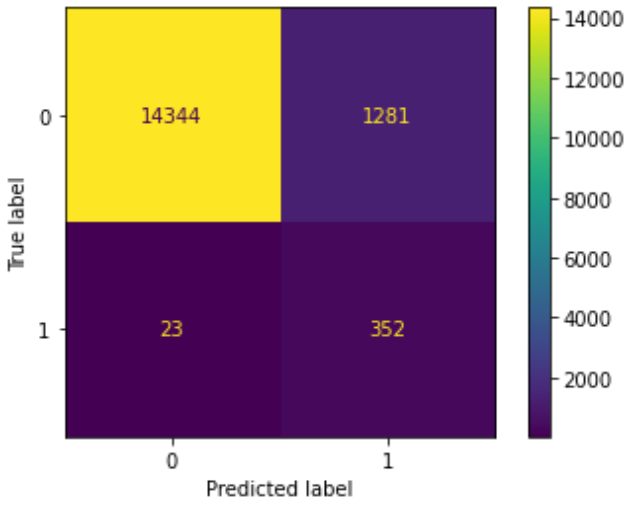
Analisando a matriz confusão da tabela acima, é observado que o algoritmo acertou 14.694 (*outliers + inliers)* dados de um total de 16.000. Como resultado, o modelo do *CBLOF* apresentou uma taxa de 91,7% de acertos para a classe majoritária (*inliers) e* 98,1% de acerto na classe minoritária (*outliers*). Através da equação 3 é calculado o custo de manutenção obtendo-se um total de $16.490,00. O tempo de processamento do treinamento do modelo foi de 17,53 s.

Este algoritmo apresentou o melhor custo total e isso foi devido ao fato deste algoritmo detectar bem os *outliers*, que possuem um custo relativamente alto.

4.5 EMPIRICAL-CUMULATIVE-DISTRIBUTION-BASED OUTLIER DETECTION

Por fim, será utilizado o algoritmo do *ECOD* cujos resultados estão expostos na matriz de confusão da tabela 4.

Tabela 5 – Matriz de confusão do algoritmo *ECOD.*



Fonte: Autoria Própria

A matriz de confusão da tabela 5 exibe os resultados do algoritmo *ECOD*, cujo acerto foi de 14.696 dados (*inliers* + *outliers).* Sendo que para classe majoritária o algoritmo teve uma taxa de acerto de 91,8% (*inliers)* e para classe minoritária (*outliers)* 93,8%. Novamente, utilizando-se a equação 3 foi calculado o custo de manutenção obtendo-se um custo total de $24.310,00. O tempo de processamento do treinamento do modelo foi de 4,71 s.

Apesar desse algoritmo ser mais recente, ele apresentou o pior resultado de custo total. É provável que isso tenha ocorrido pela falta de ajuste dos hiperparâmetros do modelo.

4.6 COMPARAÇÕES DE RESULTADOS

Ao analisar os resultados obtidos pelos modelos *One-Class SVM, Isolation Forest, CBLOF* e *ECOD,* conforme exibidos na tabela (6) pode-se aferir que o método *CBLOF* obteve o menor custo de manutenção, sendo que a precisão para *outliers* foi de 98,1% e para *inliers* 91,7%.

Tabela 6 – Comparação dos resultados obtidos por cada algoritmo do tipo *one-class*, com destaque para o melhor resultado obtido.

METÓDO	**Precisão *Outliers***	**Precisão *Inliers***	**Custo De Manutenção**	**Tempo de Processamento**
One-Class SVM	96,8%	91,8%	$18.750	16,87 s
Isolation Forest	97,0%	91,8%	$18.260	1.251,8 s
CBLOF	98,1%	91,7%	**$16.490**	17,53 s
ECOD	93,8%	91,8%	$24.310	4,71 s

Fonte: Autoria Própria

No aspecto precisão para detecção de *outliers*, o *CBLOF* apresentou o melhor de desempenho (98,1%), sendo que o *ECOD* apresentou a menor taxa de acerto (93,8%). Já para precisão na detecção de *inliers*, *todos* os algoritmos tiveram um valor de precisão equivalente (91,8%). Sendo que o *CBLOF* demonstrou um percentual de 91,7% para esse quesito.

Ao analisar o custo de manutenção, o CBLOF ($16.490) obteve o melhor resultado tendo uma diferença percentual entre o pior resultado ($24.310) de 47,4%. É importante destacar, novamente, que todos os modelos foram usados em sua configuração *default.* No quesito tempo de processamento para treinamento o modelo que obteve melhor performance foi o ECOD com 4,71 segundos.

Se comparado ao trabalho apresentado por Leão (2019), o resultado deste trabalho apresentou uma piora de 7,8% (de $15.290,00 para $16.490,00). Novamente, é importante destacar que, em contraste com o trabalho de Leão (2019), o trabalho aqui apresentado não explorou os hiperparâmetros dos modelos utilizados.

5 CONCLUSÃO

Como destacado, na área de manutenção de equipamentos e predição de falhas, o uso de métodos de aprendizado de máquina tem se potencializado, permitindo a criação de modelos que descrevem o funcionamento dos equipamentos em condições específicas. Assim, o presente trabalho apresentou quatro técnicas não supervisionadas para detecção de *outliers* em uma base de dados de falhas do sistema APS em caminhões da marca Scania. Dessa forma, os resultados obtidos através dos estimadores *One-Class SVM, Isolation Forest, CBLOF* e *ECOD* para o caso estudado, se mostraram satisfatórios, sendo o melhor desempenho alcançado com o menor valor de custo de $16.490 e tempo de treinamento de 17,53 segundos pelo estimador CBLOF em configuração *default*. Entretanto o ECOD demonstrou ser bastante interessante, visto que o tempo de processamento foi de 4,71 segundos e é possível explorar mais ainda seus parâmetros de configuração a fim de reduzir o custo de manutenção.

As técnicas de aprendizado de máquina não supervisionado que foram apresentadas neste trabalho podem ser configuradas com outros valores de parâmetros e, assim, podendo apresentar menores valores de custo. Destaca-se que os algoritmos utilizados no trabalho podem ser aplicados a várias outras situações, como em detecção de fraudes de cartão de crédito, mau funcionamento de equipamentos, defeitos ou fraude.

Como trabalhos futuros, destaca-se a utilização de algoritmos automáticos de *tuning* dos hiperparâmetros dos modelos como: G*rid search*, *Random search*, Otimização Bayesianas, entre outros. Além disso, algoritmos baseados em aprendizado profundo do tipo *one-class* podem ser utilizados, como o *Deep Semi-Supervised Anomaly Detection* e o *Deep Weakly-supervised Anomaly Detection*.

REFERÊNCIAS

AGGARWAL, Charu. **Outlier analysis**. 2. ed. New York: Springer, 2017.

AIR pressure system failures in Scania trucks. Disponível em: https://www.kaggle.com/datasets/uciml/aps-failure-at-scania-trucks-data-set. Acesso em: 14 jul. 2022.

ALTMAN, N. S. **An introduction to kernel and nearest-neighbor nonparametric regression**. [S.l.]: American Statistician, 1992.

AMINI, M.R.; GALLINARI, P. Semi-supervised learning with explicit misclassification modeling. In: INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE. 18., 2003, Acapulco, Mexico. **Proceedings…** [S.l.; s.n], 2003. P. 555-560.

AN awesome tutorial to learn outlier detection in Python using PyOD Library. Disponível em: https://www.analyticsvidhya.com/blog/2019/02/outlier-detection-python-pyod/. Acesso em: 14 jul. 2022.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 6023**: informação e documentação – referências – elaboração. Rio de Janeiro, 2002. 24 p.

BACKER, E. **Computer assisted reasoning in cluster analysis**. New York: Prentice Hall, 1995.

BEN-HUR, A. et al. A support vector clustering method. In: INTERNATIONAL CONFERENCE ON PATTERN RECOGNITION (ICPR'00). 3-7 set. 2000, Barcelona, Spain. **Proceedings…** [S.l.]: IEEE, 2000. V. 2, p. 724–727. Acesso em: 23 abr. 2022.

BIAU, G. Analysis of a random forests model. **The Journal of Machine Learning Research**, 2012. 98888:1063–1095.

BREUNIG, Markus M. et al. LOF: identifying density-based local outliers. In: ACM SIGMOD INTERNATIONAL CONFERENCE ON MANAGEMENT OF DATA. 2000. **Proceedings…** [S.l.; s.n.], 2000. p. 93–104.

CERQUEIRA, Vítor et al. **Combining boosting trees with metafeature engineering for predictive maintenance**. 2016. Disponível em: https://www.researchgate.net/publication/313067390_Combining_Boosted_Trees_. Acesso em: 17 set. 2022.

CHAWLA, Nitesh V. et al. SMOTE: synthetic minority oversampling technique. **Journal of artificial intelligence**, n. 16, p. 321-357, 2002.

COSTA, C. F.; nascimento, m. a. ida 2016 industrial challenge: using Machine Learning for Predicting Failures. In: Boström H. (ed.) et al. **Advances in intelligent**

data analysis XV. IDA 2016. Lecture Notes in Computer Science, v. 9897. [S.l.]: Springer, Cham.

CRAMER, J. S. **The origins of logistic regression.** [S.l.]: Tinbergen Institute. v. 119. p. 167–178, 2002.

FACELI, Katti et al. F. **Inteligência Artificial**: uma abordagem de aprendizagem de máquina. Rio de Janeiro: LTC, 2011.

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

GONDEK, Christopher; HAFNER, Daniel; SAMPSON, Oliver R. Prediction of failures in the air pressure system of scania trucks using a random forest and feature engineering. 2016. Disponível em: https://www.researchgate.net/publication/309195602_Prediction_of_Failures_in_the. Acesso em: 24 abr. 2022.

GOUVEIA, Cristiano Gonçalves Nascimento. **Técnicas de aprendizado de máquina aplicadas à predição de vazamentos em ramais de redes de distribuição de água**. 2022. Disponível em: https://repositorio.unb.br/handle/10482/43766.

HAWKINS, D. **Identification of Outliers**. London and New York: Chapman and Hall, 1980.

HE, Zengyou; XU, Xiaofei; DENG, Shengchun. Discovering cluster-based local outliers.**Pattern Recognition Letters**, v. 24, n. 9-10, p. 1641–1650, 2003.

HO, Tin Kam. Random Decision Forests. In: INTERNATIONAL CONFERENCE ON DOCUMENT ANALYSIS AND RECOGNITION. 3., 1995, Montreal, QC. **Proceedings…** [S.l.; s.n.], 1995. p. 278–282.

INSTITUTO FEDERAL DO ESPÍRITO SANTO. **Normas para apresentação de trabalhos acadêmicos e científicos:** documento impresso e/ou digital. 7. ed. rev. e ampl. Vitória: Ifes, 2014.

LEÃO, Mariana Spadetto. **Detecção de anomalias no sistema APS de caminhões de carga utilizando one-classSVM**. 2019. 48 f. Monografia (Graduação em Engenharia de Controle e Automação) - Instituto Federal do Espírito Santo, Serra, 2019.

LI, Zhenget al. Ecod: unsupervised outlier detection using empirical cumulative distribution functions. **IEEE Transactions on Knowledge and Data Engineering**, 2022.

LIU, F. T. et al. Isolation forest. In: IEEE INTERNATIONAL CONFERENCE ON DATA MINING. 8., 2008. **Proceedings…** [S.l.]: IEEE, 2008. p. 413–422.

MORAIS, Lucas Gabriel Cosmo, SILVA NETTO, Ademar Virgolino da. Detecção de anomalias na vibração de motores elétricos baseada em OCSV. In: SIMPÓSIO

BRASILEIRO DE AUTOMAÇÃO INTELIGENTE. 15., 17-20 out. 2021. **Anais...** [S.l.]: SBAI, 2021.

O que é aprendizagem Por Reforço? Capítulo 62. Disponível em: https://www.deeplearningbook.com.br/o-que-e-aprendizagem-por-reforco/. Acesso em: 11 set. 2022.

RAWAT, Sushil. **Predict component failure related with air pressure system at scania trucks using various machine learning methods.** Disponível em: https://www.researchgate.net/publication/349589259. Acesso em: 22 abr. 2022.

SCHÖLKOPF, Bernhard et al. Estimating the support of a high-dimensional distribution. **Neural computation**, v. 13, n. 7), p. 443-1471, 2001.

SILVA, D. da; NUNES, I.; SILVA, S. da; ALVES, E. Criação de modelo de detecção de anomalias para termômetro IoT usado em refrigeradores hospitalares. **Revista de Engenharia e Pesquisa Aplicada**, v. 6, n. 5, p. 120-128, 20 nov. 2021.

SILVA, Thaynara Leal da; CAVALIERI, Daniel Cruz; PEREIRA, Flávio Garcia. Estimativa da espessura de tiras de aço em laminador de acabamento utilizando técnicas de aprendizado de máquina. In: SEMINÁRIO DE AUTOMAÇÃO E TI. 22., 2018, São Paulo. **Anais...** [S.l.; s.n.], 2018.

SOUZA, Rafael Fernando Silva e. **Detecção e classificação de falhas em rolamentos de motores elétricos baseado em árvores de decisição**. 2022. Disponível em: https://repositorio.ufrn.br/handle/123456789/46252.

TORGO, Luis. Resource-bounded fraud detection. In: NEVES, J.; SANTOS, M. F.; MACHADO, J. M. (Ed.) EPIA 2007. **LNCS**, v. 4874, p. 449–460. Heidelberg: Springer, 2007.

WelcometoPyODdocumentation. Disponível em: https://pyod.readthedocs.io/en/latest/. Acesso em: 14 jul. 2022.