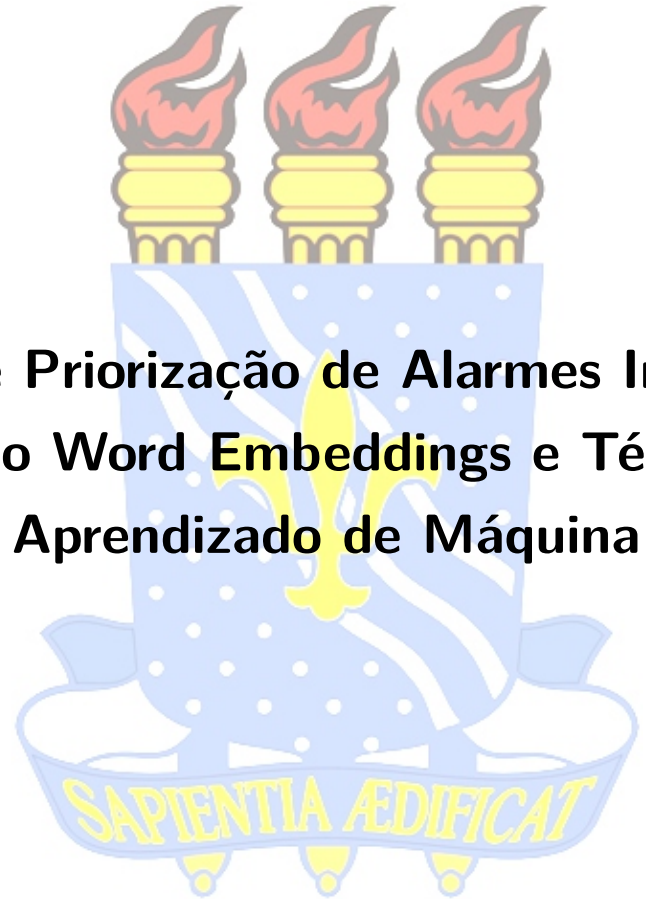
UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE ENERGIAS ALTERNATIVAS E RENOVÁVEIS
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

Isaac Emmanuel Azevedo de Medeiros



Análise e Priorização de Alarmes Industriais Utilizando Word Embeddings e Técnicas de Aprendizado de Máquina

João Pessoa
2024

Isaac Emmanuel Azevedo de Medeiros

Análise e Priorização de Alarmes Industriais Utilizando Word Embeddings e Técnicas de Aprendizado de Máquina

Dissertação apresentada ao Programa de Pós Graduação em Engenharia Elétrica da Universidade Federal da Paraíba como exigência para a obtenção do título de Mestre em Engenharia Elétrica.

Universidade Federal da Paraíba

Centro de Energias Alternativas e Renováveis

Programa de Pós Graduação em Engenharia Elétrica

Orientador: Prof. Dr. Juan Moises Mauricio Villanueva

João Pessoa

2024

Catalogação na publicação
Seção de Catalogação e Classificação

M488a Medeiros, Isaac Emmanuel Azevedo de.
Análise e priorização de alarmes industriais utilizando word embeddings e técnicas de aprendizado de máquina / Isaac Emmanuel Azevedo de Medeiros. - João Pessoa, 2024.
77 f. : il.

Orientação: Juan Moises Mauricio Villanueva.
Dissertação (Mestrado) - UFPB/CEAR.

1. Alarmes industriais. 2. Incorporação de palavras. 3. Gerenciamento de alarmes. I. Villanueva, Juan Moises Mauricio. II. Título.

UFPB/BC CDU 621.3(043)

Elaborado por CHRISTIANE CASTRO LIMA DA SILVA - CRB-15/865

UNIVERSIDADE FEDERAL DA PARAÍBA - UFPB
CENTRO DE ENERGIAS ALTERNATIVAS E RENOVÁVEIS - CEAR
DEPARTAMENTO DE ENGENHARIA ELÉTRICA - PPGEE

A Comissão Examinadora, abaixo assinada, aprova a Dissertação

Análise e Priorização de Alarmes Industriais Utilizando Word Embeddings e Técnicas de Aprendizado de Máquina

Elaborada por

Isaac Emmanuel Azevedo de Medeiros

como requisito parcial para obtenção do grau de
Mestre em Engenharia Elétrica.

Comissão Examinadora

gov.br Documento assinado digitalmente
JUAN MOISES MAURICIO VILLANUEVA
Data: 12/09/2024 08:17:56-0300
Verifique em https://validar.iti.gov.br

Prof. Dr. Juan Moises Mauricio Villanueva
Orientador - UFPB

gov.br Documento assinado digitalmente
FABIANO SALVADORI
Data: 16/09/2024 21:35:58-0300
Verifique em https://validar.iti.gov.br

Prof. Dr. Fabiano Salvadori
Examinador Interno - UFPB

gov.br Documento assinado digitalmente
IVANOVITCH MEDEIROS DANTAS DA SILVA
Data: 17/09/2024 07:45:58-0300
Verifique em https://validar.iti.gov.br

Prof. Dr. Ivanovitch Medeiros Dantas da Silva
Examinador Externo - UFRN

João Pessoa/PB , 31 de agosto de 2024

AGRADECIMENTOS

Agradeço primeiramente a Deus por tudo que Ele têm feito, e por ter me ajudado a vencer todos os desafios para chegar até aqui.

À minha querida mãe, Carla, pelo apoio indescritível e amor incondicional durante toda a minha existência. Jamais teria chegado até aqui sem você.

À minha avó Conceição, por todo o cuidado e dedicação.

Às minhas tias Karina e Patrícia, que sempre me deram todo amor e carinho.

Aos demais membros da família que se fizeram presentes na minha trajetória.

Ao meu orientador, Prof. Juan, pelo longo tempo de parceria desde o período de graduação, e por todas as contribuições e ensinamentos constantes.

À Júlia, por todo o amor. Obrigado pelo cuidado, carinho, motivações constantes e ter feito meus dias sempre mais leves e felizes.

À EPASA - Centrais Elétricas da Paraíba, por todo o suporte e fornecimento de insumos para o desenvolvimento deste trabalho.

Aos meus amigos Carlos Araújo e Rodrigo Almeida, por terem contribuído significativamente para o meu crescimento pessoal e profissional. Obrigado pela lealdade e todo o conhecimento técnico e ensinamentos transmitidos.

Aos amigos Clayson, Rafael e Victor, pelos incentivos e discussões proveitosas.

Por fim, a todos os que contribuíram de maneira direta ou indireta para esta conquista.

RESUMO

A análise do conteúdo de alarmes industriais é de suma importância para a detecção e prevenção de falhas em processos operacionais. Os alarmes funcionam como um sistema de alerta, sinalizando à equipe de operação sobre condições anormais e potenciais falhas em tempo real. Entretanto, a geração excessiva de registros por parte dos sistemas pode dificultar a identificação e a resposta eficaz diante de situações críticas. Por isso, torna-se imprescindível o desenvolvimento de uma gestão eficiente dos alarmes, a fim de priorizá-los e agrupá-los de maneira inteligente. Além disso, ao realizar uma análise minuciosa dos dados de alarmes industriais, é possível obter um entendimento mais aprofundado das condições operacionais, reconhecer padrões recorrentes e identificar tendências, além de agir de forma proativa para evitar falhas. Neste estudo, foram coletados e utilizados dados de eventos e alarmes extraídos do sistema SCADA (*Supervisory Control and Data Acquisition*) de uma usina termoelétrica situada no estado da Paraíba. Realizou-se uma análise exploratória, a fim de entender os impactos operacionais causados pela volumetria dos alarmes e, para estes e seus respectivos *clusters*, buscar padrões envolvendo sequências temporais, que podem sugerir causalidade, bem como auxiliar na determinação de causas raiz para determinados registros. Técnicas de processamento de linguagem natural (PLN) foram utilizadas no pré-processamento dos textos dos alarmes para generalizar informações, eliminando identificadores de equipamentos e elementos com baixa relevância semântica. Foi utilizado o modelo de linguagem BERT (*Bidirectional Encoder Representations from Transformers*) para a representação numérica do texto, e aplicadas técnicas de clusterização e classificação para o agrupamento eficiente dos alarmes. Por meio da clusterização das mensagens dos alarmes, utilizando o algoritmo *K-means*, com os os *clusters* obtidos, aplicou-se o algoritmo de classificação *Support Vector Classifier* (SVM) com *kernel* linear, alcançando uma acurácia superior a 99% no conjunto de dados de teste. Foi possível, portanto, rotular uma nova amostra com eficiência considerável. A utilização do BERT para transformar as mensagens dos alarmes em *embeddings* bem como o pré-processamento de texto contribuíram diretamente para os resultados obtidos. A abordagem realizada nesse trabalho não apenas permite melhorar a gestão dos alarmes, como também contribui para um ambiente operacional mais seguro e eficiente, o que é fundamental para a sustentabilidade e a produtividade da indústria.

Palavras-chave: Alarmes Industriais; Gerenciamento de Alarmes; Clusterização; Processamento de Linguagem Natural; Classificação; Incorporação de Palavras.

ABSTRACT

The analysis of industrial alarm content is of utmost importance for the detection and prevention of failures in operational processes. Alarms function as an alert system, signaling the operations team about abnormal conditions and potential failures in real-time. However, the excessive generation of records by these systems can hinder the identification and effective response to critical situations. Therefore, it is essential to develop efficient alarm management, aiming to prioritize and intelligently group alarms. Additionally, by conducting a thorough analysis of industrial alarm data, it is possible to gain a deeper understanding of operational conditions, recognize recurring patterns, identify trends, and take proactive measures to prevent failures. In this study, event and alarm data were collected and used from the SCADA (Supervisory Control and Data Acquisition) system of a thermoelectric plant located in the state of Paraíba. An exploratory analysis was conducted to understand the operational impacts caused by the volume of alarms, and for these alarms and their respective clusters, patterns involving temporal sequences were sought, which may suggest causality and assist in determining root causes for specific records. Natural language processing (NLP) techniques were used in the preprocessing of alarm texts to generalize information, eliminating equipment identifiers and elements with low semantic relevance. The BERT (Bidirectional Encoder Representations from Transformers) language model was used for the numerical representation of the text, and clustering and classification techniques were applied for the efficient grouping of alarms. By clustering alarm messages using the K-means algorithm, and with the obtained clusters, the Support Vector Classifier (SVM) algorithm with a linear kernel was applied, achieving an accuracy greater than 99% on the test dataset. It was thus possible to label a new sample with considerable efficiency. The use of BERT to transform alarm messages into embeddings, as well as the text preprocessing, directly contributed to the results obtained. The approach taken in this work not only improves alarm management but also contributes to a safer and more efficient operational environment, which is essential for the sustainability and productivity of the industry.

Keywords: Industrial Alarms; Alarm Management; Clustering; Classification;Natural Language Processing; Word embedding.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

UGD	*Unidade Geradora Diesel*
GCP	*Genset Control Panel*
SVM	*Support Vector Machine*
SVC	*Support Vector Classification*
NLP	*Natural Language Processing*
SCADA	*Supervisory Control and Data Acquisition*
BERT	*Bidirectional Encoder Representations from Transformers*
BLSTM	*Bidirectional Long Short-Term Memory*
t-SNE	*t-distributed Stochastic Neighbor Embedding*
CLP	*Controlador Lógico Programável*

SUMÁRIO

1 INTRODUÇÃO

1.1 PERTINÊNCIA E MOTIVAÇÃO DO TRABALHO

A indústria carrega o estigma de ser um ambiente rico em dados e escasso no desenvolvimento de conhecimento, em parte porque o setor é inerentemente conservador sobre seus processos e métodos. Estudos indicam que, embora a indústria de manufatura gere mais dados do que qualquer outro setor da economia, grande parte destes dados não é explorada pelas empresas (HARTMANN; KING; NARAYANAN, 2015; AARDT, 2015). Um destes estudos relata o exemplo de uma indústria de petróleo e gás que descarta 99% de seus dados antes que sejam utilizados para tomada de decisões. Isso foi substancialmente enfatizado pelo advento da Manufatura Inteligente e sua integração, além das organizações orientadas ao desempenho, com aplicação intensiva e generalizada de tecnologias baseadas em informações de rede ao longo da cadeia de manufatura e suprimentos (DAVIS et al., 2012).

A grande competitividade atual nos negócios têm obrigado as indústrias a transformarem dados em informações úteis e oportunas. Avanços recentes nas áreas de Ciência de Dados e Big Data têm auxiliado as empresas nesse esforço de transformar a cadeia de valor da manufatura, permitindo a aquisição de informações valiosas a partir de dados brutos provenientes de plantas industriais, apoiando operadores, analistas e gerentes na tomada de decisão, planejamento de ações e melhoria contínua das operações da planta (BEZERRA et al., 2019).

Os alarmes industriais desempenham uma função de grande importância nas usinas termoelétricas, pois ajudam a garantir a operação segura e eficiente das instalações. Tais alarmes são projetados para detectar e alertar os operadores sobre as condições anormais, problemas no funcionamento dos equipamentos ou processos da usina. A importância dos alarmes industriais em usinas termoelétricas pode ser compreendida sob várias perspectivas.

Em primeiro lugar, os alarmes industriais auxiliam na manutenção de segurança da planta. Usinas termoelétricas possuem processos complexos e potencialmente perigosos, como a queima de combustíveis fósseis e geração de altas temperaturas e pressões. Alarmes industriais podem detectar condições anormais, como vazamentos, superaquecimento ou falhas de equipamentos, alertando prontamente os operadores para adotar medidas apropriadas para evitar acidentes ou mitigar riscos potenciais (YANG; DZIEGIELEWSKI, 2007).

Em segundo lugar, os alarmes industriais são essenciais para a operação eficiente

das usinas termoelétricas. Através destes, é possível detectar desvios das condições normais de operação, como variações de temperatura, pressão, por exemplo, que podem indicar ineficiências ou falhas nos equipamentos (COSTA et al., 2019).

Uma técnica comumente utilizada para análise de dados de alarmes é a análise de similaridade. Este método consiste em identificar excessos de alarmes semelhantes, com base nos padrões de ocorrência dos registros. Ao agrupar sequências de alarmes semelhantes, torna-se mais fácil entender as causas subjacentes e tomar medidas apropriadas para prevenir ou mitigar incidentes futuros (AHMED et al., 2013).

Os sistemas de alarmes industriais enfrentam desafios devido ao excesso de registros, necessitando de melhores ferramentas de gerenciamento e apresentação. Em (CAI et al., 2019a), técnicas de agrupamento foram propostas para agrupar alarmes correlacionados, facilitando a remoção de alarmes redundantes e a identificação da causa raiz.

O agrupamento de texto é uma técnica fundamental para categorizar e analisar grandes volumes de dados textuais não estruturado (MEHTA; BAWA; SINGH, 2021). A eficácia do agrupamento depende de dois fatores principais: modelos de representação de texto e algoritmos de agrupamento (JAN, 2022). As técnicas de representação de texto transformam dados não estruturados em formatos vetoriais estruturados, capturando informações semânticas (JAN, 2022). Avanços recentes, como técnicas baseadas em *word embeddings*, mostraram melhorias significativas no desempenho do agrupamento para grandes conjuntos de dados de texto (MEHTA; BAWA; SINGH, 2021). Algoritmos de clusterização, como o K-means, são amplamente usado para essa tarefa, oferecendo uma abordagem eficaz para agrupar textos semelhantes (RAMKUMAR; NETHRAVATHY, 2019).

Esse trabalho tem como objetivo principal o desenvolvimento de uma metodologia para a clusterização e classificação de alarmes industriais, por meio de técnicas de aprendizado de máquina, bem como explorar análises de ocorrência de padrões sequenciais de alarmes, redundância de registros em curtos intervalos de tempo. A clusterização permitiu agrupar os alarmes apenas pelo conteúdo do texto, utilizando métodos de Processamento de Linguagem Natural (PLN), sem a necessidade de classificação ou rótulo prévio nos dados. Dessa maneira, torna-se possível priorizar os registros, filtrando apenas os que pertencem a determinados rótulos e subsistemas, ou ordenando por grau de criticidade de cada grupo. Além disso, permite-se identificar os subsistemas que possuem falhas mais frequentes ou severas, que podem levar a paradas dos motores ou problemas de operação, bem como os que necessitam de mais intervenções de manutenção.

1.1.1 OBJETIVO GERAL

O objetivo deste trabalho é desenvolver uma metodologia para o agrupamento inteligente de mensagens de alarmes por meio de técnicas de aprendizado de máquina e Processamento de Linguagem Natural, utilizando dados provenientes do *SCADA* de uma usina termoelétrica. Com os dados organizados em *clusters*, propôs-se a criação de visualizações à parte no sistema supervisório da planta, que permitem filtrar os alarmes pertencentes aos grupos mais críticos ou ordenar os registros pelo grau de relevância.

Além disso, o trabalho busca estabelecer discussões e explorar padrões de sequências temporais, possíveis causas raiz para o surgimento de determinados alarmes, bem como a redundância de registros em curtos intervalos de tempo. Outro aspecto abordado foi a mensuração dos subsistemas com maior quantidade de alarmes, e da volumetria de registros para cada diferentes granularidades de tempo, a fim de estimar o impacto na área operacional.

A Figura 1 contém um fluxograma que sintetiza o problema de pesquisa que a metodologia aqui desenvolvida busca solucionar: ausência de recursos adicionais para priorização dos alarmes. Agrupando os alarmes de uma maneira escalável e inteligente, é possível entender padrões e determinar os equipamentos ou subsistemas de maior criticidade e com maior frequência de falhas. Portanto, contribui-se diretamente para o aumento da eficiência operacional e para a tomada decisão pelos gestores, por meio do planejamento estratégico dos recursos destinados à manutenção dos ativos pertencentes aos grupos ou subsistemas mais críticos da planta.

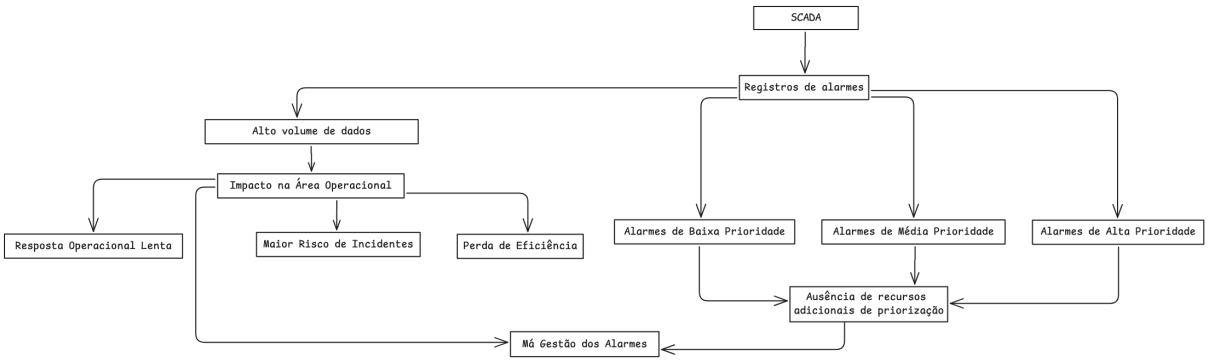


Figura 1 – Síntese da problemática de pesquisa

1.1.2 OBJETIVOS ESPECÍFICOS

Dos objetivos específicos, os principais deste trabalho são os seguintes:

- Trazer uma contribuição científica e apresentar um estudo de caso sobre o tema de análise de alarmes industriais e técnicas de Processamento de Linguagem Natural

(PLN) aplicadas a esta área;

- Agrupamento e clusterização de alarmes industriais de uma usina termoelétrica, buscando extrair informações relevantes de padrões de comportamento dos alarmes ao longo do tempo;
- Analisar os resultados da clusterização dos alarmes, detectar padrões recorrentes e informações relevantes para aprimorar a compreensão das condições operacionais e fornecer suporte à tomada de decisões;
- Transformar o aprendizado não supervisionado, presente na tarefa clusterização, em supervisionado, por meio da aplicação de algoritmos de classificação para rotulagem de novas amostras utilizando os *clusters* já obtidos;
- Avaliar a eficácia das técnicas de PLN aplicadas à análise de alarmes industriais e relevância dos insumos obtidos;

1.1.3 CONTRIBUIÇÕES

Este trabalho possui, como principal contribuição, o desenvolvimento uma metodologia para o agrupamento inteligente de alarmes industriais utilizando técnicas de aprendizado de máquina. Com essa abordagem, torna-se possível identificar registros semelhantes de forma escalável e filtrar os mais críticos, aprimorando a gestão de alarmes. Além disso, as análises realizadas oferecem insumos para a compreensão das causas raiz, padrões de ocorrência, e mecanismos para eliminar registros repetidos em curtos intervalos de tempo.

O estudo explora o uso de algoritmos de clusterização, classificação, e modelos de codificação de texto. Também incorpora conceitos fundamentais de processamento de linguagem natural (PLN), essenciais para o pré-processamento eficaz dos dados e para o tratamento de dados não numéricos em aplicações de aprendizado de máquina.

1.1.4 ESTRUTURA E ORGANIZAÇÃO DO TRABALHO

Além deste capítulo introdutório, o trabalho está composto por mais 4 capítulos, descritos a seguir:

- Capítulo 2: este capítulo discorre sobre alguns trabalhos que possuem interseção com esta pesquisa, trazendo os métodos e aplicações propostas pelos respectivos autores.
- Capítulo 3: este capítulo abordará sobre a fundamentação teórica e conceitos utilizados para o desenvolvimento deste trabalho.

- Capítulo 4: este capítulo engloba detalhes acerca da metodologia empregada na concepção e evolução desta pesquisa. Serão apresentadas informações acerca dos algoritmos e técnicas adotadas, além de abranger as etapas de pré-processamento que foram desenvolvidas.
- Capítulo 5: esta seção apresenta em detalhes os resultados obtidos por meio das técnicas e algoritmos empregados na metodologia.
- Capítulo 6: finalmente, neste capítulo, as conclusões deste estudo são delineadas, sintetizando os resultados alcançados e propondo aprimoramentos para o desenvolvimento de trabalhos futuros.

2 TRABALHOS RELACIONADOS

Em (CAI et al., 2019b), os autores comentam sobre a evolução significativa dos sistemas de alarmes industriais nos últimos anos, tanto no quesito de volumetria de registros, bem como da complexidade do gerenciamento, desafiando cada vez mais as habilidades de tomada de decisão dos operadores. Tais desafios de gerenciamento geralmente surgem devido à presença de alarmes mal configurados e excessivos. Isso demanda melhores ferramentas para que os operadores entendam as relações existentes entre vários eventos de alarmes, permitindo uma melhor tomada de decisão. Foi proposto um método de agrupamento de alarmes utilizando incorporação de palavras (*word embedding*). Os objetivos principais concentraram-se em fornecer insumos para a remoção de alarmes redundantes e oferecer uma base sólida para uma subsequente análise de causalidade e identificação de causa raiz dos alarmes. O método proposto foi aplicado aos eventos de alarme observados em uma planta de aquecimento e resfriamento central localizada em um *campus* universitário.

Em (FAHIMIPIREHGALIN; WEISS; VOGEL-HEUSER, 2020), os autores descrevem sobre a importância do sistema de gerenciamento de alarmes em plantas industriais de larga escala. Nestas, a alta conectividade resulta em dependências entre os alarmes gerados, levando a um aumento considerável dos registros em condições anormais, o que pode ser perigoso se os alarmes não forem tratados adequadamente pelos operadores. Foi proposto um método baseado em dados para detectar sequência causal dos alarmes, através dos arquivos de log. O método inclui a aplicação de clusterização para agrupar alarmes próximos no tempo. A análise de similaridade entre os clusters detectados permitiu identificar categorias de alarmes e analisar causas usando entropia de transferêcia.

Em (MANCA; DIX; FAY, 2021), os autores descreveram um método novo de análise de similaridade de inundação de alarmes (AFSA), que aborda desafios comuns, como a ambiguidade na ordem dos alarmes e a ocorrência de alarmes irrelevantes em subsequências similares. O método proposto utiliza séries de alarmes como entrada para duas abordagens de clusterização baseadas na frequência do termo–inverso da frequência nos documentos (TF-IDF) estendidas, uma técnica de redução de dimensionalidade e uma validação de outliers. Foram destacadas as propriedades dinâmicas ao utilizar variáveis de alarmes característicos e suas coativações. O abordagem adotada foi comparada a três outros métodos relevantes da literatura, demonstrando uma melhora na performance e robustez do AFSA ao integrar dados de séries de alarmes. Os resultados mostraram uma redução da influência da ambiguidade na ordem dos alarmes e dos alarmes irrelevantes, superando um desafio persistente nas pesquisas envolvendo gerenciamento de alarmes.

Em (RAVI; KULKARNI, 2023), foi abordado sobre a diversidade de tipos de

informações, como conteúdo de redes sociais, blogs, artigos de notícias e a necessidade de uma compreensão mais profunda destes dados disponíveis, de forma a utilizá-los de maneira prática, como detecções de eventos, análise de sentimentos, entre outras. Comentou-se também sobre a área de Processamento de Linguagem Natural (NLP), como sendo uma área de estudo que se dedica à análise de informações textuais, por meio do conhecimento linguístico e modelos de aprendizado de máquina que permitem classificar e agrupar textos semelhantes, por exemplo. *Word Embedding* foi apresentada como uma das técnicas mais importantes de NLP, permitindo representar palavras por meio de vetores numéricos, que capturam relações e aspectos semânticos no conteúdo do texto. No estudo realizado foram aplicadas várias técnicas de *Word Embedding* a *tweets* de canais de notícias populares, aplicando o algoritmo *K-Means* para agrupar os vetores resultantes. Os resultados apontaram que o modelo *Bidirectional Encoder Representations from Transformers* (BERT) alcançou a maior taxa de precisão quando combinado com o *K-Means*, evidenciando a eficácia do BERT na representação de palavras e na melhoria da análise e agrupamento de textos similares.

Em (SHEN; LIU, 2021), o artigo propôs um modelo de classificação de sentimentos que combina o modelo BERT com uma rede neural BLSTM (*Bidirectional Long Short-Term Memory*). Diferentemente dos métodos tradicionais, como *Word2Vec* e *Glove*, que não levam em consideração o contexto das palavras, o modelo utiliza o BERT para gerar vetores de palavras com informações contextuais. Posteriormente, a BLSTM extraiu as características associadas ao contexto e, por fim, um mecanismo de ponderação utilizado atribuiu pesos às informações extraídas para destacar as mais importantes. O modelo alcançou uma taxa de precisão de **89.17%** no conjunto de dados SST (*Stanford Sentiment Treebank*), demonstrando uma melhoria significativa na precisão em comparação com outros métodos.

No artigo (ZHANG et al., 2023), foi apresentado um novo método para análise de similaridade de "enchentes"de alarmes em instalações industriais, abordando as limitações de métodos existentes, que verificam apenas a similaridade entre *strings* de texto, desconsiderando as correlações entre ocorrências de alarmes. A análise de similaridade em "enchentes"de alarmes compara sequências de "enchentes"a fim de detectar padrões de ocorrência. Tais padrões podem oferecer informações que podem ser de grande valia na identificação de causa raiz das "enchentes"de alarmes. O método proposto utiliza *word embeddings* e a distância MSM (*move-split-merge*) para capturar a similaridade entre sequências de alarmes. A eficácia do método foi demonstrada por um estudo de caso com dados de alarmes de um modelo industrial público do processo de Monômero de Acetato de Vinila.

Após a tarefa de busca de publicações correlatas com este trabalho, notou-se que existem oportunidades para explorar o desenvolvimento de uma metodologia aplicando

as técnicas aqui propostas, como a utilização do modelo de codificação de texto BERT, bem como a clusterização do conteúdo de alarmes, a fim agrupar de maneira inteligente estes registros. Além disso, neste trabalho é realizado um estudo de caso envolvendo uma Usina Termoelétrica, e propõe mitigar o problema de gestão de alarmes, através da filtragem dos registros mais relevantes, agrupados de maneira inteligente pelos algoritmos de clusterização e classificação implementados.

3 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, será abordada a fundamentação teórica dos principais temas que sustentam o desenvolvimento deste trabalho.

Será apresentada uma revisão bibliográfica a respeito das técnicas de Processamento de Linguagem Natural utilizadas, um resumo dos dados que foram utilizados no desenvolvimento da pesquisa, bem como o tema de Alarmes Industriais. Serão explorados os procedimentos para a realização do pré-processamento do conteúdo das mensagens dos alarmes, algoritmos de clusterização e de classificação.

3.1 SISTEMA SCADA

Os sitemas **SCADA** (Supervisory Control and Data Acquisition) existem desde o advento dos sistemas de controle. Os primeiros sistemas "SCADA"utilizavam a aquisição de dados por meio de painéis de medição, luzes e registradores de gráficos (MEDIDA, 2008). O controle supervisório era exercido pelo operador, acionando manualmente vários botões de controle. Estes dispositivos foram e ainda são utilizados na supervisão, controle e aquisição de dados em plantas, fábricas e instalações de geração de energia.

SCADA refere-se à combinação de telemetria e aquisição de dados. O SCADA engloba a coleta das informações, transferindo-as de volta para o local central, realizando qualquer análise e controle necessários e, em seguida, exibindo essas informações em várias telas ou mostradores do operador. As ações de controle necessárias são então transmitidas de volta para o processo. O CLP, ou Controlador Lógico Programável, ainda é um dos sistemas de controle mais amplamente utilizados na indústria. À medida que as necessidades cresceram para monitorar e controlar mais dispositivos, os CLPs foram distribuídos e os sistemas se tornaram mais inteligentes e menores em tamanho. CLPs e Sistemas de Controle Distribuído são utilizados de acordo com Figura 2.

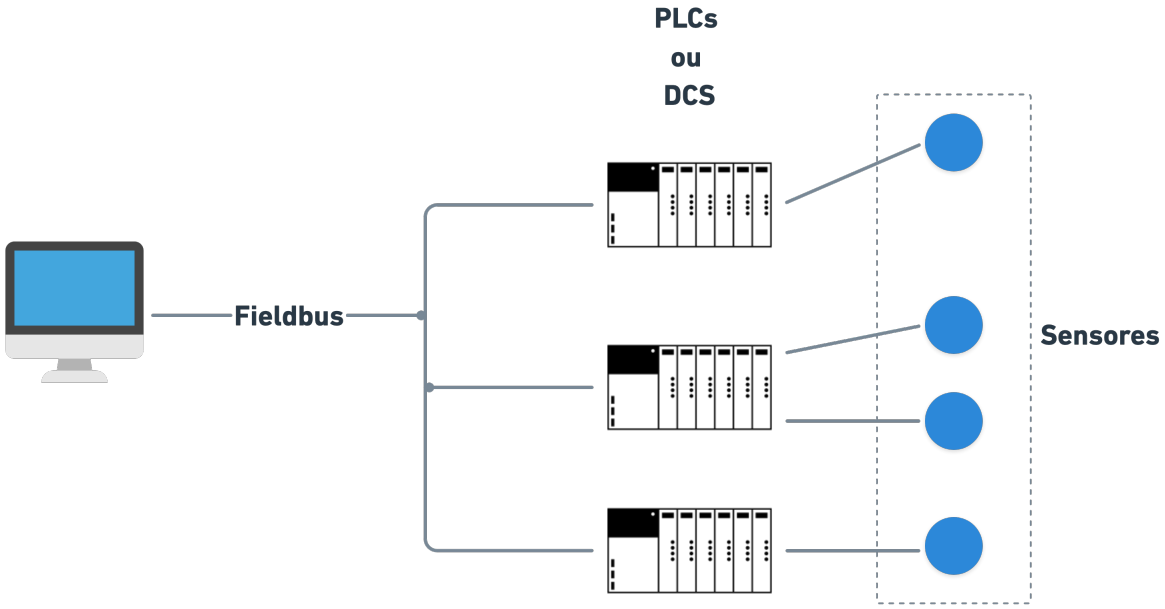


Figura 2 – PC para PLCs ou DCS e Sensores com rede de comunicação *fieldbus*.

As vantagens do sistema **SCADA PLC/DCS** são:

- O computador pode gravar e armazenar uma quantidade muito grande de dados.
- Os dados podem ser exibidos de qualquer maneira que o usuário desejar.
- Milhares de sensores em uma ampla área podem ser conectados ao sistema.
- O operador pode incorporar simulações de dados reais ao sistema.
- Muitos tipos de dados podem ser coletados dos RTUs.
- Os dados podem ser visualizados de qualquer lugar, não apenas no local.

As desvantagens são:

- O sistema é mais complicado do que o tipo de sensor para painel.
- Diferentes habilidades operacionais são necessárias, como analistas de sistema e programadores.
- Com milhares de sensores, ainda há muitos fios para lidar.
- O operador só pode ver até onde o PLC alcança.

3.2 ALARMES INDUSTRIAIS

Um alarme é considerado como um aviso ao operador iniciado por uma variável de processo (ou medição) que ultrapassa um limite definido ao se aproximar de um valor indesejável ou inseguro. O aviso inclui sons audíveis, indicações visuais (por exemplo, luzes piscando e texto, mudanças na cor do fundo ou do texto, e outras alterações gráficas ou pictóricas) e mensagens. O problema anunciado requer ação do operador. A condição de estar *em alarme* é comunicada através de sons intrusivos e avisos colocados em unidades de exibição de vídeo e outros dispositivos para chamar a atenção. O operador pode gerenciar esses sons e avisos apenas por meio de ações específicas de "silenciar o alarme"ou "reconhecer o alarme", utilizando a infraestrutura existente e planejada da plataforma de alarme. Normalmente, essa plataforma é parte integrante da infraestrutura do sistema de controle de processos (PCs) (ROTHENBERG, 2009).

Alarmes industriais são parte integrante dos sistemas SCADA, servindo como notificações para condições anormais ou falhas nos processos industriais (QIU et al., 2011). Sistemas **SCADA** (*Supervisory Control and Data Acquisition*) desempenham um papel crucial em ambientes industriais, fornecendo monitoramento e controle em tempo real de vários processos (SVERKO; GRBAC; MIKUC, 2022).

Em (GONZALEZ; REDER; MELERO, 2016), os autores comentam sobre os impactos causados pelo tempo de inatividade e falha das turbina eólicas na rentabilidade de um parque eólico. A detecção precoce de falhas pode facilitar a mudança de da manutenção corretiva para a abordagem preditiva. Foi apresentada uma metodologia econômica para combinar várias técnicas de análise de alarmes, utilizando dados do **SCADA**, a fim de detectar falhas em componentes. A abordagem categoriza os alarmes de acordo com uma taxonomia previamente analisada, transformando dados em informações relevantes para avaliar o status dos componentes. Diferentes técnicas de análise de alarmes foram aplicadas para dois propósitos: a avaliação da capacidade do sistema de alarmes SCADA para detectar desvios e a investigação da relação entre problemas de componentes que são seguidas por ocorrências de problemas em outros. O estudo destacou a relação entre o comportamento defeituoso em diferentes componentes, entre falhas distintas e condições ambientais adversas.

Em (YANG et al., 2023), os autores comentam sobre a grande relevância dos sistemas de alarmes na eficiência de instalações industriais complexas. No entanto, devido ao aumento do tamanho das plantas e a crescente complexidade dos processos industriais, o excesso de alarmes configura um grande desafio para os sistemas que os integram. A extração de padrões de um banco de dados com excesso de registros de alarmes pode auxiliar na análise da causa raiz destes, no suporte à decisão e na configuração de um modelo de supressão de alarmes. Contudo, devido ao grande tamanho do banco de dados de alarmes e

o ao problema da ambiguidade de sequência dos alarmes, os algoritmos existentes sofrem de sobrecarga computacional excessiva, padrões de alarme incompletos e saídas redundantes. Para resolver tais problemas, os autores propuseram um método de extração de padrões de alarme baseado no algoritmo *PrefixSpan* aprimorado. Inicialmente propôs-se uma estratégia de pré-casamento baseada em prioridade para agrupar sequências semelhantes com antecedência. Em segundo lugar, foi aprimorado o *PrefixSpan* considerando intervalos de tempo para tolerar a ambiguidade de ordem de curto prazo nas sequências de alarmes excessivos. Por fim, foi utilizada uma metodologia para encontrar padrões de alarmes representativos. Foi avaliada a eficácia e aplicabilidade do método proposto usando um banco de dados com registros excessivos de alarmes de uma unidade real de hidrogenação de diesel.

Pesquisadores têm se concentrado em analisar dados de alarme do SCADA para melhorar a confiabilidade e o desempenho dos sistemas industriais. Por exemplo, em (QIU et al., 2011) foi realizado um estudo sobre análise de alarmes de turbinas eólicas no SCADA para melhorar a confiabilidade. Foram analisados sinais do SCADA, como saída de energia, velocidade do vento e temperaturas de rolamento, para desenvolver algoritmos de monitoramento de condições para detectar falhas incipientes. Isso ajudou a reduzir o tempo de inatividade, aumentar a disponibilidade e otimizar os cronogramas de manutenção.

A crescente e considerável volumetria de alarmes do SCADA pode representar desafios para operadores e mantenedores. Principalmente, a questão das enchentes de alarmes em turbinas eólicas operando em grandes parques eólicos. Em (QIU et al., 2011) foi proposta uma metodologia para priorizar os alarmes com base em um padrão da indústria, permitindo que os operadores se concentrem nos alarmes mais críticos e os abordem de forma eficaz. Essa abordagem ajuda a lidar com o alto volume de alarmes e a melhorar a eficiência no tratamento de alarmes.

A análise dos dados de alarme provenientes do SCADA também envolve a detecção e diagnóstico de alarmes falsos. Em (MARUGAN; MÁRQUEZ, 2019) foi feita uma abordagem de análise avançada para detectar e identificar falhas e alarmes falsos em turbinas eólicas. Para tanto, utilizaram correlação e ajustes de curvas para reconhecer padrões e comportamentos anormais em variáveis do SCADA, permitindo a identificação de alarmes falsos e falhas em componentes (MARUGAN; MÁRQUEZ, 2019). Essa análise ajuda a reduzir alarmes falsos, melhorar a confiabilidade do sistema e otimizar os esforços de manutenção.

3.2.1 TÉCNICAS DE PRÉ-PROCESSAMENTO DE TEXTO

O pré-processamento é uma tarefa fundamental e de extrema relevância em mineração de texto, Processamento de Linguagem Natural (NLP) e recuperação de informação

(IR). Na área de Mineração de Texto, o pré-processamento de dados é usado para extrair conhecimentos relevantes partindo de dados não estruturados (GURUSAMY; KANNAN, 2014). Além disso, envolve a limpeza e transformação de dados textuais não estruturados para prepará-los para algoritmos de aprendizado de máquina (KALRA; AGGARWAL, 2018).

Tarefas comuns de pré-processamento incluem remoção de números, links e e-mails; conversão de texto para minúsculas; lematização; e tokenização (SILVA et al., 2023; KALRA; AGGARWAL, 2018). Essas técnicas podem impactar significativamente o desempenho de tarefas de classificação de texto (ELER et al., 2018).

Neste trabalho, foram aplicados alguns métodos de pré-processamento de texto, como Tokenização, remoção de *Stop Words* e Lematização para documentos de texto.

3.2.1.1 TOKENIZAÇÃO

Tokenização é o processo de dividir um fluxo de texto em palavras, frases, símbolos ou outros elementos significativos chamados *tokens*. O objetivo da tokenização é a exploração das palavras em uma sentença. A lista de tokens torna-se a entrada para o processamento subsequente, como análise sintática ou mineração de texto. A tokenização é útil tanto na linguística (onde é uma forma de segmentação de texto) quanto na Ciência da Computação, onde faz parte da análise lexical (GURUSAMY; KANNAN, 2014). Na Figura 3 é possível observar o processo de tokenização de palavras em uma sentença.

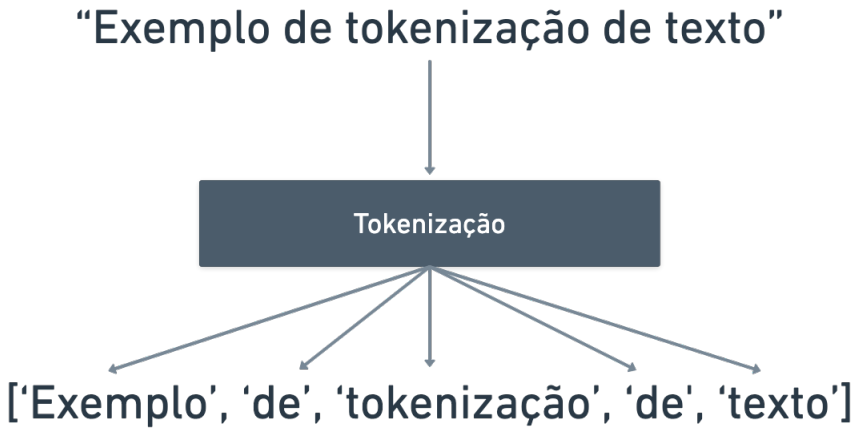


Figura 3 – Tokenização de palavras.

3.2.1.2 REMOÇÃO DE STOP WORDS

Muitas palavras em documentos aparecem de forma muito recorrente, porém são essencialmente sem significado, pois são utilizadas para unir palavras em uma frase. Em geral é compreendido que as *stop words* não contribuem para o contexto ou conteúdo dos documentos textuais. Devido à alta frequência, a presença destas palavras na mineração

de texto provoca um obstáculo na compreensão do conteúdo dos documentos. As *stop words* são palavras comuns, como "e", "são", "isso", etc., que não são úteis na classificação de texto e, dessa forma, precisam ser removidas. No entanto, o desenvolvimento de uma lista de *stop words* é difícil e inconsistente entre as fontes textuais. Este processo também reduz os dados de texto e melhora o desempenho do sistema. Em geral os documentos de texto lidam com essas palavras que não são úteis em aplicações envolvendo mineração de texto (GURUSAMY; KANNAN, 2014). A Figura 4 ilustra o processo ilustra os *tokens* após o processo de remoção das *stop words*.

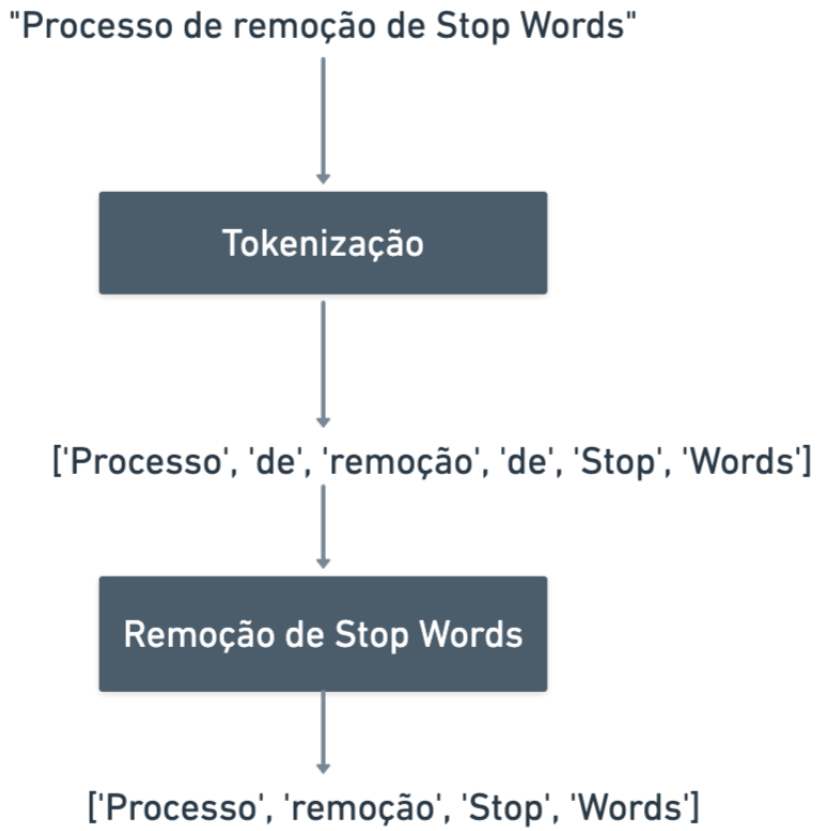


Figura 4 – Remoção de *Stop Words*.

3.2.1.3 LEMATIZAÇÃO

A lematização é um método que utiliza o vocabulário e a análise morfológica das palavras, e busca remover terminações inflexionais, retornando assim as palavras à sua forma de dicionário. Para que o processo seja feito de forma correta, é analisado se as palavras da consulta são empregadas como verbos ou substantivos. A lematização também ajuda a corresponder sinônimos pelo uso de um *thesaurus*, de modo que quando há a busca por "quente", a palavra "morno"também seja correspondida. A técnica de lematização tem sido utilizada em várias línguas para recuperação de informações (BALAKRISHNAN; ETHEL, 2014). Na Figura 5 é possível visualizar a utilização da técnica de Lematização.

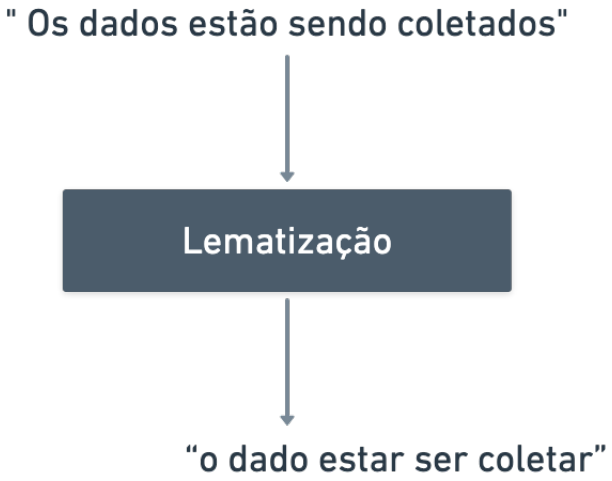


Figura 5 – Técnica de lematização.

3.2.1.4 REMOÇÃO DE PALAVRAS NÃO RECONHECIDAS PELO MODELO DE LINGUAGEM

Utilizou-se o módulo *spacy* e o modelo de linguagem em inglês *en_core_web_sm* no processamento das mensagens dos alarmes. O módulo *spacy* é uma biblioteca em Python que fornece uma infraestrutura para processamento de linguagem natural eficiente e de alto desempenho. Foi carregado o modelo de linguagem em inglês para utilização posterior no processamento.

Foram realizadas iterações em cada token do texto processado para verificar se as palavras eram reconhecidas pelo modelo de linguagem. As palavras reconhecidas foram mantidas na lista.

3.2.2 MODELO DE LINGUAGEM BERT

O modelo de representação de linguagem *BERT* significa Representações de codificador bidirecional de transformadores. Diferentemente de alguns outros modelos, o BERT foi projetado para pré-treinar representações bidirecionais profundas a partir de texto não rotulado, condicionando conjuntamente tanto o contexto à esquerda quanto à direita em todas as camadas. Como resultado, o modelo BERT pode ser ajustado com apenas uma camada de saída adicional para criar modelos de última geração que permitem a utilização em muitas tarefas, como respostas a perguntas e inferência de linguagem, sem modificações substanciais na arquitetura específica da tarefa (DEVLIN et al., 2018b).

Existem duas estratégias existentes para aplicar representações de linguagem pré-treinadas em tarefas subsequentes: *fine-tuning* e *feature-based*. A abordagem baseada em *features*, como o ELMo (PETERS et al., 2018), utiliza arquiteturas específicas de tarefas que incluem as representações pré-treinadas como características adicionais. A

abordagem de *fine-tuning*, como o Generative Pre-trained Transformer (OpenAI GPT) (RADFORD et al., 2018), introduz parâmetros mínimos específicos da tarefa e é treinada nas tarefas subsequentes simplesmente ajustando todos os parâmetros pré-treinados. As duas abordagens compartilham a mesma função objetivo durante o pré-treinamento, onde utilizam modelos de linguagem unidirecionais para aprender representações gerais de linguagem.

Existem duas etapas na implementação do BERT: *pre-training* e *fine-tuning*. Durante o *pre-training*, o modelo é treinado com dados não rotulados em diferentes tarefas de pré-treinamento (DEVLIN et al., 2019). Para *fine-tuning*, o modelo BERT é inicializado com os parâmetros pré-treinados, e todos os parâmetros passam pelo *fine-tuning* através de dados rotulados das tarefas subsequentes. Cada uma destas possui os modelos ajustados separadamente, embora sejam inicializados com os mesmos parâmetros pré-treinados.

Para que o BERT lide com uma variedade de tarefas subsequentes, na entrada é possível representar de forma inequívoca tanto uma sentença única quanto um par (ou seja, ⟨Pergunta, Resposta⟩) em uma sequência de *tokens* (DEVLIN et al., 2018a). Ao longo dessa tarefa, uma "sentença"pode ser um trecho arbitrário de qualquer sequência contínua de texto, que pode não ser uma frase completa ou correta gramaticalmente. Uma "sequência"refere-se à sequência de *tokens* de entrada para o BERT, que pode ser uma única sentença ou duas sentenças combinadas. Para um dado *token*, sua representação de entrada é construída somando os *embeddings* correspondentes de *token*, segmento e posição. Uma visualização desta construção é ilustrada na Figura 6.

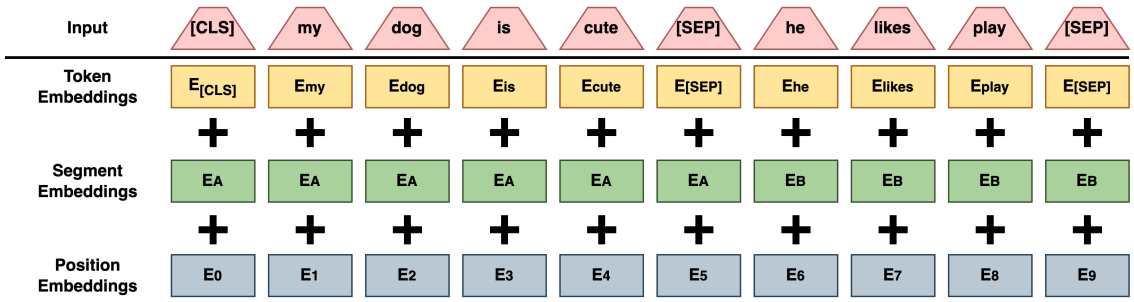


Figura 6 – Representação de entrada (BERT). Adaptado de (DEVLIN et al., 2018b).

Para auxiliar na distinção entre as duas frases durante o treinamento, a entrada é processada da seguinte forma antes de ser inserida no modelo:

- Um *token* [**CLS**] é inserido no início da primeira frase e um *token* [**SEP**] é inserido no final de cada frase.
- Um *embedding* de sentença, indicando sentença **A** ou sentença **B**, é adicionado a cada token. *Embeddings* de sentença, em termos de conceito, são semelhantes aos

embeddings de token, mas com um vocabulário de 2.

- Um *embedding* posicional é adicionado a cada token para indicar sua posição na sequência.

Os *embeddings* de entrada são a soma dos *embeddings* dos tokens, os *embeddings* de segmentação e *embeddings* de posição.

Para prever se a segunda frase está de fato conectada à primeira, os seguintes passos são realizados (DEVLIN et al., 2018b):

- A sequência de entrada completa passa pelo modelo *Transformer*.
- A saída do token **[CLS]** é transformada em um vetor com dimensões $2x1$, usando uma camada de classificação simples (matrizes de pesos e *biases*).
- Cálculo probabilidade de *IsNextSequence* com Softmax.

3.2.3 WORD EMBEDDINGS

Word embeddings são representações vetoriais de palavras que capturam relacionamentos semânticos e sintáticos em um espaço de baixa dimensão (LEBRET, 2016). Esses *embeddings* provaram ser muito relevantes para várias tarefas de PLN, incluindo análise de sentimentos, classificação de texto e geração de frases (SULEIMAN; AWAJAN, 2018; LEBRET, 2016). Diferentes abordagens para criar *word embeddings* incluem métodos tradicionais, estáticos e contextualizados, com modelos como o BERT contribuindo significativamente para *embeddings* contextualizados (NEELIMA; MEHROTRA, 2023).

Existem algumas técnicas de *word embeddings* bastante conhecidas:

- Term Frequency-Inverse Document Frequency (TF-IDF);
- Bag of Words(BoW);
- Word2Vec;
- Global Vector for Word Representation (Glove);

O *TF-IDF*, Frequência do Termo e da Frequência Inversa do Documento, é um dos métodos mais utilizados para representar dados textuais. No entanto, o *TF-IDF* não pode considerar a posição e o contexto de uma palavra em uma frase. Já o modelo *Bidirectional Encoder Representations from Transformers* (BERT) pode produzir uma representação de texto que incorpora a posição e o contexto de uma palavra em uma frase. Além disso, vários métodos de extração de características e normalização também são aplicados para

a representação de dados para texto (SUBAKTI; MURFI; HARIADI, 2022). Ainda em (SUBAKTI; MURFI; HARIADI, 2022), as simulações realizadas mostraram que o BERT superou o *TF-IDF* em 28 das 36 métricas avaliadas.

Neste trabalho, a geração de *word embeddings* foi através da técnica *Bidirectional Encoder Representations from Transformers* (BERT).

3.2.4 ALGORITMOS DE CLUSTERIZAÇÃO

A clusterização é uma tarefa de grande importância na mineração de dados, e desempenha um papel fundamental no processo de Descoberta de Conhecimento em Bancos de Dados (KDD). Trata-se de uma técnica de aprendizado não supervisionado, utilizada para análise exploratória de dados, que permite identificar padrões em conjuntos de dados que não podem ser facilmente categorizados. O objetivo principal da clusterização é agrupar objetos semelhantes com base em características comuns, garantindo que os elementos pertencentes aos mesmos grupos estejam mais próximos uns dos outros do que dos objetos pertencentes a outros *clusters* (BINDRA; MISHRA, 2017; BINDRA; MISHRA; SURYAKANT, 2018). Vários algoritmos de clusterização foram desenvolvidos sob diferentes paradigmas para lidar com variados tipos de dados e características do agrupamento. Avanços recentes em algoritmos de clusterização visam integrar diferentes abordagens e lidar com dados sequenciais de alta dimensão que possuem múltiplos relacionamentos (BINDRA; MISHRA, 2017; BINDRA; MISHRA; SURYAKANT, 2018). A clusterização possibilita a extração de informações implícitas em grandes conjuntos de dados, cujo volume aumentou significativamente com os avanços tecnológicos (PETTA et al., 2020).

3.2.5 ALGORITMO K-MEANS

O algoritmo *K-means* é uma técnica de aprendizado não supervisionado amplamente utilizada para agrupar dados em *clusters* (GÉRON, 2021). É um algoritmo iterativo que busca dividir os dados em k grupos distintos, onde k é um valor pré-definido pelo usuário. O funcionamento básico do *K-means* é o seguinte:

1. Inicialização: Seleção aleatória de k centroides iniciais no espaço de dados.
2. Atribuição: Cálculo da distância entre cada ponto de dados e os centroides e atribuição de cada ponto ao *cluster* com o centroide mais próximo.
3. Atualização: Recálculo dos centroides de cada *cluster* como a média dos pontos atribuídos a ele.

4. Iteração: Repetição dos passos 2 e 3 até que a convergência seja alcançada, isto é, quando não ocorrerem mais alterações nos centroides, ou o número máximo de iterações for atingido.

O algoritmo *K-means* é eficiente e de simples implementação, porém sua eficácia depende muito da escolha adequada do número de *clusters* e da inicialização dos centroides. Além da facilidade de implementação, a vasta quantidade de aplicações do *K-means* na literatura, para clusterização de texto, motivaram a utilização deste algoritmo.

Na clusterização de texto, o *K-means* pode ser aplicado usando técnicas de Processamento de Linguagem Natural (NLP). Primeiramente, os textos são pré-processados, incluindo etapas como *tokenização*, remoção de *stop words*, vetorização etc. Em seguida, o algoritmo *K-means* é aplicado para agrupar os textos em *k clusters* com base em suas características, como frequência de termos ou representação vetorial.

Para exemplificar o funcionamento do *K-means*, foi escrito um código simples em Python com um conjunto de dados de pontos bidimensionais. Na Figura 7, é possível visualizar o agrupamento dos pontos em dois *clusters* com base nas suas similaridades.

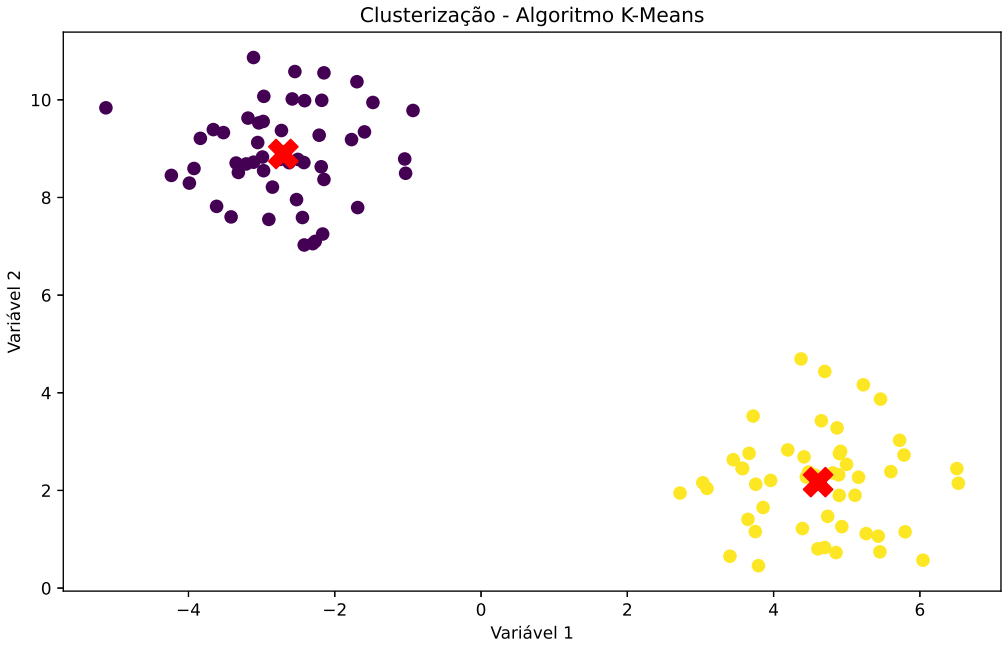


Figura 7 – Clusterização com Algoritmo *K-Means*.

Para a construção da Figura 7, foi gerado um conjunto de dados com 100 pontos bidimensionais divididos em duas distribuições gaussianas distintas. Em seguida, foi aplicado o algoritmo *K-means* com agrupamento em dois *clusters*. Os pontos coloridos do

gráfico mostram os *clusters* aos quais foram atribuídos, e os centroides de cada *cluster* estão marcados com um **"X"** vermelho.

3.2.5.1 MÉTODOS DE INICIALIZAÇÃO DO CENTROIDE

Caso já haja alguma aproximação de onde os centroides devem estar posicionados (se já tenha sido aplicado algum outro algoritmo de clusterização), é possível especificar a lista de centroides como um array **NumPy** (GÉRON, 2021). Outra solução é executar o algoritmo diversas vezes com diferentes inicializações aleatórias e manter a melhor solução. O número de inicializações aleatórias é controlado pelo hiperparâmetro *n_init*, utilizando a biblioteca *Scikit-Learn*. Por padrão este número é igual a 10, ou seja, o algoritmo é executado 10 vezes quando ocorre a chamada do método *fit()*, e a *Scikit-Learn* armazena a melhor solução. Para detectar a melhor solução, é utilizada uma métrica de desempenho, denominada *inércia* do modelo, que é a distância quadrada média entre cada instância e o centroide mais próximo. Uma das melhorias mais importantes no algoritmo K-means, o K-means++, foi proposta em um artigo de 2006 (*https://homl.info/37*) por David Arthur e Sergei Vassilvitskii. Eles apresentaram uma proposta de etapa de inicialização mais eficiente, que tende a selecionar centroides distantes um do outro, e tal melhoria torna o *K-means* bem menos propenso a convergir para uma solução distante do ideal. Arthur e Vassilvitskii demonstraram que o cálculo adicional durante a etapa de inicialização mais inteligente é promissor, uma vez que possibilita reduzir definitivamente o número de vezes que o algoritmo precisa ser executado antes de encontrar a solução ideal. O algoritmo de inicialização *K-means++* funciona da seguinte forma:

1. Seleção de um centroide $c^{(1)}$, escolhido uniformemente e de maneira aleatória no conjunto de dados.

2. Utilizar um novo centroide $c^{(i)}$, escolhendo uma instância $x^{(i)}$ com a probabilidade $\frac{D(x^{(i)})^2}{\sum_{j=1}^{m} D(x^{(j)})^2}$, em que $D(\mathbf{X}^{(i)})$ é a distância $x^{(i)}$ e o centroide mais próximo que já foi escolhido. Essa distribuição de probabilidade assegura que instâncias mais distantes dos centroides já escolhidos tenham maior chance de serem selecionadas como centroides.

3. Repetir a etapa anterior até que todos os k centroides tenham sido escolhidos.

A classe *KMeans* da biblioteca *scikit-learn* utiliza este método de inicialização por padrão. Caso haja a necessidade a necessidade de empregar o método original, é possível definir o parâmetro *init* como ***random***.

3.2.6 DETERMINANDO O NÚMERO IDEAL DE CLUSTERS

Em geral, definir o número de *clusters* (valor de k) não é uma tarefa fácil (GÉRON, 2021). Caso o valor de k escolhido seja incorreto, os resultados podem ser impactados negativamente. Não é suficiente selecionar o modelo com a menor inércia, pois esta não é uma boa métrica de desempenho quando se trata de escolher o k, já que a inércia fica cada vez menor à medida que o valor de k aumenta. Na prática, quanto mais *clusters* existirem, mais próxima será a instância do centroide mais próximo e, consequentemente, menor será a inércia. A Figura 8 exemplifica a representação da inércia em função de k.

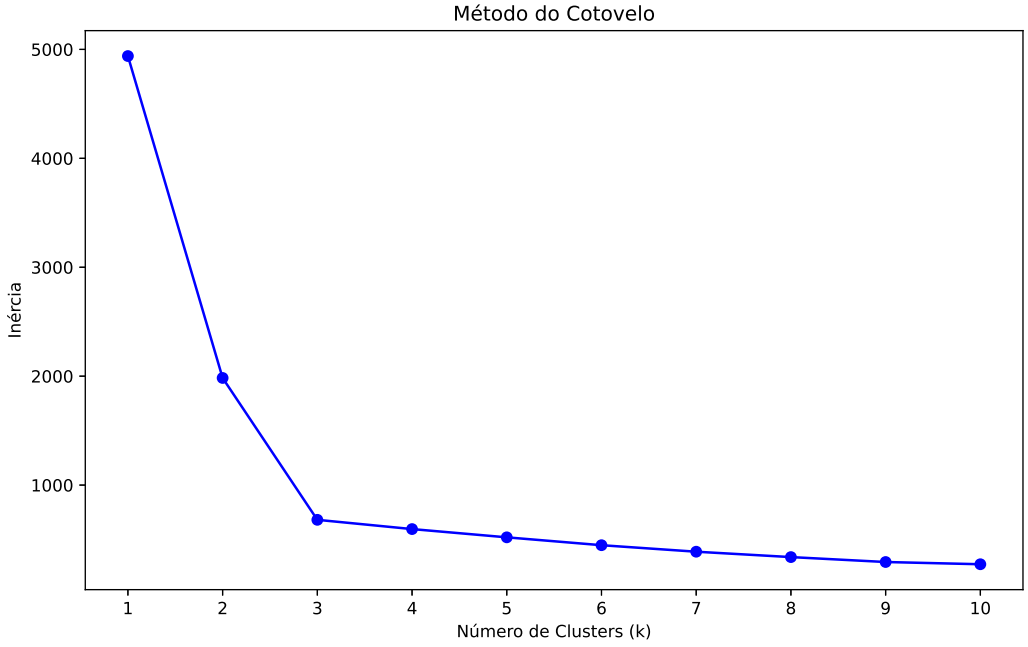


Figura 8 – Inércia em função de k.

Conforme é possível observar na Figura 8, a inércia diminui rapidamente ao passo que o valor de k foi aumentado até 3, e depois diminui mais lentamente conforme aumentamos k. Esta curva tem aproximadamente o formato de um braço e existe um "cotovelo"em $k = 3$. Assim, caso não houvesse o conhecimento prévio, o número 3 seria uma boa escolha.

Esta técnica de escolher o menor valor para o número de *clusters* é um tanto simplista (GÉRON, 2021). Uma abordagem mais precisa (porém mais onerosa em termos computacionais) é utilizar o coeficiente de *sillhouette* médio em todas as instâncias. O coeficiente de *silhouette* de uma instância é igual a $(b-a)/max(a,b)$, em que a é a distância média para as outras instâncias no mesmo *cluster* (ou seja, a distância média intracluster) e b é o valor mínimo dentre as médias das distâncias entre uma amostra e todos os outros

pontos de um outro *cluster*. O coeficiente de *silhouette* pode variar entre -1 e +1. Um coeficiente próximo a **+1** significa que a instância está dentro de seu próprio *cluster* e distante de outros *clusters*, enquanto um coeficiente próximo a **0** significa que está próximo a uma fronteira de *cluster* e, por último, um coeficiente próximo a **-1** significa que a instância pode ter sido atribuída ao *cluster* errado. Na Figura 9 é possível visualizar a qualidade do agrupamento de dados através do coeficiente de *silhouette*, variando o número de *clusters*.

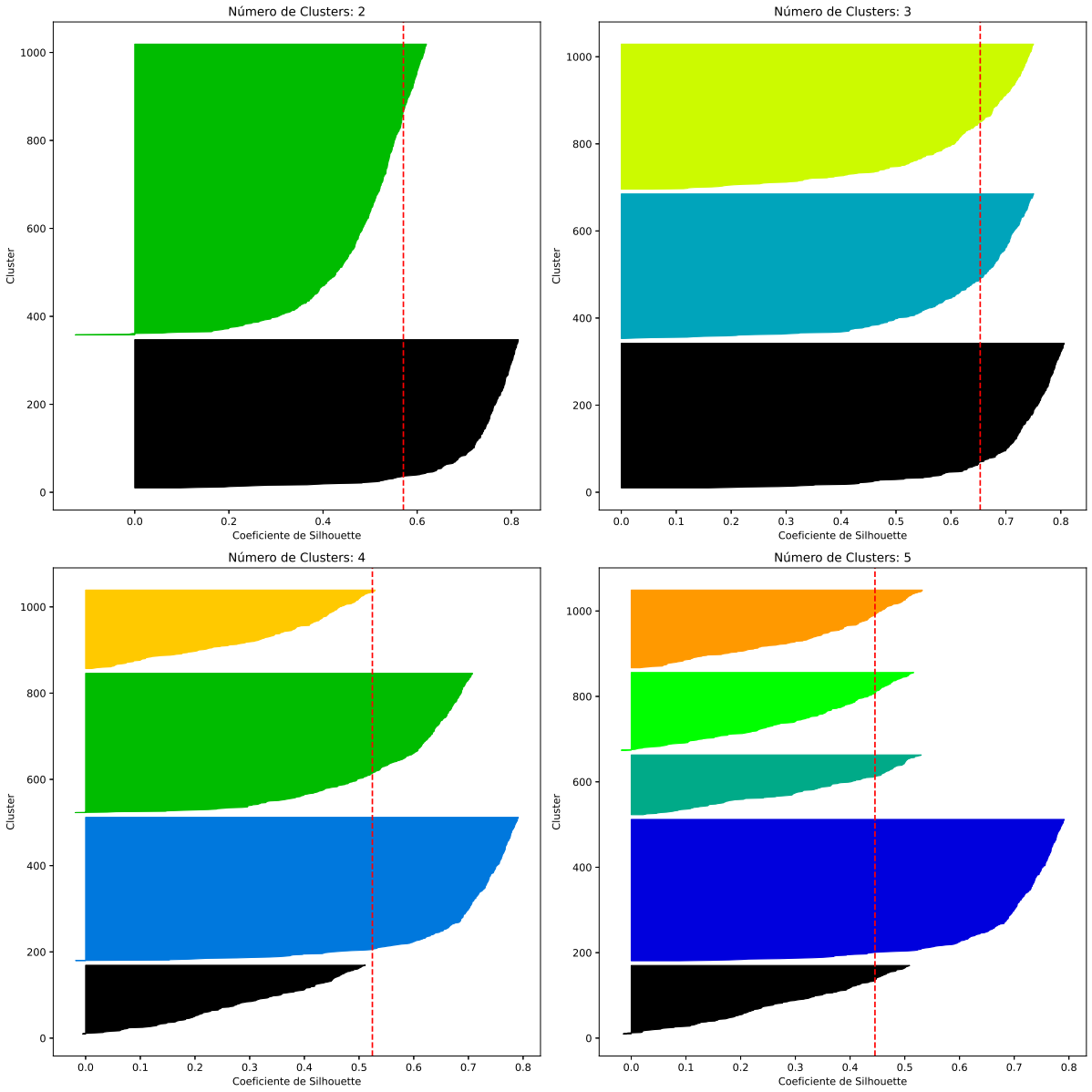


Figura 9 – Análise do coeficiente de *silhouette* para 4 *clusters*.

Para cada número de *clusters*, o gráfico de barras horizontais mostra a distribuição dos coeficientes de *silhouette* para os pontos dos grupos correspondentes. Cada barra representa um *cluster*, e a respectiva largura corresponde ao valor do coeficiente de

silhouette. A linha vermelha vertical pontilhada representa o valor médio do coeficiente de *silhouette* para o número de *clusters* especificado. Este valor é uma medida da qualidade geral da clusterização. Quanto maior o valor, melhor a qualidade do agrupamento dos dados.

O número ótimo de *clusters* é aquele que maximiza o valor médio do coeficiente de *silhouette*. Na Figura 9, isso é indicado pelo gráfico com a linha vermelha pontilhada mais à direita.

3.2.7 MINI BATCH K-MEANS

O *Mini Batch K-Means* é uma variante do algoritmo **K-Means**, que utiliza *mini-lotes* para reduzir o tempo de computação, ao mesmo tempo que tenta otimizar a função objetivo. Mini-lotes são subconjuntos de dos dados de entrada, amostrados aleatoriamente em cada iteração de treinamento. Estes mini-lotes reduzem drasticamente o esforço computacional para convergir para uma solução local. Ao contrário de outros algoritmos que reduzem o tempo de convergência do *K-Means*, o *Mini Batch K-Means* geralmente produz resultados que são ligeiramente piores que o algoritmo padrão (GÉRON, 2021).

O algoritmo itera entre dois passos principais, semelhantes ao *K-Means* padrão. No primeiro passo, amostras são retiradas aleatoriamente do conjunto de dados para formar um mini-lote. Estas amostras são atribuídas ao centroide mais próximo. No segundo passo, os centroides são atualizados. Diferente do *K-Means*, isso é realizado em uma base amostral. Para cada amostra no mini-lote, o centroide atribuído é atualizado tomando a média contínua da amostra e de todas as amostras anteriores atribuídas a este centroide. Isso tem o efeito na taxa de mudança no centroide ao longo do tempo. Essas etapas são executadas até a convergência ou até que um número pré-determinado de iterações seja alcançado.

O *Mini Batch K-Means* converge mais rápido que o *K-Means*, mas a qualidade dos resultados é reduzida. Na prática, esta diferença pode ser bastante pequena. Foi construído um código em Python com dados aleatórios a fim de comparar ambos os algoritmos.

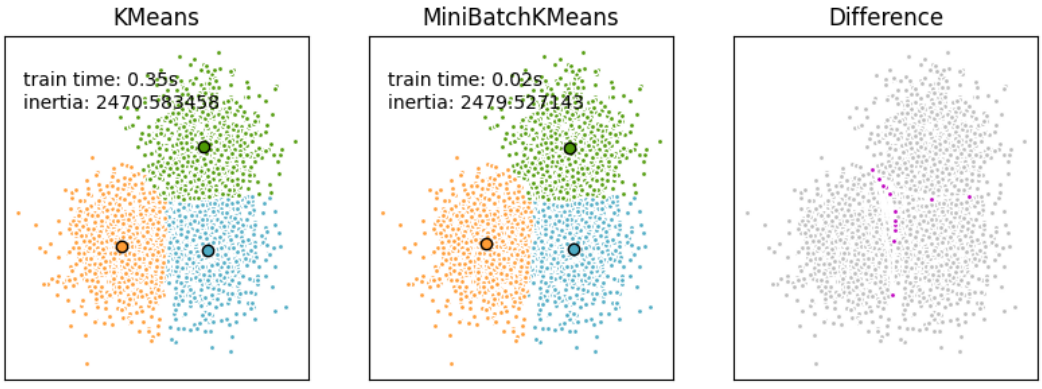


Figura 10 – Comparação - *K-Means* padrão e *Mini Batch K-Means*.

3.2.8 ALGORITMO *SUPPORT VECTOR MACHINE*

Support Vector Machines (SVM) têm sido usadas em muitas tarefas de Processamento de Linguagem Natural (NLP) (LI; BONTCHEVA; CUNNINGHAM, 2009). Trata-se de um algoritmo de aprendizado de máquina supervisionado, que tem alcançado um excelente desempenho em muitas tarefas envolvendo aprendizado. Em particular, SVM é um algoritmo de aprendizado popular para tarefas de Processamento de Linguagem Natural (NLP), como marcação de classe gramatical (GIMÉNEZ; MARQUEZ, 2003; NAKAGAWA; KUDO; MATSUMOTO, 2001), desambiguação de sentido de palavras (LEE; NG; CHIA, 2004), segmentação de frases nominais (KUDO; MATSUMOTO, 2000), extração de informação (ISOZAKI; KAZAWA, 2002; LI; BONTCHEVA; CUNNINGHAM, 2004), extração de relações (ZHOU et al., 2005), rotulagem de funções semânticas (HACIOGLU et al., 2004) e análise de dependência (FUJIO; MATSUMOTO, 1998; YAMADA; MATSUMOTO, 2003). A maioria das aplicações adotam os mesmo passos: primeiro, transformam o problema em uma tarefa de classificação multiclasse; depois, convertem o problema multiclasse em vários problemas de classificação binária; em seguida, um classificador SVM é treinado para cada classificação binária; e, finalmente, os resultados dos classificadores são combinados para obter a solução para o problema original de NLP.

O SVM se destaca no tratamento de dados de alta dimensão ao definir e maximizar a margem entre o limite de decisão e as amostras de treinamento (CRUZ, 2000). O SVM cria um hiperplano ideal que maximiza a distância para todos os exemplos de treinamento, resultando em fortes capacidades de generalização (COSTA; CASTRO, 2019). O algoritmo utiliza vetores de suporte, que são pontos de dados mais próximos do limite de decisão, para determinar a separação ideal entre classes (BHARADWAJ; PRAKASH; KANAGACHIDAMBARESAN, 2021). A eficácia do SVM foi demonstrada em várias aplicações, incluindo reconhecimento facial, detecção de intrusão e classificação de texto (ESPINOZA et al., 2021). Pesquisas recentes mostraram que ajustar os parâmetros do *kernel* por meio de abordagens de aprendizado de métricas pode melhorar significativamente o desempenho do classificador SVM (COSTA; CASTRO, 2019). No geral, o SVM oferece alta precisão em comparação a outros classificadores, como regressão logística e árvores de decisão (ESPINOZA et al., 2021).

Tarefas de Processamento de Linguagem Natural (PLN) frequentemente envolvem vetores de características esparsas e de alta dimensão, que separam efetivamente exemplos positivos e negativos, tornando Máquinas de Vetores de Suporte particularmente adequadas para essas tarefas (KAESTNER, 2013). Métodos de *kernel* permitem que as SVMs lidem com separabilidade não linear e processem estruturas linguísticas complexas como *strings*, árvores e gráficos, sem representação explícita de vetores de características (COLLINS; DUFFY, 2001; MOSCHITTI, 2012). Esses *kernels* podem capturar representações ricas e de alta dimensão de estruturas linguísticas, facilitando tarefas como análise sintática

(COLLINS; DUFFY, 2001).

3.2.9 SVC COM *KERNEL* LINEAR

O SVC (*Support Vector Classifier*) com *kernel* linear é um algoritmo de classificação de suporte vetorial que utiliza uma função de *kernel* linear para o mapeamento dos dados de entrada em um espaço de alta dimensionalidade (TAKAHASHI, 2015). Neste espaço, é possível encontrar um hiperplano que separa os dados de duas classes com o mínimo de margem de separação, que se trata da distância entre os pontos de cada classe. Devido ao *kernel* em questão, o hiperplano utilizado na classificação deve estar na condição linear.

A função *kernel* linear é definida como:

$$k(a, b) = a^T b. \tag{3.1}$$

Pesquisas recentes exploraram a combinação de *embeddings* BERT com vários classificadores para tarefas de classificação de texto. (CORDEIRO; RABELO; MOURA, 2022) compararam algoritmos clássicos como SVM com BERT para classificar comunicações irregulares, descobrindo que BERT obteve o melhor desempenho com 96% de pontuação F1. (ILIC; GARCÍA-MARTÍNEZ; PASTOR, 2022) conduziram uma revisão abrangente de modelos de classificação de texto, comparando abordagens tradicionais como SVM e *Random Forest* com modelos de última geração usando *embeddings* BERT e GPT-2. Seus resultados mostraram que os modelos BERT e GPT-2 tiveram melhor desempenho, com BERT superando ligeiramente o GPT-2 em tarefas de classificação binária. Para problemas multiclasse, o *C-Support Vector Classifier* e o BERT exibiram o melhor desempenho, destacando a eficácia das abordagens baseadas em BERT em vários cenários de classificação de texto.

Para ilustrar a aplicação do SVC com *kernel* linear, foi construído um código em linguagem Python, utilizando o conjunto de dado *iris*, disponível na biblioteca *scikit-learn*. Após a aplicação do algoritmo, foram plotados os resultados para a visualização das fronteiras de decisão. Nas Figuras 11 e 12, é possível visualizar as fronteiras de decisão no conjunto de treino e teste, respectivamente

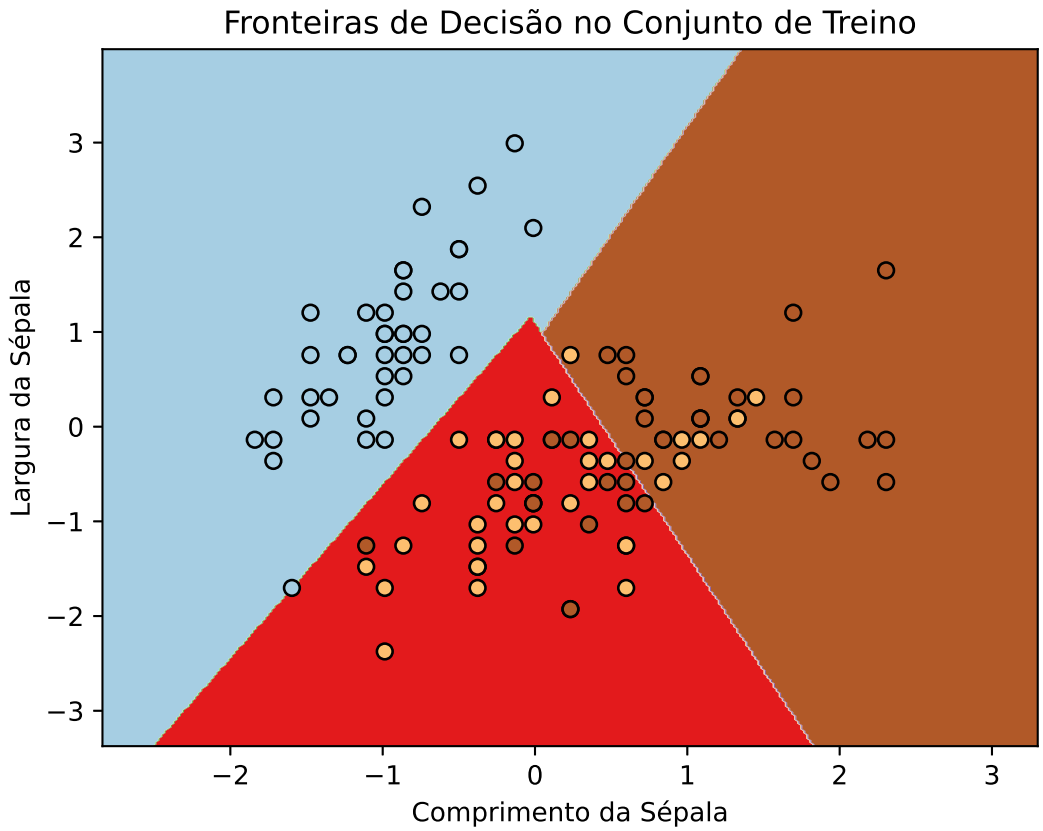


Figura 11 – Fronteiras de decisão no conjunto de treino.

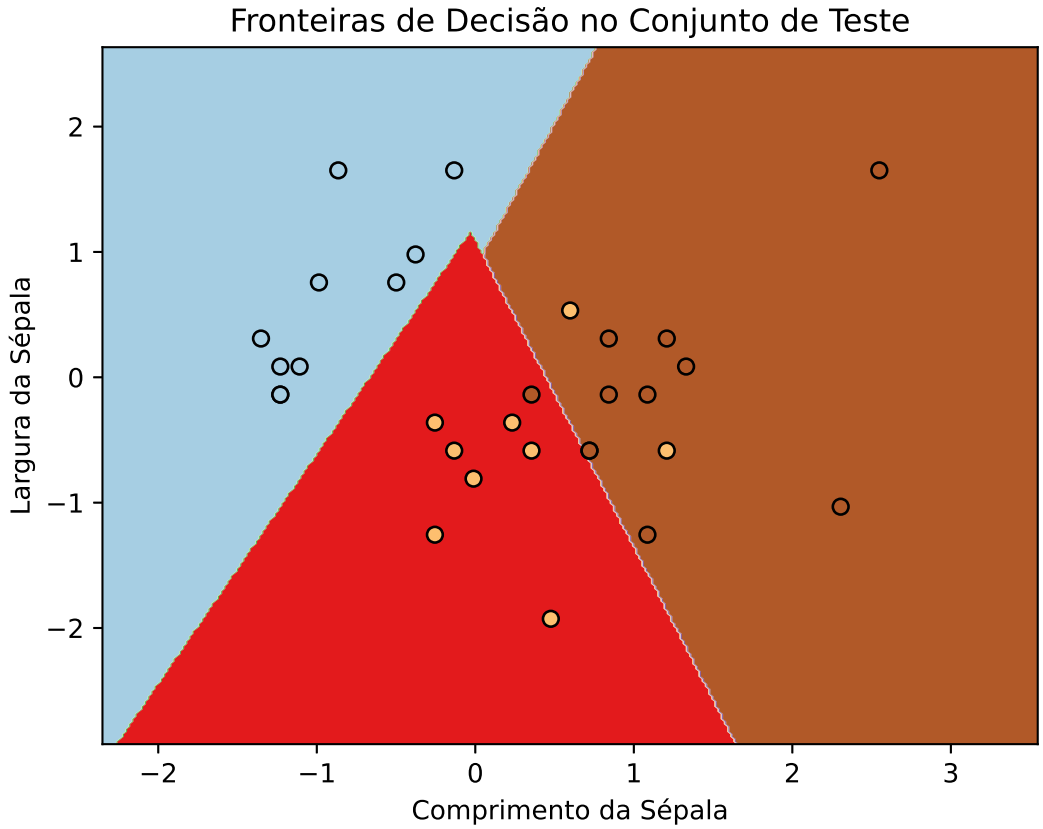


Figura 12 – Fronteiras de decisão no conjunto de teste.

Na Figura 12, é possível visualizar a separação dos dados em diferentes classes. As cores de cada ponto caracterizam suas respectivas classes verdadeiras.

3.2.10 MÉTODO *T-SNE*

O método *t-SNE* (*t-Distributed Stochastic Neighbor Embedding*) é uma técnica para visualizar dados de alta dimensionalidade, atribuindo a cada ponto de dados uma localização em um mapa bidimensional ou tridimensional. É uma variação da técnica *Stochastic Neighbor Embedding*, que é mais fácil de otimizar e produz visualizações melhores, reduzindo a aglomeração de pontos no centro do mapa. O *t-SNE* destaca-se por revelar estruturas em diferentes escalas e é especialmente útil para dados de alta dimensionalidade que residem em múltiplas, mas relacionadas, subdimensões. Além disso, a técnica pode usar caminhadas aleatórias em grafos de vizinhança para influenciar a exibição de subconjuntos de dados, podendo se tornar superior a outras técnicas de visualização não paramétricas, como *Sammon mapping*, *Isomap* e *Locally Linear Embedding* (MAATEN; HINTON, 2008).

4 MATERIAIS E MÉTODOS

Neste capítulo, será descrita a metodologia adotada para a confecção deste trabalho, bem como os recursos utilizados para o seu desenvolvimento.

4.1 MATERIAIS

O método proposto nesta pesquisa foi aplicado aos registros históricos de alarmes de uma Usina Termoelétrica situada no estado da Paraíba. Trata-se de uma empresa produtora independente de energia, com um total de 342 MW de capacidade instalada.

Foram extraídos registros dos anos de 2019 a 2022 para desenvolver este trabalho, totalizando 18.427.862 alarmes e eventos.

A usina conta com 40 unidades geradoras a Diesel (UGD'S). Cada unidade geradora é composta pela máquina primária (motor a combustão DIESEL/HFO) e por um gerador da fabricante Hyundai. 38 UGD's têm motores MAN STX 18V 32/40, e duas UGD's possuem motores MAN STX 9L 32/40. Na Figura 13, é ilustrado um motor com a configuração 18V 32/40.

Figura 13 – Motor MAN STX 18V 32/40. Retirado de: (HEEYAS. . . ,).

4.2 MÉTODOS

4.2.1 DIAGRAMA DE BLOCOS DO PROCEDIMENTO PROPOSTO

Esta pesquisa será dividida em duas etapas. Na primeira, as atividades ficaram concentradas na extração dos alarmes provenientes dos processos industriais da usina termoelétrica que consiste no objeto de estudo, pré-processamento dos alarmes utilizando técnicas de NLP e agrupamento destes em categorias semelhantes, através da aplicação de algoritmos de clusterização. Na Figura 14, é possível visualizar resumidamente as etapas da metodologia adotada no primeiro estágio deste trabalho.

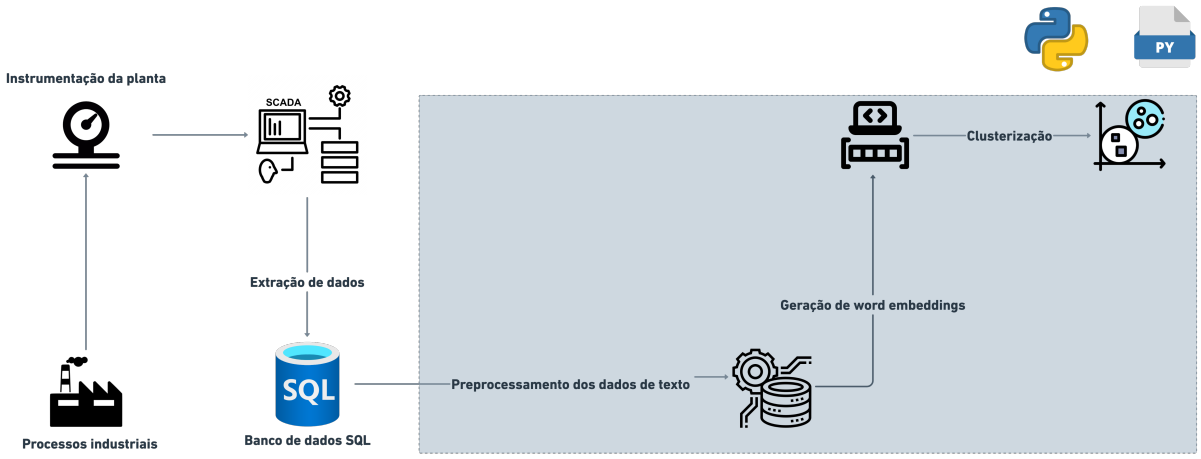


Figura 14 – Metodologia - Primeira etapa da pesquisa.

Inicialmente, a amostra de dados utilizada totalizou 40.000 registros de alarmes, distribuídos aleatoriamente em um período de 4 anos. Isso ocorreu principalmente devido às limitações de hardware da máquina utilizada nas primeiras execuções e iterações envolvendo a geração dos *embeddings* e aplicação dos algoritmos de clusterização.

Na segunda fase da pesquisa, inicialmente também foi utilizada a amostra reduzida, mencionada anteriormente. Com base nos *clusters* encontrados na etapa de clusterização, foi implementado um modelo de classificação de texto utilizando os dados rotulados. O conjunto dos dados a serem utilizados no modelo foi dividido em treino e teste. Após isso, foi avaliado o desempenho do modelo e foram rotulados 40.000 novos registros. A Figura 15 sintetiza os procedimentos realizados nesta fase do trabalho.

Figura 15 – Metodologia - Segunda etapa da pesquisa.

Fundamentadas as principais etapas necessárias que compõem as tarefas de pré-processamento dos alarmes e rotulagem, de posse dos resultados, propõe-se a implementação dos métodos aqui discutidos em uma aplicação que interaja diretamente com sistemas supervisórios, para permitir a filtragem de alarmes pertencentes a *clusters* específicos. Dessa forma, seria possível criar uma interface separada para que os operadores consigam visualizar apenas os grupos mais críticos ou subsistemas específicos.

4.2.2 EXTRAÇÃO DOS DADOS

Para o desenvolvimento deste trabalho, foi extraída uma amostra de 18427862 registros de alarmes e eventos presentes no sistema SCADA da planta termoelétrica. Os dados foram coletados para os anos de 2019 a 2022, em formato tabular. Os alarmes possuem 3 níveis de severidade:

- Severidade 0: Alarmes que indicam shutdown dos motores da usina;
- Severidade 1: Alarmes de processo;
- Severidade 2: Eventos;

Em geral, os alarmes da usina possuem um padrão de formato de escrita, composto pela área a qual o registro se refere, acompanhado de uma descrição, que pode indicar um *status*, ou atingimento de um *setpoint* pré-configurado para o alarme, etc. Por exemplo, o identificador *BCP* (*Boiler Control Panel*), identifica os alarmes do sistema de condensado da planta. O fluxograma da Figura 16 apresenta o padrão de escrita dos alarmes utilizando como base um registro que indica o nível limite de água de um dos tanques de condensado da planta, identificado pela *tag 2TO24*. O texto por extenso do alarme em questão é **BCP 2: D0071 LSHHH2T024 Feed Water Level Alarm High High High 2T024 (=0)**.

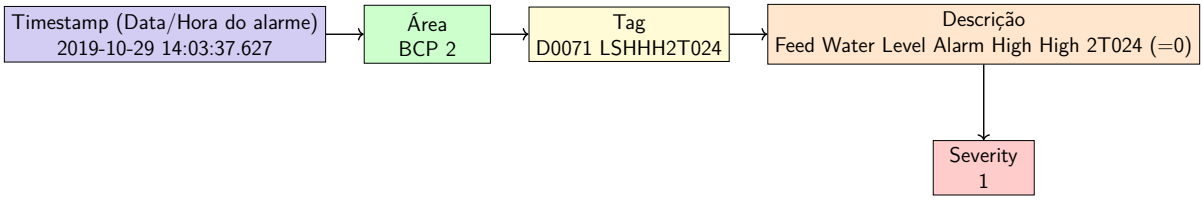


Figura 16 – Padrão de escrita dos alarmes

Na Tabela 1, tem-se uma amostra com os dados extraídos do sistema SCADA.

Tabela 1 – Formato do conjunto de dados.

Timestamp	Message	Severity
2019-10-29 13:48:18.650	CCP 02: Pump 10P003 Unselected mode	1
2019-10-29 13:57:55.617	CCP 2: D1451 Inactive	2
2019-10-29 14:19:44.840	GCP 29: L0210 2PZL7180 Emergency Stop Air Pressure Low Shut Down	0

Os textos dos alarmes, severidade, limites de valores e demais configurações são feitas no supervisório. Quando se tem como referência uma variável analógica, por exemplo,

o nível de água nas caldeiras, é possível monitorar até quatro níveis de alarme: muito baixo, baixo, alto e muito alto. Também é possível monitorar uma variável pela especificação de múltiplas subcondições, através de alarmes discretos, monitorar variáveis digitais pela especificação de alarme em borda de subida ou na borda de descida, dentre outras possibilidades.

Outro aspecto sobre os alarmes da usina, objeto desta pesquisa, é que devido a existência de motores iguais, bem como de sensores, por exemplo, a diferenciação das mensagens dos alarmes ocorre principalmente pelo uso de *tags* específicas que identificam os equipamentos da planta. Isto é exemplificado na Tabela 2, que mostra alarmes referentes a dois motores diferentes: o primeiro registro remete ao motor de número 1, e o segundo ao de número 03.

Tabela 2 – Alarmes com descrições análogas.

Timestamp	Message	Severity
2022-08-04 09:29:14.937	GCP 01: D0167 Cylinder Lubrication 1EM2470 Switching Failure	1
2022-08-04 09:29:26.750	GCP 03: D0167 Cylinder Lubrication 1EM2470 Switching Failure	1

4.2.3 PRÉ-PROCESSAMENTO

Na primeira etapa do trabalho foram pré-processados os textos dos alarmes da amostra utilizada. O pré-processamento é uma etapa fundamental em tarefas de NLP e na análise de dados, visando melhorar a qualidade e a usabilidade de dados textuais para análise ou modelagem subsequente (BRITO; GOMES, 2019). As etapas executadas foram as seguintes (BIRD; KLEIN; LOPER, 2009):

- Conversão dos caracteres para minúsculo, buscando a uniformização dos dados;
- Remoção de caracteres indesejados, como pontuação, símbolos, números e caracteres especiais. Dessa forma, é evitada a introdução de ruído no modelo de NLP, como também simplifica o processamento e torna o modelo mais generalizável;
- Remoção de *Stopwords*: eliminação de palavras muito comuns e com pouco significado, como artigos, preposições e pronomes. Tais palavras não contribuem significativamente para a classificação dos alarmes, e podem ser removidas para melhorar a eficiência do processamento;
- Lematização: redução das palavras às suas raízes correspondentes, retirando todas as inflexões;

- Remoção de palavras não pertencentes à língua inglesa: esta etapa foi utilizada para a remoção de *tags* de equipamentos e sensores, que permaneceram após as etapas de pré-processamento anteriores;

A Figura 17 detalha as etapas de pré-processamento realizadas em uma das mensagens dos alarmes presentes no conjunto de dados utilizado para o desenvolvimento deste trabalho.

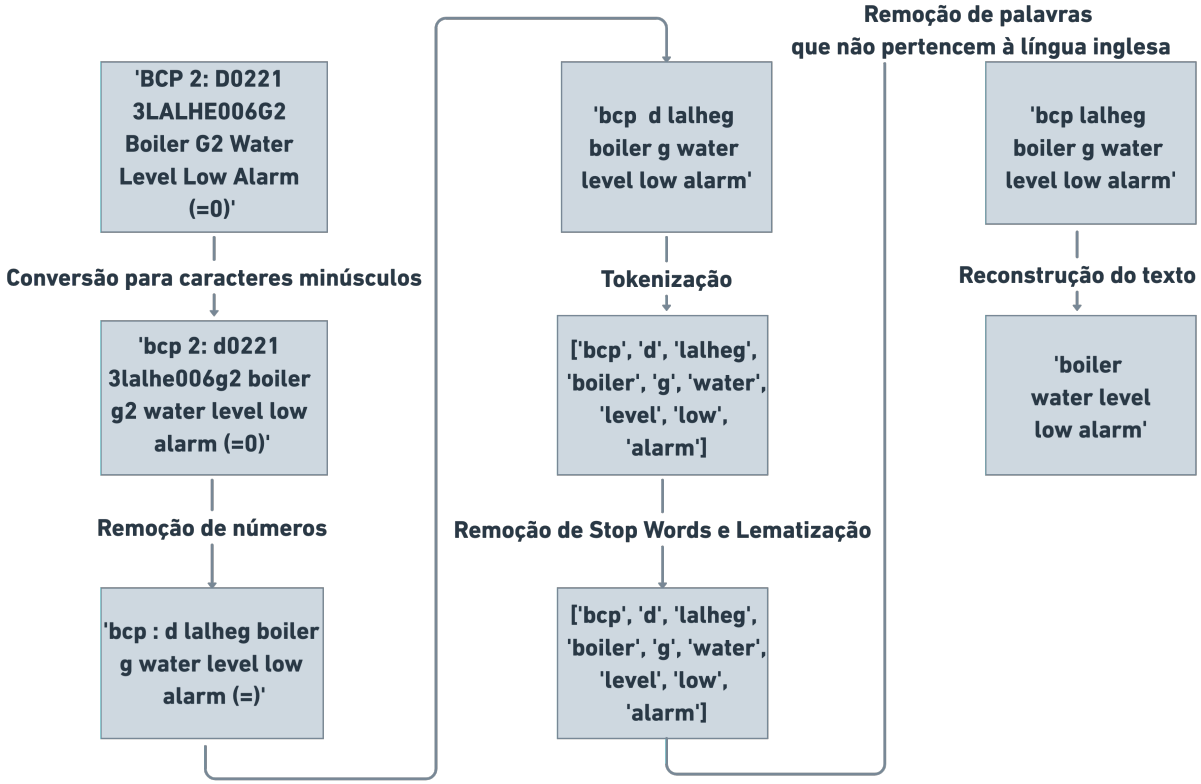


Figura 17 – Etapas de pré-processamento das mensagens dos alarmes.

4.2.4 *WORD EMBEDDINGS*

Após todas as etapas do tratamento dos dados de texto, as mensagens alarmes foram transformadas em embeddings utilizando o modelo pré-treinado *bert-base-nli-mean-tokens* e a biblioteca *sentence-transformers*. Este modelo é utilizado na transformação de sentenças, destinado ao mapeamento de frases e parágrafos para um espaço vetorial denso de 768 dimensões, que pode ser utilizado em aplicações envolvendo agrupamento ou busca semântica. Na Tabela 3 é possível visualizar uma mensagem do conjunto de dados de alarmes, após a etapa de pré-processamento, e os dez primeiros elementos dos *embeddings* correspondentes.

Tabela 3 – Mensagem pré-processada de alarme e *Embeddings* correspondentes.

Sentença	**boiler water level low alarm**
Embeddings	**[0.06018863 0.39419553 0.49162325 0.5049072 0.32218906 0.17818248 1.5778452 0.09567354 0.13199821 -0.3159805 ...]**

4.3 RECURSOS E MÓDULOS PYTHON UTILIZADOS

Para a construção da análise dos dados dos alarmes utilizando técnicas de Processamento de Linguagem Natural (NLP) e algoritmos de clusterização, foram empregados diversos recursos e módulos da linguagem de programação Python. Deste grupo, destacam-se os seguintes:

1. Biblioteca NLTK (*Natural Language Toolkit*): O NLTK é uma das principais bibliotecas para Processamento de Linguagem Natural em Python. Esta oferece uma ampla gama de recursos, como tokenização, remoção de *stopwords*, lematização, *stemming*, entre outros. Através do NLTK, foram realizadas etapas fundamentais de pré-processamento dos textos de alarmes, de forma a deixá-los adequados para a aplicação dos algoritmos de aprendizado de máquina.

2. Biblioteca *scikit-learn*: A biblioteca *scikit-learn* é uma das mais utilizadas em aprendizado de máquina em Python. Esta disponibiliza uma grande variedade de algoritmos de classificação, *clustering*, regressão e pré-processamento de dados. Além disso, foi utilizada a função de vetorização TF-IDF para transformar os textos em vetores numéricos.

3. Biblioteca Pandas: O Pandas é uma biblioteca amplamente utilizada para análise de dados em Python. Neste trabalho, o Pandas foi empregado para a manipulação e exploração dos dados durante o processo de análise.

4. Bibliotecas para visualização: Também foram empregadas bibliotecas utilizadas para visualização de dados, como *Matplotlib* e *Seaborn*.

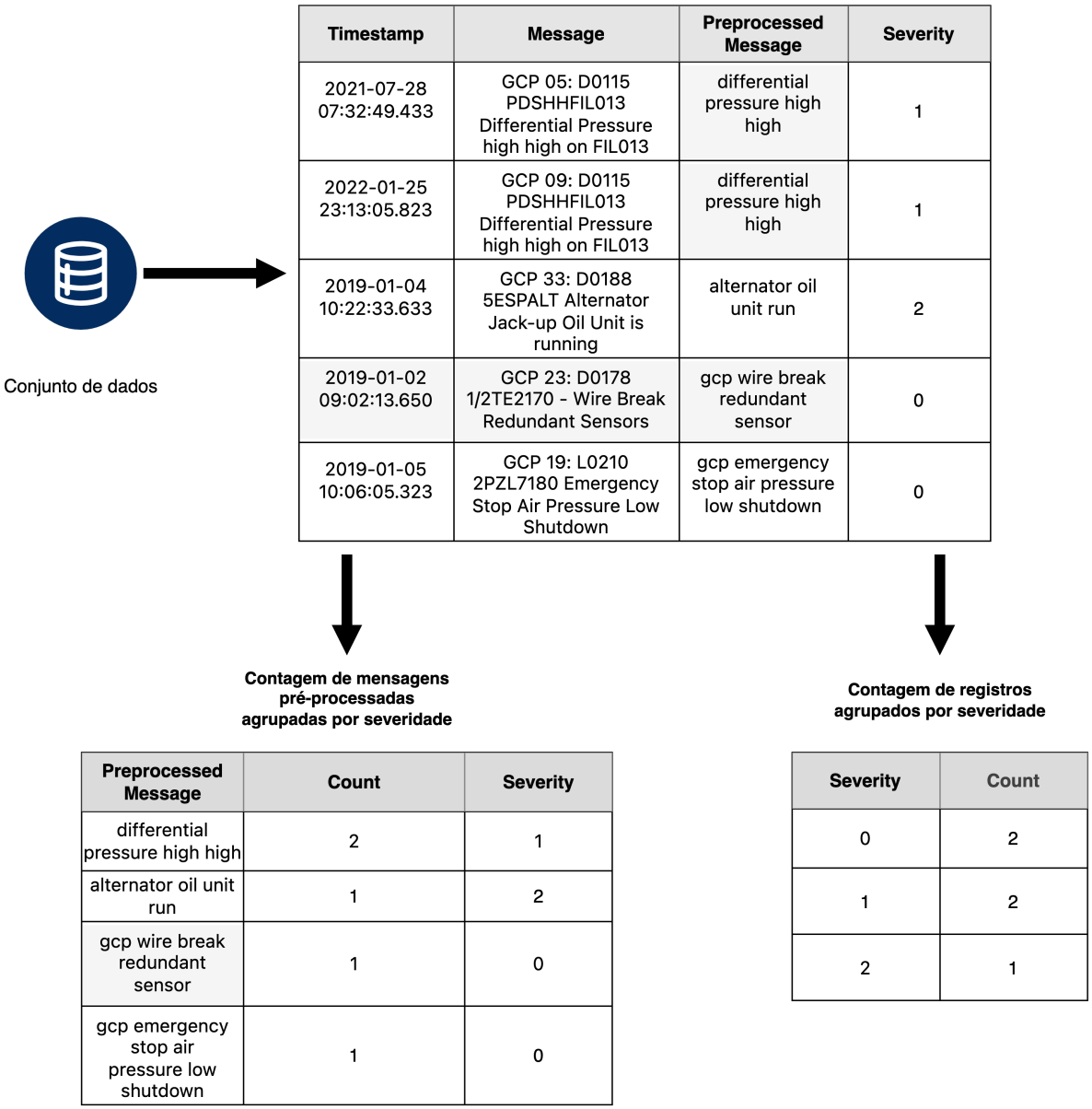
5. Biblioteca *numpy*: O *numpy* é uma biblioteca essencial para computação numérica em Python. O *numpy* foi utilizado em várias etapas da análise.

4.3.1 ANÁLISE EXPLORATÓRIA

Após as etapas de pré-processamento de texto, foi feita uma análise exploratória para extrair dados sumarizados, e entender a distribuição e volumetria dos registros de alarmes e eventos, em relação às informações disponíveis no conjunto de dados.

A análise do número de registros de alarmes por dia, por hora, ou minutos, é de grande valia para mensurar o impacto que os registros de alarmes causam aos operadores.

Além disso, analisar a volumetria por severidade também é uma tarefa importante, uma vez que a priorização dos registros dos alarmes também pode utilizar a severidade como filtro. Para uma métrica preliminar, uma vez que o pré-processamento remove *tags* e outros identificadores no texto, a contagem de registros agrupados pelas mensagens pré-processadas pode fornecer uma estimativa dos alarmes que mais impactam. A Figura 18 ilustra os procedimentos para sumarização dos dados.



Timestamp	Message	Preprocessed Message	Severity
2021-07-28 07:32:49.433	GCP 05: D0115 PDSHHFIL013 Differential Pressure high high on FIL013	differential pressure high high	1
2022-01-25 23:13:05.823	GCP 09: D0115 PDSHHFIL013 Differential Pressure high high on FIL013	differential pressure high high	1
2019-01-04 10:22:33.633	GCP 33: D0188 5ESPALT Alternator Jack-up Oil Unit is running	alternator oil unit run	2
2019-01-02 09:02:13.650	GCP 23: D0178 1/2TE2170 - Wire Break Redundant Sensors	gcp wire break redundant sensor	0
2019-01-05 10:06:05.323	GCP 19: L0210 2PZL7180 Emergency Stop Air Pressure Low Shutdown	gcp emergency stop air pressure low shutdown	0

Preprocessed Message	Count	Severity
differential pressure high high	2	1
alternator oil unit run	1	2
gcp wire break redundant sensor	1	0
gcp emergency stop air pressure low shutdown	1	0

Severity	Count
0	2
1	2
2	1

Figura 18 – Sumarização dos dados dos alarmes.

Além disso, outra análise proposta, que motiva principalmente a priorização e filtragem dos registros, foi avaliar o impacto da volumetria dos alarmes e eventos que se repetem em um curto intervalo de tempo, causando sobrecargas para os operadores e

prejudicando a abordagem imediata dos alarmes mais críticos. Para tanto, foi implementado um código para verificar registros de conteúdos idênticos gerados sequencialmente, em um intervalo de tempo menor ou igual a dez minutos. Na tabela 4, são mostrados registros do conjunto de dados que se repetiram em um curto espaço de tempo.

Tabela 4 – Alarmes repetidos dentro do intervalo de 5 minutos.

Message	**InTime**	**FLAG_REP_10_MIN**
CCP 1: D0701 1C003 Working Air Compressor Running	2022-08-04 10:15:44.753	0
CCP 1: D0701 1C003 Working Air Compressor Running	2022-08-04 10:15:44.753	1
CCP 1: D0510 5ES1P072 Drain Pump LO Sludge Tank Sludge Module 1P072 On	2022-08-04 10:17:10.943	0
CCP 1: D0510 5ES1P072 Drain Pump LO Sludge Tank Sludge Module 1P072 On	2022-08-04 10:17:10.943	1
GCP 29: Wirebreak 0017 5TE6570B Exhaust Gas Temperature Cylinder 5B	2022-08-04 13:19:53.833	0
GCP 29: Wirebreak 0017 5TE6570B Exhaust Gas Temperature Cylinder 5B	2022-08-04 13:29:49.777	1

As linhas que hachuradas em amarelo, cujos valores da coluna **FLAG_REP_10_MIN** são iguais a 1, correspondem às repetições de registros de eventos e alarmes dentro de um intervalo de dez minutos.

4.3.2 CLUSTERIZAÇÃO DO CONTEÚDO DOS ALARMES

Para a tarefa de clusterização das mensagens pré-processadas dos alarmes, o primeiro passo foi a geração dos *embeddings*, utilizando o modelo pré-treinado **bert-base-nli-mean-tokens**. Após isso, foi definido um valor máximo de *clusters* (k), a fim de iterar até este limite e determinar o valor ideal. Implementou-se uma função para executar o algoritmo *k-means*, e determinar o número ótimo de *clusters*. Para isso, foram utilizados dois métodos distintos: método do cotovelo e *silhouette score*. Para cada um dos métodos, foi escolhido o valor máximo de k, resultante das iterações realizadas. Com os respectivos números de *clusters* encontrados, o *k-means* foi ajustado, adicionando os rótulos de cada

cluster aos registros do conjunto de dados. A Figura 19, resume as etapas realizadas no processo de clusterização.

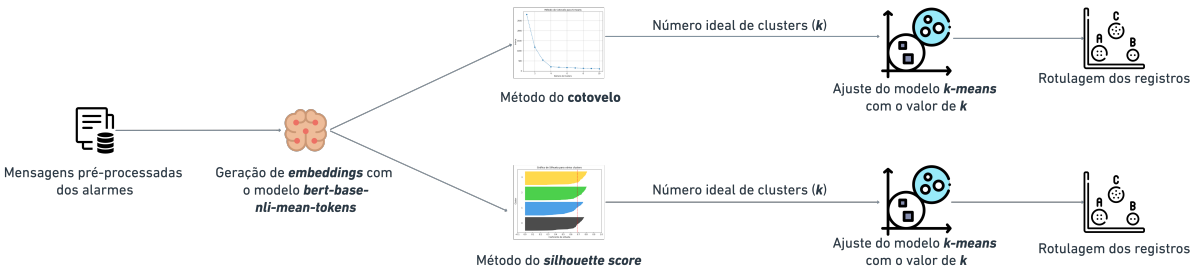


Figura 19 – Esquemático do processo de clusterização.

Para trabalhar com um maior volume de dados, foi explorado o uso da biblioteca *Faiss*, que é utilizada na busca de similaridade e clusterização envolvendo vetores densos e de forma eficiente. Esta biblioteca contém algoritmos que realizam buscas em vetores de qualquer tamanho, incluindo os que possivelmente não se encaixam na memória RAM disponível. *Faiss* é escrita em **C++**, e alguns dos algoritmos mais simples são implementados na GPU. Seu desenvolvimento se deu primordialmente na *FAIR*, a equipe de pesquisa fundamental em Inteligência Artificial da empresa **Meta**.

4.3.3 ETAPA DE CLASSIFICAÇÃO

Explorou-se também a classificação de novas amostras, utilizando os registros rotulados na etapa de clusterização. Dessa forma, a ideia central foi a transformação do aprendizado não supervisionado em supervisionado. Após a remoção dos dados nulos, e utilizando as mensagens dos alarmes já pré-processadas e seus *clusters* correspondentes, os dados foram divididos em subconjuntos de treino e teste, 80% e 20%, respectivamente.

Na classificação, foi utilizada a classe ***LinearSVC***, pertencente ao algoritmo **SVC** (*Support Vector Classification*), utilizando um *kernel* linear. Trata-se de uma variante do SVM (*Support Vector Machine*), e apresenta maior eficiência que o SVC padrão com *kernel* linear quando há um grande volume de registros, pois utiliza a biblioteca *liblinear*, que é otimizada para amostras maiores.

Na Figura 20, tem-se as etapas realizadas no processo de classificação de registros não rotulados.

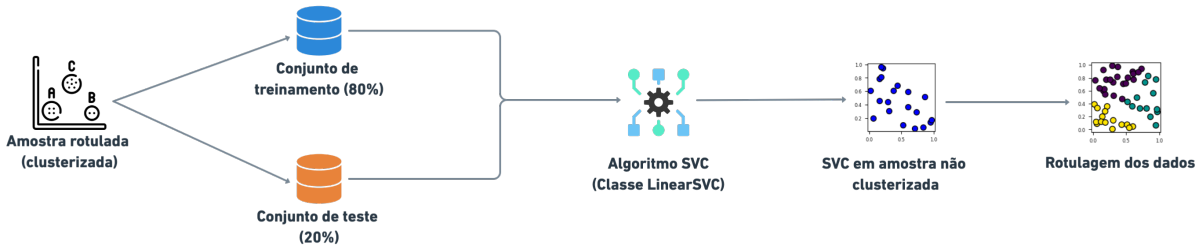


Figura 20 – Esquemático do processo de classificação dos dados.

4.3.4 ANÁLISE DE SEQUÊNCIAS TEMPORAIS

Outro assunto explorado foi a análise de sequências temporais nos alarmes, a fim de detectar padrões existentes nos dados. Com isso, torna-se possível identificar causas raiz para o surgimento de determinados registros, bem como prever futuras falhas no processo. Para tanto, implementou-se um código para detectar transições entre os os *clusters*, e contabilizar as mais representativas. Utilizando inicialmente a amostra clusterizada, ordenou-se os dados por data e hora dos registros, verificando o registro atual e, após este, todos os alarmes que surgiram em até dez minutos e seus *clusters* correspondentes. Contabilizou-se as transições entre *clusters*, e foram observadas as dez mais representativas em termos de volumetria. Na Figura 21, é possível visualizar os passos realizados para análise das transições de registros entre *clusters*.

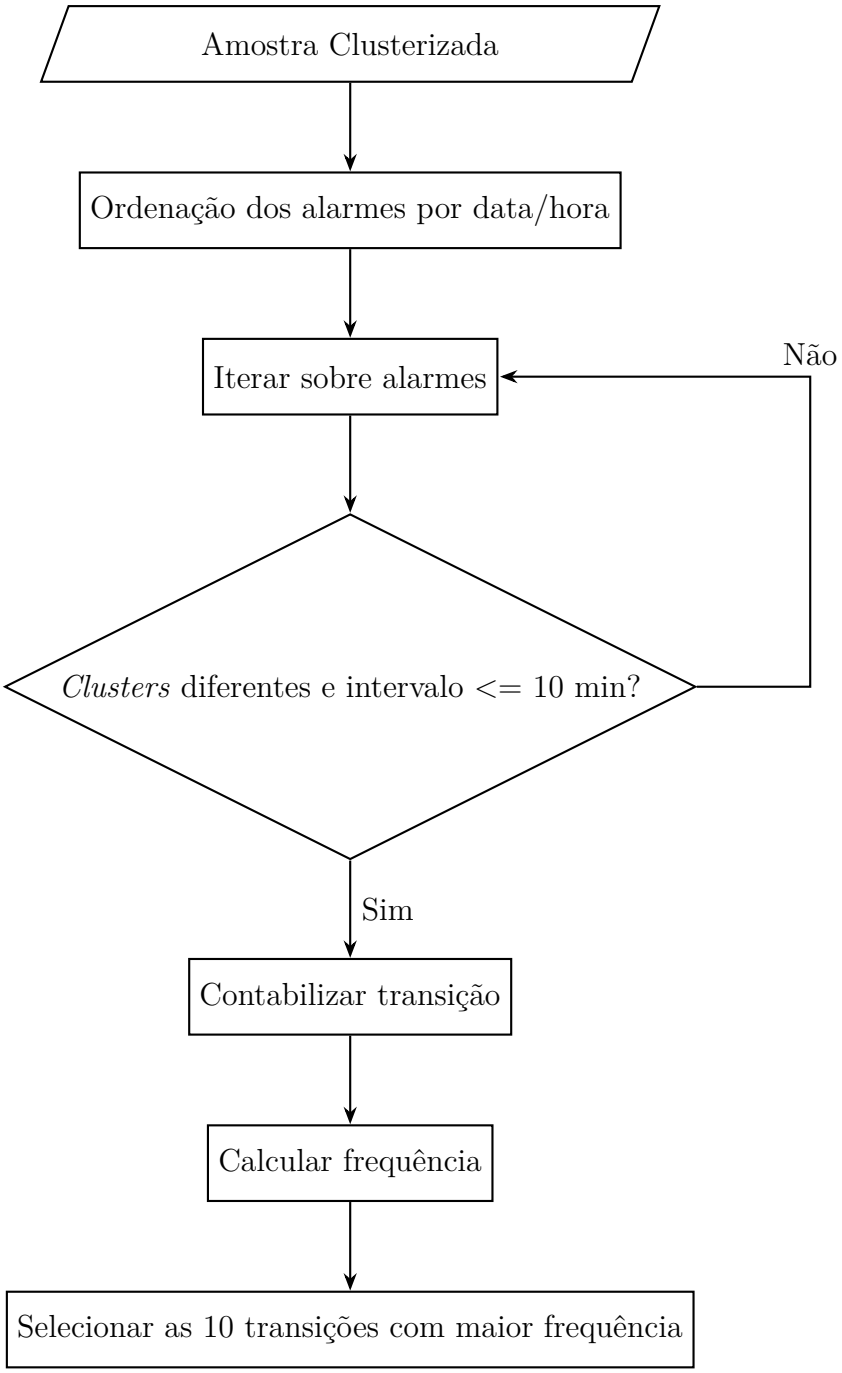


Figura 21 – Fluxograma para análise de transições de *clusters*.

Observando a frequência de sequências temporais envolvendo alarmes pertencentes a diferentes *clusters*, e que ocorrem em um curto intervalo de tempo, é possível filtrar os registros consecutivos que se referem aos mesmos motores da planta, por exemplo. Dessa maneira, se os padrões de determinadas sequências de alarmes se repetem para vários motores, é possível inferir uma correlação existente entre diferentes equipamentos que, apesar de pertencerem às mesmas máquinas e estarem em diferentes *clusters*, podem sugerir causas raiz de falhas para os mais diversos ativos da planta e problemas operacionais.

A Figura 22 exemplifica um cenário existente nos dados de alarmes da usina, em que o padrão sequencial de registros se repete para vários motores. Ainda na Figura 22, os dados ***cluster* atual** são os alarmes que antecedem imediatamente os registros das linhas correspondentes na tabela do ***cluster* seguinte**.

***Cluster* Atual**

Data/Hora - Cluster Atual	Cluster Atual	Mensagem - Cluster Atual	Área - Cluster Atual
2019-01-04 05:00:12.093	79	GCP 22: Pump PALT	TNE.ManualMode.UG22
2019-01-19 20:30:01.807	79	GCP 18: Pump PALT	TPB.ManualMode.UG18
2019-01-29 04:10:12.433	79	GCP 03: Pump PALT	TPB.ManualMode.UG03
2019-02-04 05:00:18.383	79	GCP 07: Pump PALT	TPB.ManualMode.UG07
2019-02-09 12:23:11.923	79	GCP 31: Pump PALT	TNE.ManualMode.UG31
2019-02-10 19:58:04.797	79	GCP 11: Pump PALT	TPB.ManualMode.UG11
2019-02-10 20:12:06.827	79	GCP 19: Pump PALT	TPB.ManualMode.UG19
2019-03-15 10:30:35.917	79	GCP 21: Pump PALT	TNE.ManualMode.UG21
2019-04-16 03:52:22.933	79	GCP 12: Pump PALT	TPB.ManualMode.UG12
2019-05-27 04:26:09.350	79	GCP 18: Pump PALT	TPB.ManualMode.UG18
2019-05-27 09:58:19.433	79	GCP 10: Pump PALT	TPB.ManualMode.UG10
2019-05-27 11:22:55.253	79	GCP 30: Pump PALT	TNE.ManualMode.UG30
2019-05-30 04:27:47.077	79	GCP 03: Pump PALT	TPB.ManualMode.UG03

***Cluster* seguinte**

Data/Hora - Cluster Seguinte	Cluster Seguinte	Mensagem - Cluster Seguinte	Área - Cluster Seguinte
2019-01-04 05:00:24.297	133	GCP 22: D0167 Cylinder Lubrication 1EM2470 Switching Failure	TNE.ManualMode.UG22
2019-01-19 20:30:31.257	133	GCP 15: D0167 Cylinder Lubrication 1EM2470 Switching Failure	TPB.ManualMode.UG18
2019-01-29 04:11:06.670	133	GCP 01: D0167 Cylinder Lubrication 1EM2470 Switching Failure	TPB.ManualMode.UG03
2019-02-04 05:14:39.717	133	GCP 09: D0167 Cylinder Lubrication 1EM2470 Switching Failure	TPB.ManualMode.UG07
2019-02-09 12:23:52.703	133	GCP 31: D0167 Cylinder Lubrication 1EM2470 Switching Failure	TNE.ManualMode.UG31
2019-02-10 19:58:46.417	133	GCP 11: D0167 Cylinder Lubrication 1EM2470 Switching Failure	TPB.ManualMode.UG11
2019-02-10 20:17:23.147	133	GCP 15: D0167 Cylinder Lubrication 1EM2470 Switching Failure	TPB.ManualMode.UG19
2019-03-15 10:30:43.123	133	GCP 21: D0167 Cylinder Lubrication 1EM2470 Switching Failure	TNE.ManualMode.UG21
2019-04-16 03:52:38.043	133	GCP 33: D0167 Cylinder Lubrication 1EM2470 Switching Failure	TPB.ManualMode.UG12
2019-05-27 04:27:17.763	133	GCP 32: D0167 Cylinder Lubrication 1EM2470 Switching Failure	TPB.ManualMode.UG18
2019-05-27 10:16:41.680	133	GCP 22: D0167 Cylinder Lubrication 1EM2470 Switching Failure	TPB.ManualMode.UG10
2019-05-27 11:23:20.843	133	GCP 30: D0167 Cylinder Lubrication 1EM2470 Switching Failure	TNE.ManualMode.UG30
2019-05-30 04:28:33.537	133	GCP 02: D0167 Cylinder Lubrication 1EM2470 Switching Failure	TPB.ManualMode.UG03

Figura 22 – Transições entre *clusters* e sequências temporais.

Na Figura 22, é possível verificar que o padrão de ocorrência sequencial entre os alarmes com descrição **GCP XX: Pump PALT** e **GCP XX: D0167 Cylinder Lubrication 1EM2470 Switching Failure**, pertencentes aos *clusters* **79** e **133**, respectivamente, se repete para diferentes motores. Expandindo esta análise para todo o

conjunto de dados disponível, é possível estabelecer relações de causas raiz para o surgimento de determinados alarmes e, dessa maneira, determinar as fontes para diversas falhas que ocorrem nos equipamentos da planta. Além disso, observar cenários análogos aos da Figura 22, envolvendo alarmes de diferentes subsistemas, também possibilita um melhor entendimento e diagnóstico de falhas mais complexas.

5 RESULTADOS E DISCUSSÕES

Após o pré-processamento dos dados dos alarmes, foi realizada uma análise exploratória para quantificar a distribuição dos registros conforme a severidade. A Figura 23 apresenta um gráfico de barras que ilustra o volume e a proporção de registros em cada categoria de severidade.

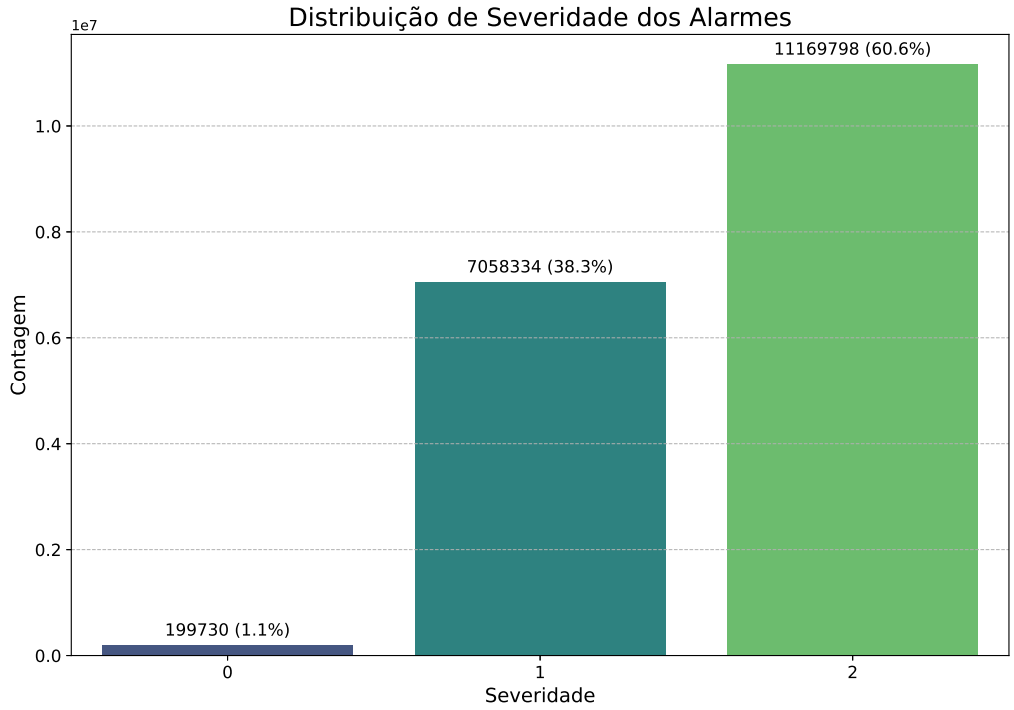


Figura 23 – Distribuição dos alarmes e eventos pela severidade.

Os registros com severidade **2** correspondem a eventos, enquanto os demais, com severidades **0** e **1**, são classificados como falhas.

Além disso, foi realizada uma análise para identificar quais alarmes possuem maior representatividade em cada nível de severidade. Para essa análise, foram utilizadas as mensagens pré-processadas, devido à generalização do texto no pós-processamento. A Figura 24 apresenta as mensagens dos alarmes mais frequentes com severidade 0.

Analisou-se também quais descrições de alarmes possuem maior representatividade, para cada severidade correspondente. Para tanto, devido à generalização do texto pós-processamento, utilizaram-se as mensagens pré-processadas. Na Figura 24, é possível

visualizar as mensagens dos alarmes mais frequentes com severidade 0.

Além disso, foi realizada uma análise para identificar quais descrições dos alarmes possuem maior representatividade em cada nível de severidade. Para isso, foram utilizadas as mensagens pré-processadas, devido à generalização do texto, com a ausência das *tags* dos equipamentos e outros elementos. A Figura 24 apresenta as descrições dos alarmes mais frequentes com **severidade 0**.

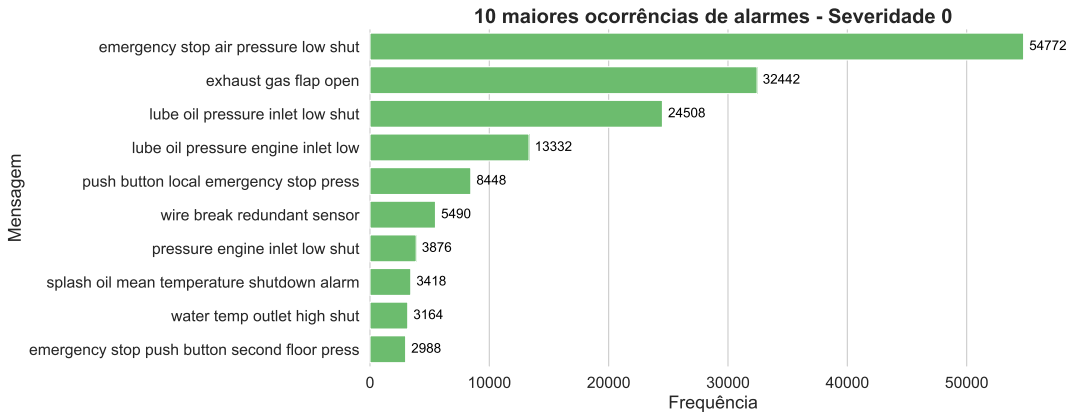


Figura 24 – Mensagens dos alarmes mais recorrentes - Severidade 0.

De forma similar, as Figuras 25 e 26 exibem os gráficos correspondentes às severidades **1** e **2**, respectivamente.

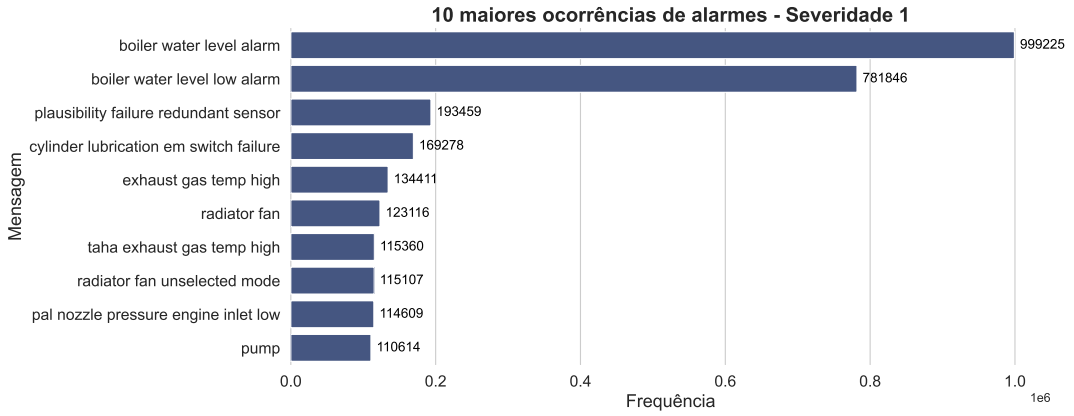


Figura 25 – Mensagens dos alarmes mais recorrentes - Severidade 1.

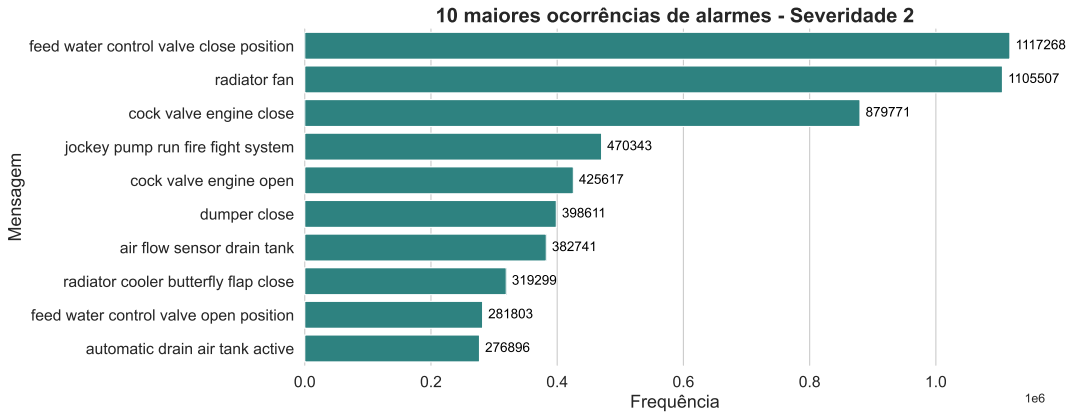


Figura 26 – Mensagens dos eventos mais recorrentes - Severidade 2.

Conforme mencionado na metodologia deste trabalho, foram analisadas repetições sequenciais de alarmes em intervalos curtos de tempo. Os registros com conteúdo idêntico, e que se repetem até dez minutos, representam **73,4%** de todo o conjunto de dados. O elevado percentual de repetições ressalta a importância da implementação de mecanismos para remover alarmes redundantes, bem como aplicar os métodos aqui propostos, desenvolvendo uma aplicação que execute em ambiente de produção, e interaja diretamente com o sistema supervisório da planta, permitindo filtrar ou priorizar os alarmes pertencentes a determinados *clusters*.

Além disso, contabilizou-se a média da quantidade de alarmes por dia, hora e minuto. Na Tabela 5 é possível visualizar estas informações.

Tabela 5 – Média de registros por severidade.

Severidade	Média Diária	Média Horária	Média por Minuto
0	161.724696	24.731303	8.201454
1	5359.403189	235.954202	11.858083
2	8461.968182	358.569484	13.365719

Os números acima são bastante úteis para se ter uma estimativa da quantidade de registros processados diariamente, no caso da implementação de uma aplicação em tempo real utilizando a metodologia aqui proposta. Desconsiderando os eventos para esses cálculos e analisando apenas os dados com severidade **0** e **1** (alarmes de *shutdown* dos motores e de processo, respectivamente), obteve-se uma média aproximada de 5.511 registros diários, 242 registros por hora e 12 alarmes por minuto.

Finalizada a análise exploratória inicial, constatou-se o impacto significativo da elevada volumetria de alarmes, o que representa um desafio considerável para a gestão eficiente desses registros pela área operacional.

Com o objetivo de agrupar de maneira inteligente os alarmes com similaridade entre si, para uma posterior priorização dos grupos mais críticos, foi executado o algoritmo de clusterização K-means. Inicialmente, utilizou-se uma amostra de 10.000 registros aleatórios por ano, abrangendo os anos de 2019 a 2022, totalizando 40.000 alarmes. O principal motivo da utilização de uma amostra mais reduzida, em comparação com a quantidade de dados disponíveis, foi a limitação computacional para lidar com maiores volumes de dados. Foram realizadas iterações para determinar o número ideal de *clusters* por meio dos métodos do cotovelo e coeficiente de *silhouette*.

Ao utilizar o método do cotovelo, observou-se um ponto onde a curva de soma dos quadrados das distâncias *intra-cluster* começou a nivelar, conforme Figura 27. Este ponto, conhecido como "cotovelo", sugere um equilíbrio potencial entre a complexidade do modelo e a representação dos dados. Nesse contexto, o método do cotovelo sugeriu um número ótimo de *clusters* (k) igual a 9.

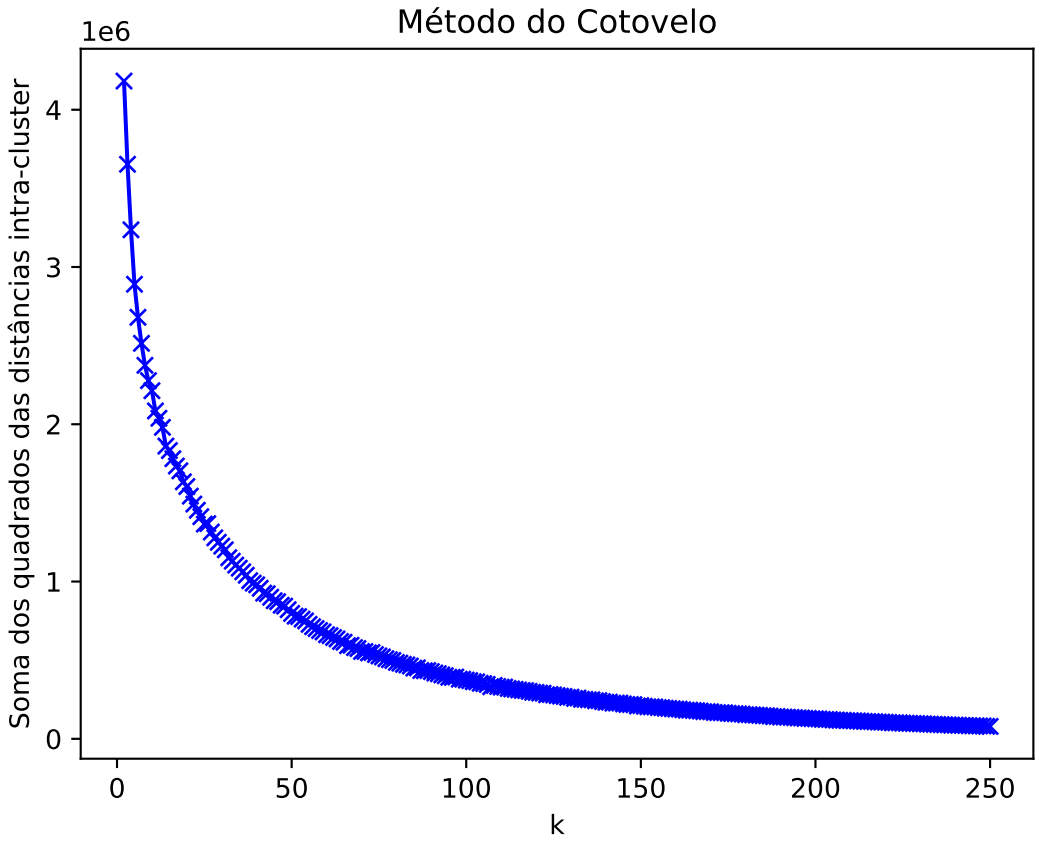


Figura 27 – Método do cotovelo - Algoritmo K-Means.

Utilizando o método do cotovelo e analisando os rótulos atribuídos aos registros dos alarmes, observou-se que algumas mensagens não correlatas foram agrupadas no mesmo *cluster*. A Tabela 6 ilustra a presença de mensagens com similaridades muito baixas ou inexistentes no *cluster* 0.

Tabela 6 – Formato do conjunto de dados.

Timestamp	Message	Cluster - Método do Cotovelo
2021-06-16 00:04:53.547	CCP 2: D2617 Dumper closed - BF6531-L2-V2	0
2020-02-18 11:43:04.413	Electrical: T13 LV C.B. Unselected mode	0
2019-10-29 14:19:45.417	GCP 35: D0076 1HS1014 Remote Operation Selected	0

Por outro lado, o método do coeficiente de *silhouette* forneceu uma perspectiva diferente. Ao avaliar os resultados para diversos números de *clusters*, constatou-se que o valor máximo foi alcançado com 249 *clusters*. Esse método apresentou um nível de distinção significativamente maior entre os grupos, com cada *cluster* contendo mensagens de alta similaridade. Na Figura 28, é possível visualizar um gráfico representando os valores de k (número de *clusters*) em função do coeficiente de *silhouette*:

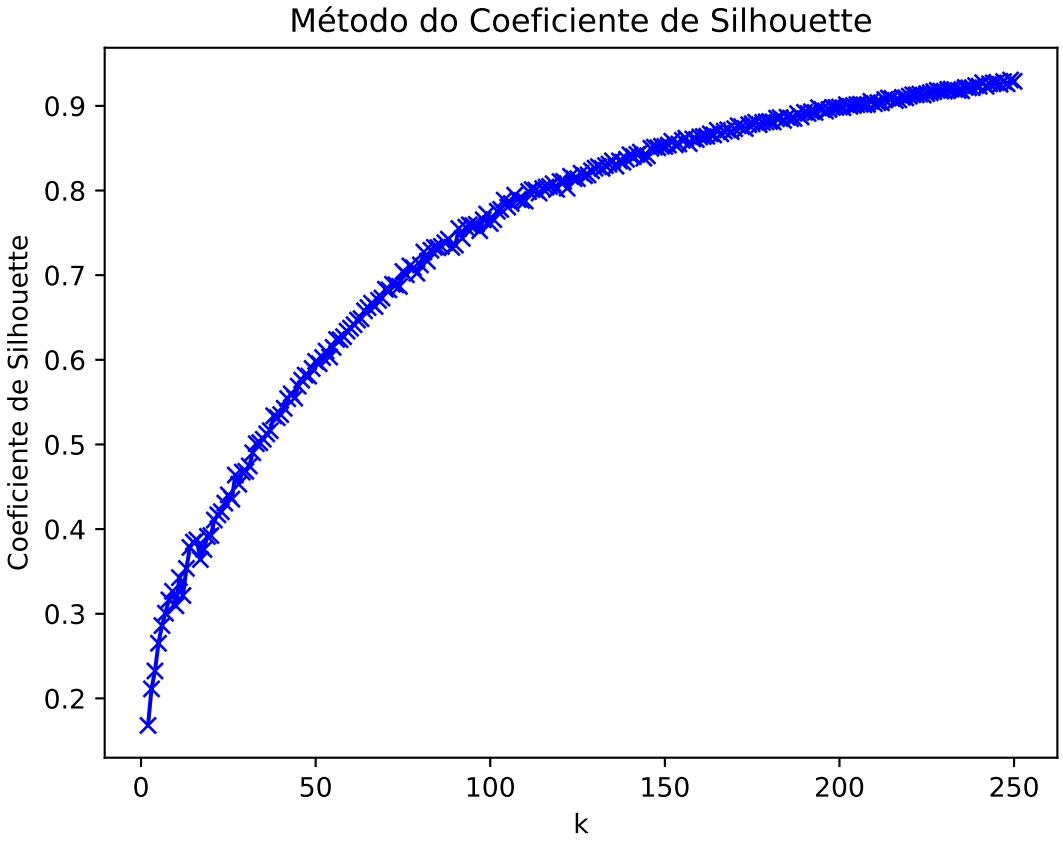


Figura 28 – Método do coeficiente de *silhouette* - Algoritmo K-Means.

Após o processo de clusterização, foram realizadas validações nos resultados, a fim de garantir que os alarmes foram agrupados corretamente. Por exemplo, no *cluster* com rótulo 4, foram agrupados registros relacionados ao sistema de condensado da planta, especificamente envolvendo as caldeiras. Na Tabela 7 é possível visualizar uma amostra das mensagens dos alarmes pertencentes ao ***cluster* 4**.

Tabela 7 – Cluster 4 (Sistema de condensado).

Message	**Cluster**
BCP 2: D0244 3LAHHHE006J2 Boiler J2 Water Level High/High Alarm (=0)	4
BCP 2: D0292 3LAHHHE006P2 Boiler P2 Water Level High/High Alarm (=0)	4
BCP 2: D0220 3LAHHHE006G2 Boiler G2 Water Level High/High Alarm (=0)	4
BCP 2: D0196 3LAHHHE006D2 Boiler D2 Water Level High/High Alarm (=0)	4

Os registros da Tabela 7 corresponderam a um total de **2713 alarmes**.

No *cluster* 8, por exemplo, foram agrupados os alarmes que remetem aos radiadores da planta, conforme Tabela 8.

Tabela 8 – Cluster 8 (Radiadores da planta).

Message	**Cluster**
CCP 02: Radiator Fan 44HE003A2E2 Unselected mode	8
CCP 02: Radiator Fan 39HE003L2Q2 Unselected mode	8
CCP 02: Radiator Fan 43HE003L2Q2 Unselected mode	8
CCP 02: Radiator Fan 5HE003L2Q2 Unselected mode	8
CCP 02: Radiator Fan 38HE003R2V2 Unselected mode	8
CCP 01: Radiator Fan 4HE003F1K1 Unselected mode	8

Os registros da Tabela 8 totalizaram 698 alarmes da amostra analisada.

Conforme é possível observar nas duas tabelas, a clusterização resultou em uma distribuição bem definida dos alarmes em cada grupo. Os textos dos alarmes de cada *cluster* possuem conteúdos semelhantes, e fazem referência aos mesmos subsistemas da planta.

Como é possível observar nas duas tabelas, a clusterização resultou em uma distribuição bem definida dos registros em cada grupo. Os textos dos alarmes de cada *cluster* apresentam conteúdos semelhantes e fazem referência aos mesmos subsistemas da planta.

Na Figura 29, é possível visualizar a distribuição espacial de pontos dos dez *clusters* com maior volumetria, por meio do método estatístico *t-SNE*.

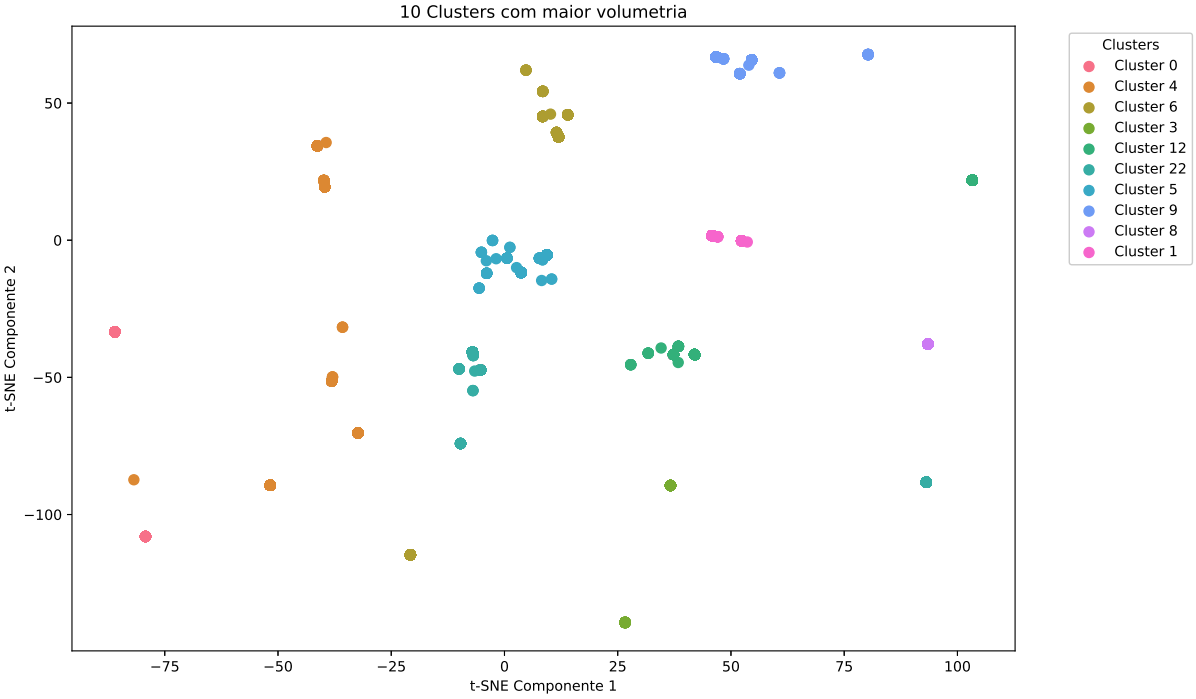


Figura 29 – Distribuição espacial dos 10 *clusters* com maior volumetria.

Ainda na Figura 29, observou-se que os *clusters* **0** e **4** apresentam pontos próximos. De fato, esses grupos contêm registros com mensagens similares, conforme mostrado na Tabela 9.

Tabela 9 – *Clusters* 0 e 4.

Message	**Cluster**
BCP 2: D0291 2LZLHE006P2 Boiler P2 Water Level Low (=0) Safety Supervision	0
BCP 2: D0173 3LALHE006B2 Boiler B2 Water Level Low Alarm (=0)	0
BCP 2: D0195 2LZLHE006D2 Boiler D2 Water Level Low (=0) Safety Supervision	0
BCP 2: D0341 3LALHE006U2 Boiler U2 Water Level Low Alarm (=0)	0
BCP 2: D0221 3LALHE006G2 Boiler G2 Water Level Low Alarm (=0)	0
BCP 2: D0197 3LALHE006D2 Boiler D2 Water Level Low Alarm (=0)	0
BCP 2: D0244 3LAHHHE006J2 Boiler J2 Water Level High/High Alarm (=0)	4
BCP 2: D0292 3LAHHHE006P2 Boiler P2 Water Level High/High Alarm (=0)	4
BCP 2: D0220 3LAHHHE006G2 Boiler G2 Water Level High/High Alarm (=0)	4
BCP 2: D0196 3LAHHHE006D2 Boiler D2 Water Level High/High Alarm (=0)	4

Após a validação dos grupos gerados, o próximo passo foi classificar novas amostras de dados utilizando os rótulos identificados.

O classificador foi aplicado na amostra **clusterizada**, com 40.000 registros, utilizando o **SVC** com *kernel* linear. Após o treinamento do modelo de classificação, a acurácia alcançada foi de **99,8%** no conjunto de teste, correspondente à **20% do total da amostra**. Foram obtidos **248** *clusters* no total.

Uma das justificativas para esse resultado foi o pré-processamento adequado das mensagens dos alarmes. Ao eliminar *tags* dos equipamentos e outros elementos com menor significância semântica, o conteúdo das mensagens pode se tornar idêntico ou mais similar. Em muitos casos do conjunto de dados utilizado, a diferença entre os registros está apenas nos identificadores dos motores ou dos equipamentos envolvidos. Além disso, a limpeza dos dados de texto contribui diretamente para melhores resultados em técnicas envolvendo aprendizado de máquina.

Após a execução do algoritmo de classificação em uma amostra aleatória **não rotulada** com 40.000 registros, obteve-se **2.177** mensagens de alarmes para o *cluster* 4 e **252** mensagens para o *cluster* 8. Observando Tabela 10, os dados foram classificados de forma muito consistente, pois, para os mesmo rótulos de *clusters*, as mensagens possuem o

mesmo padrão das obtidas na etapa de clusterização.

Tabela 10 – *Clusters* 4 e 8 da amostra classificada.

Message	**Rótulo - Classificação**
BCP 2: D0292 3LAHHHE006P2 Boiler P2 Water Level High/High Alarm (=0)	4
BCP 2: D0244 3LAHHHE006J2 Boiler J2 Water Level High/High Alarm (=0)	4
BCP 2: D0220 3LAHHHE006G2 Boiler G2 Water Level High/High Alarm (=0)	4
BCP 2: D0196 3LAHHHE006D2 Boiler D2 Water Level High/High Alarm (=0)	4
CCP 01: Radiator Fan 27HE003L1Q1 Unselected mode	8
CCP 01: Radiator Fan 42HE003L1Q1 Unselected mode	8
CCP 02: Radiator Fan 3HE003L2Q2 Unselected mode	8
CCP 01: Radiator Fan 26HE003R1V1 Unselected mode	8
CCP 02: Radiator Fan 3HE003F2K2 Unselected mode	8
CCP 01: Radiator Fan 23HE003L1Q1 Unselected mode	8

Buscando uma melhor visualização dos erros entre os *clusters* reais (amostra clusterizada) e dos rótulos preditos pelo algoritmo de classificação (amostra classificada), foi implementado um código para gerar um gráfico que contém a distribuição das predições por *cluster*, conforme Figura 30. Utilizou-se uma função $y = x$ para representar os rótulos preditos no eixo y e os *clusters* no eixo x, destacando os rótulos que divergiram dos valores reais.

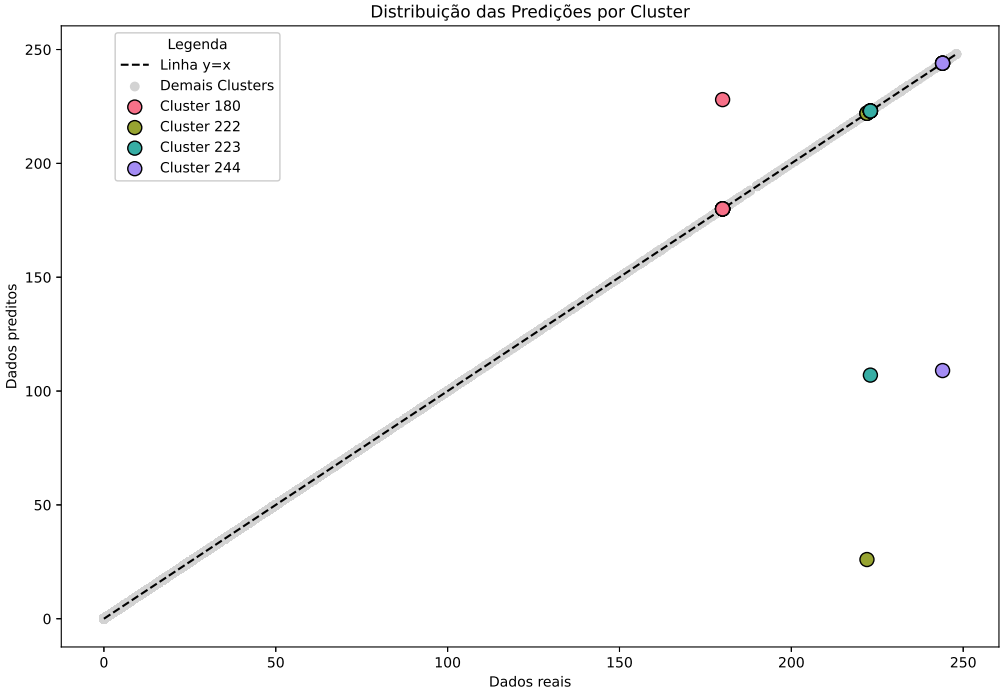


Figura 30 – Distribuição das predições por *cluster*.

Por meio da Figura 30, verificou-se que apenas 4 dos 248 rótulos obtidos pela classificação apresentaram divergências em relação à amostra clusterizada. Foram calculadas as taxas de erro para cada um destes rótulos, observando o valor complementar da acurácia. Na Tabela 11 é possível visualizar as taxas de erro por rótulo.

Tabela 11 – Taxas de erro - Rótulos com acurácia inferior a 100%.

Cluster	Taxa de Erro (%)
180	11.11
222	25.00
223	12.50
244	20.00

Apesar das divergências entre os valores verdadeiros e os preditos apresentados nos *clusters* da Tabela 11, esses casos correspondem a apenas 1.70% de toda a amostra.

Na fase final de desenvolvimento desse trabalho, com o auxílio da Universidade Federal da Paraíba, foi possível acessar uma máquina com configurações de *hardware*

significativamente superiores em comparação à utilizada anteriormente para processar os algoritmos. Dessa forma, tornou-se factível a execução do algoritmo de clusterização para 400.000 registros, distribuídos em amostras aleatórias de 100.000, para os anos de 2019 a 2022.

Analogamente aos passos anteriores, realizaram-se iterações a fim de encontrar o valor ótimo de *clusters*. Para o método do cotovelo, o número de *clusters* obtido foi 3, inferior ao valor da amostra reduzida utilizada anteriormente. A Figura 31 ilustra o comportamento dos valores de k em função da soma dos quadrados das distâncias *intra-cluster*:

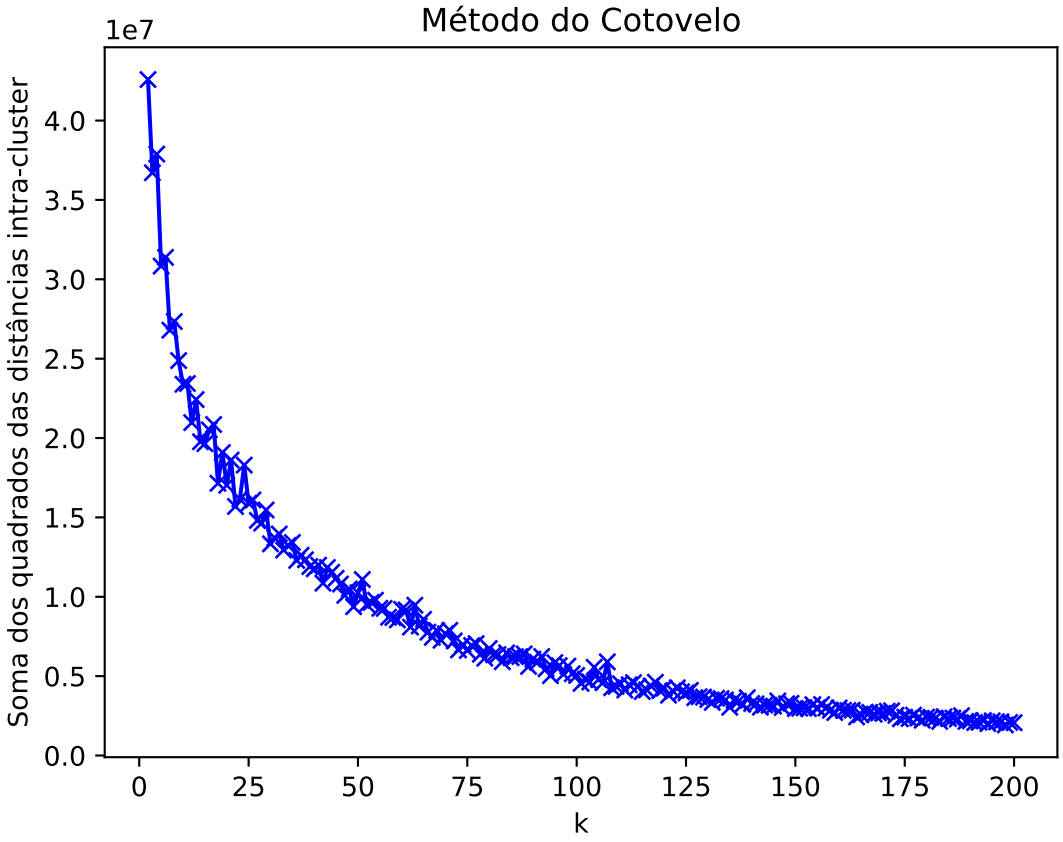


Figura 31 – Método do cotovelo - Algoritmo K-Means.

Já para o método do coeficiente de *silhouette*, o número ótimo de *clusters* encontrado foi de **199**, conforme Figura 32.

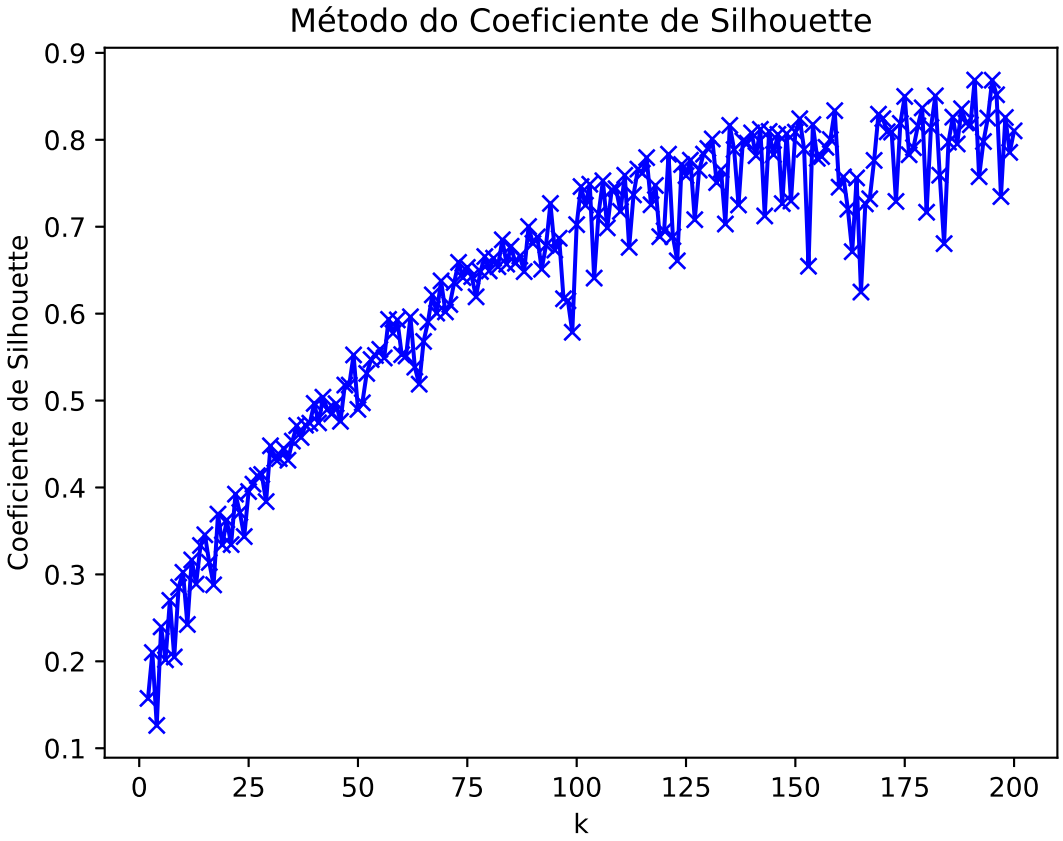


Figura 32 – Método do Coeficiente de *Silhouette* - Algoritmo K-Means.

A distribuição da volumetria dos alarmes por cada *cluster* obtido pode ser visualizada na Figura 33.

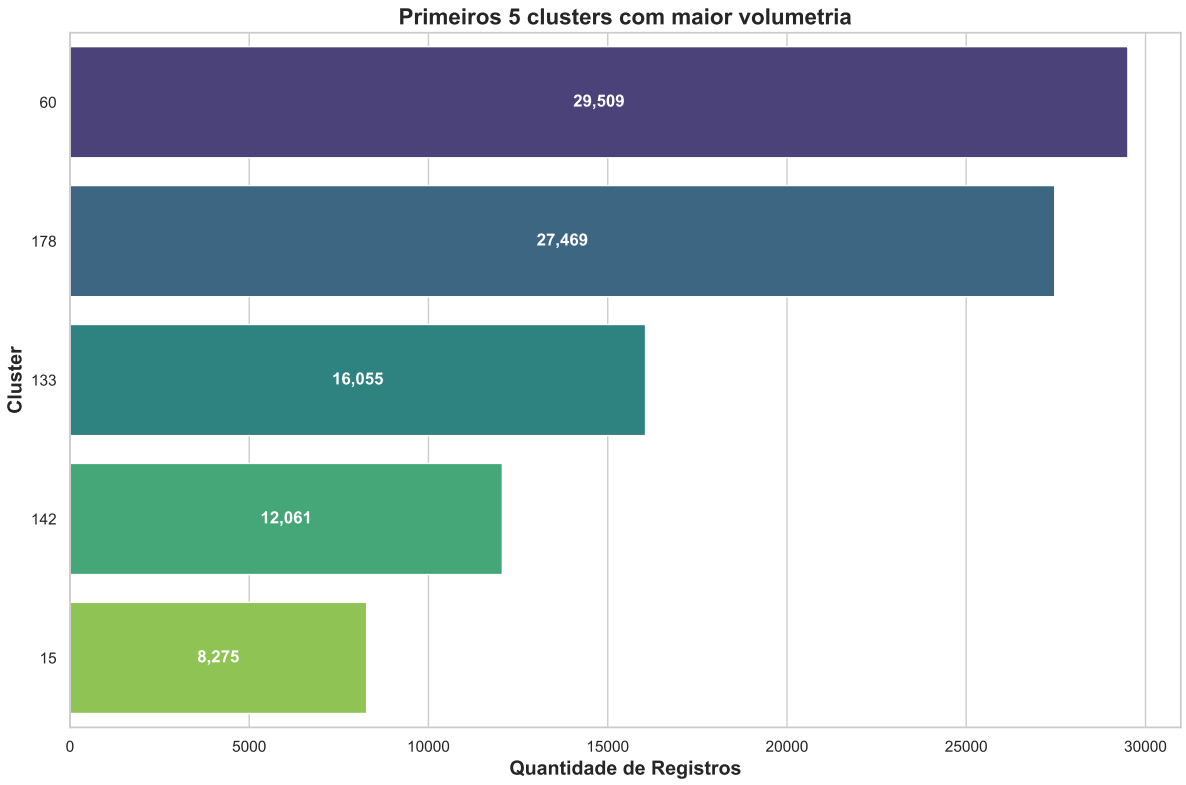


Figura 33 – Quantidade de alarmes por *cluster*.

Com a amostra significativamente maior, o resultado do agrupamento dos alarmes também foi muito eficiente. Para demonstrar este resultado, foram geradas amostras de cinco registros para cada *cluster* encontrado, conforme Tabela 12.

Tabela 12 – Amostra de 3 mensagens - 10 *clusters* com maior volumetria.

Cluster	**Mensagem**
133	GCP 15: D0167 Cylinder Lubrication 1EM2470 Switching Failure
133	GCP 04: D0167 Cylinder Lubrication 1EM2470 Switching Failure
133	GCP 40: D0167 Cylinder Lubrication 1EM2470 Switching Failure
142	GCP 19: D0177 1/2PTT001 - Plausibility Failure Redundant Sensors
142	GCP 02: D0174 1/2PT2170 - Plausibility Failure Redundant Sensors
142	GCP 15: D0175 1/2PT2570 - Plausibility Failure Redundant Sensors
146	GCP 17: L0051 1PAL3470 Nozzle CW Pressure Engine Inlet Low
146	GCP 01: L0051 1PAL3470 Nozzle CW Pressure Engine Inlet Low
146	GCP 28: L0051 1PAL3470 Nozzle CW Pressure Engine Inlet Low
15	GCP 11: D0102 PDSH2105 Differential Pressure Switch High
15	GCP 17: D0102 PDSH2105 Differential Pressure Switch High
15	GCP 37: D0102 PDSH2105 Differential Pressure Switch High
178	BCP 2: D0220 3LAHHHE006G2 Boiler G2 Water Level High/High Alarm (=0)
178	BCP 2: D0220 3LAHHHE006G2 Boiler G2 Water Level High/High Alarm (=0)
178	BCP 2: D0220 3LAHHHE006G2 Boiler G2 Water Level High/High Alarm (=0)
39	GCP 31: Pump P007 Unselected mode
39	GCP 11: Pump P007 Unselected mode
39	GCP 35: Pump P007 Unselected mode
60	BCP 2: D0221 3LALHE006G2 Boiler G2 Water Level Low Alarm (=0)
60	BCP 2: D0221 3LALHE006G2 Boiler G2 Water Level Low Alarm (=0)
60	BCP 2: D0221 3LALHE006G2 Boiler G2 Water Level Low Alarm (=0)
65	CCP 01: Radiator Fan 48HE003A1E1 Unselected mode
65	CCP 02: Radiator Fan 15HE003F2K2 Unselected mode
65	CCP 02: Radiator Fan 15HE003A2E2 Unselected mode
74	CCP 02: Radiator Fan 20HE003F2K2
74	CCP 02: Radiator Fan 28HE003R2V2
74	CCP 01: Radiator Fan 11HE003F1K1
78	GCP 08: L0203 1TAH6575B Exhaust Gas Temp Before TC B High
78	GCP 33: L0203 1TAH6575B Exhaust Gas Temp Before TC B High
78	GCP 38: L0203 1TAH6575B Exhaust Gas Temp Before TC B High

Analisando os padrões de sequências de alarmes pertencentes a diferentes *clusters*, em um intervalo de até dez minutos após cada registro observado, foram obtidas as dez transições observada com maior frequência na amostra clusterizada com 400.000 dados. No eixo horizontal da Figura 34 (**Cluster Atual**), estão os rótulos dos primeiros *clusters* de cada transição observada, e as cores da legenda indicam os *clusters* seguintes (**Próximo Cluster**). No eixo vertical encontram-se as frequências de cada padrão de sequência encontrado, e os rótulos correspondentes em branco, localizados em todas as barras do gráfico.

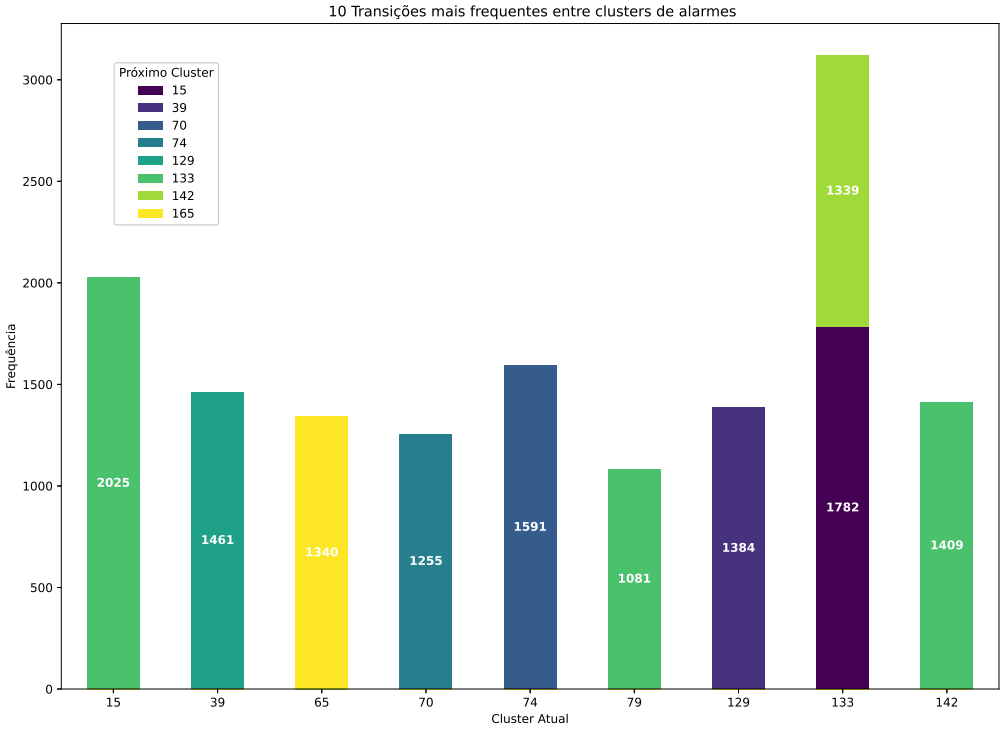


Figura 34 – Frequência de transições entre *clusters*.

Outra observação sobre a Figura 34 é a existência de menos frequência de transições entre *clusters* que o esperado, pois, apesar da amostra clusterizada conter 400.000 dados, os registros foram escolhidos de maneira aleatória para cada ano, e não foram ordenados pelas datas e horários dos alarmes.

Na Tabela 13, é possível verificar as mensagens dos alarmes e a data/hora dos registros que se enquadram nas transições **(74, 70)** e **(133, 15)**.

Tabela 13 – Exemplos de transições de *clusters* encontradas.

2019-10-17 11:01:29.623	74	CCP 01: Radiator Fan 2HE003L1Q1	2019-10-17 11:01:29.653	70	CCP 01: HT8 Butterfly Flap 8BF3110F1K1
2021-11-22 18:31:50.430	74	CCP 01: Radiator Fan 41HE003F1K1	2021-11-22 18:31:50.430	70	CCP 01: LT9 Butterfly Flap 9BF4614F1K1
2020-01-27 16:55:16.903	74	CCP 01: Radiator Fan 49HE003A1E1	2020-01-27 16:55:16.903	70	CCP 01: LT4 Butterfly Flap 4BF4614A1E1
2022-03-23 04:47:37.597	133	GCP 40: D0167 Cylinder Lubrication 1EM2470 Switching Failure	2022-03-23 04:47:45.470	15	GCP 37: D0102 PDSH2105 Differential Pressure Switch High
2022-01-11 03:47:49.750	133	GCP 26: D0167 Cylinder Lubrication 1EM2470 Switching Failure	2022-01-11 03:47:49.750	15	GCP 26: D0102 PDSH2105 Differential Pressure Switch High
2022-03-01 03:45:48.627	133	GCP 11: D0167 Cylinder Lubrication 1EM2470 Switching Failure	2022-03-01 03:45:48.627	15	GCP 11: D0102 PDSH2105 Differential Pressure Switch High

Devido ao fato das máquinas e outros equipamentos operarem de forma simultânea, nas transições de *clusters* é possível que hajam alarmes pertencentes a diferentes motores, por exemplo. Porém, para analisar a causalidade entre eles, devem ser considerados apenas os registros do mesmo motor, por exemplo. Na Tabela 13, por exemplo, na última linha ambos os alarmes se referem ao mesmo motor, conforme o identificador *GCP 11* no início das mensagens. Assim, torna-se factível analisar se há relação de causalidade entre os registros. Neste caso, há uma forte causalidade entre os alarmes, pois quando ocorre a falha de lubrificação dos cilindros do motor, representado pela mensagem *GCP 11: D0167 Cylinder Lubrication 1EM2470 Switching Failure* (*Cluster* 133) o *switch* representado no alarme *GCP 11: D0102 PDSH2105 Differential Pressure Switch High* é acionado.

Por fim, dados os rótulos dos alarmes obtidos por meio da clusterização, propõe-se a implementação de um modelo de classificação com execução *online* para desenvolver uma tela à parte no supervisório da planta objeto deste estudo. Com os *clusters* obtidos nesta pesquisa, é possível selecionar os mais críticos e mapeá-los e, dessa forma, filtrar apenas os registros desejados. A Figura 35, por meio de um esquemático, exemplifica a tela contendo apenas os alarmes priorizados e o filtro por *cluster*.

Data/Hora	Mensagem	Severidade	Cluster
2019-04-06 22:24:53.493	**BCP 2: D0220 3LAHHHE006G2 Boiler G2 Water Level High/High Alarm (=0)**	1	178
2019-04-06 22:25:35.433	**BCP 2: D0221 3LALHE006G2 Boiler G2 Water Level Low Alarm (=0)**	1	60

Figura 35 – Esquemático de proposta de tela com filtro por *cluster*.

6 CONCLUSÃO

Neste trabalho, foi desenvolvida uma metodologia para o agrupamento inteligente de alarmes utilizando técnicas de aprendizado de máquina, demonstrada com um caso de uso aplicado a uma usina termoelétrica. Inicialmente, foi realizada uma análise exploratória para justificar a necessidade de rotular os registros em *clusters*, devido ao grande volume de alarmes recebidos diariamente pela área operacional. Observou-se que aproximadamente 73% dos registros consistiram em alarmes idênticos e repetidos dentro de um intervalo de dez minutos, o que levou à proposta de criar um mecanismo para evitar essas repetições.

Com a clusterização dos alarmes e compreendendo o impacto causado pela volumetria considerável de registros, um dos maiores focos deste trabalho foi propor a priorização dos alarmes através da filtragem dos *clusters* considerados mais críticos para a área operacional.

Devido às limitações computacionais para lidar com a grande quantidade de dados extraída do sistema *SCADA* da planta, foi utilizada uma amostra reduzida para realizar a clusterização das mensagens dos alarmes, contendo 40.000 registros. Devido aos registros não possuírem rótulos ou classificações bem definidas, foi empregado o algoritmo *K-means*, um método de aprendizado não supervisionado, para agrupar alarmes semelhantes. Para a escolha do número ideal de *clusters*, o método do cotovelo sugeriu um número ideal de 9 grupos, porém estes apresentaram uma heterogeneidade significativa nos dados *intra-cluster*. Utilizando o coeficiente de *silhouette*, foram obtidos 249 *clusters*, nos quais os alarmes foram bem agrupados, mantendo apenas mensagens similares em cada *cluster*. Com os dados rotulados, foi aplicado um algoritmo de classificação, *SVC* com *kernel* linear, em uma nova amostra não rotulada, contendo também 40.000 registros. Os resultados foram consideravelmente satisfatórios, uma vez que foram obtidos, para os mesmos rótulos da amostra clusterizada, os mesmo padrões de mensagens.

Já no estágio final da pesquisa, por meio do acesso a melhores recursos computacionais, a clusterização foi realizada com uma amostra de dados consideravelmente maior, com 400.000 registros. Aplicando o método do coeficiente de *silhouette*, foi obtido um número ideal de **199** *clusters*. Apesar da divergência nos rótulos dos *clusters* em relação à amostra reduzida, observou-se que os padrões de agrupamento das mensagens se mantiveram.

Ainda, utilizando a amostra clusterizada com 400.000 dados, foram realizadas análises de padrões de transições recorrentes entre *clusters*, ou seja, alarmes de diferentes grupos que ocorrem de forma sequencial no intervalo de até 10 minutos. Para as transições que ocorrem com maior frequência, foram observadas sequências de alarmes que possuem relação de causalidade entre si, possibilitando determinar as causas raiz de alguns registros,

e evidenciar as relações diretas existentes entre diferentes equipamentos no surgimento de falhas, e que podem contribuir diretamente para um planejamento estratégico de manutenção mais proativo por parte dos gestores.

Como uma das principais contribuições deste trabalho, é proposta a implementação de um modelo de classificação de texto, utilizando os rótulos obtidos com a maior amostra obtida via clusterização, para aplicar o algoritmo aqui proposto (*SVC* com *kernel* linear) de forma *online*, integrado ao sistema supervisório da usina, possibilitando a criação de telas de monitoramento à parte e a filtragem dos *clusters* mais críticos, melhorando a eficiência na gestão dos alarmes. Para tanto, o *SCADA* da planta, objeto de estudo desse trabalho, possui uma plataforma para gerenciamento de dados e um módulo adicional, responsável pela execução de códigos em linguagem *Python* baseados em eventos simulados ou de tempo real.

Para trabalhos futuros, destacam-se os seguintes pontos a serem explorados:

- Utilização de novos algoritmos para clusterização e classificação de texto, bem como compará-los aos já utilizados;
- Implementar algoritmos para predição de falhas de processo, utilizando dados históricos dos sensores da planta e dos alarmes;
- Explorar de uma maneira mais completa a determinação de causas raiz dos alarmes, a fim de catalogar detalhadamente as falhas, subsistemas ofensores e possíveis erros operacionais;

REFERÊNCIAS

AARDT, D. van. More data is only useful if it leads to more wisdom. *IT in Manufacturing. URL: https://www. instrumentation. co. za/8423a (Acessado em 01-10-2021)*, 2015. Citado na página 12.

AHMED, K. et al. Similarity analysis of industrial alarm flood data. *IEEE Transactions on Automation Science and Engineering*, v. 10, p. 452–457, 2013. Citado na página 13.

BALAKRISHNAN, V.; ETHEL, L.-Y. Stemming and lemmatization: A comparison of retrieval performances. *Lecture Notes on Software Engineering*, v. 2, p. 262–267, 01 2014. Citado na página 25.

BEZERRA, A. et al. Extracting value from industrial alarms and events: A data-driven approach based on exploratory data analysis. *Sensors*, MDPI, v. 19, n. 12, p. 2772, 2019. Citado na página 12.

BHARADWAJ; PRAKASH, K. B.; KANAGACHIDAMBARESAN, G. R. Pattern recognition and machine learning. *Programming with TensorFlow*, 2021. Disponível em: <https://api.semanticscholar.org/CorpusID:234262311>. Citado na página 35.

BINDRA, K.; MISHRA, A. A detailed study of clustering algorithms. *2017 International Conference on Current Trends in Computer, Electrical, Electronics and Communication (CTCEEC)*, p. 752–757, 2017. Disponível em: <https://api.semanticscholar.org/CorpusID:52163140>. Citado na página 29.

BINDRA, K.; MISHRA, A.; SURYAKANT. Effective data clustering algorithms. *Advances in Intelligent Systems and Computing*, 2018. Disponível em: <https://api.semanticscholar.org/CorpusID:58333639>. Citado na página 29.

BIRD, S.; KLEIN, E.; LOPER, E. *Natural language processing with Python: analyzing text with the natural language toolkit*. [S.l.]: "O'Reilly Media, Inc.", 2009. Citado na página 43.

BRITO, P. F. de; GOMES, L. P. da S. G. Desenvolvimento do módulo de pre-processamento da ferramenta sentimentall. *Revista Singular - Engenharia, Tecnologia e Gestão*, 2019. Disponível em: <https://api.semanticscholar.org/CorpusID:127133548>. Citado na página 43.

CAI, S. et al. Clustering analysis of process alarms using word embedding. *Journal of Process Control*, 2019. Disponível em: <https://api.semanticscholar.org/CorpusID:203137397>. Citado na página 13.

CAI, S. et al. Clustering analysis of process alarms using word embedding. *Journal of Process Control*, Elsevier, v. 83, p. 11–19, 2019. Citado na página 17.

COLLINS, M.; DUFFY, N. P. Convolution kernels for natural language. In: *Neural Information Processing Systems*. [s.n.], 2001. Disponível em: <https://api.semanticscholar.org/CorpusID:396794>. Citado 2 vezes nas páginas 35 e 36.

CORDEIRO, F.; RABELO, R. de A. L.; MOURA, R. S. Classification of irregularity communications in public ombudsmen using supervised learning algorithms. *Anais do XIX Encontro Nacional de Inteligência Artificial e Computacional (ENIAC 2022)*, 2022. Disponível em: <https://api.semanticscholar.org/CorpusID:255894490>. Citado na página 36.

COSTA, A. S. P. d. et al. Application of near-infrared for online monitoring of heavy fuel oil at thermoelectric power plants. part i: Development of chemometric models. *Industrial &Amp; Engineering Chemistry Research*, v. 58, p. 15681–15692, 2019. Citado na página 13.

COSTA, L. V. da; CASTRO, C. L. de. Regularizaçao do classificador de kernel density estimation com funçoes de base radial. 2019. Citado na página 35.

CRUZ, F. P. Máquina de vectores soporte adaptativa y compacta. In: . [s.n.], 2000. Disponível em: <https://api.semanticscholar.org/CorpusID:169368104>. Citado na página 35.

DAVIS, J. et al. Smart manufacturing, manufacturing intelligence and demand-dynamic performance. *Computers & Chemical Engineering*, Elsevier, v. 47, p. 145–156, 2012. Citado na página 12.

DEVLIN, J. et al. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018. Disponível em: <http://arxiv.org/abs/1810.04805>. Citado na página 27.

DEVLIN, J. et al. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. Citado 4 vezes nas páginas 7, 26, 27 e 28.

DEVLIN, J. et al. Bert: Pre-training of deep bidirectional transformers for language understanding. In: *North American Chapter of the Association for Computational Linguistics*. [s.n.], 2019. Disponível em: <https://api.semanticscholar.org/CorpusID:52967399>. Citado na página 27.

ELER, D. M. et al. Analysis of document pre-processing effects in text and opinion mining. *Inf.*, v. 9, p. 100, 2018. Disponível em: <https://api.semanticscholar.org/CorpusID:46923701>. Citado na página 24.

ESPINOZA, C. M. P. et al. Predicción del clima por medio de una estación meteorológica y la medición de la precipitación por sistema de pesaje. In: . [s.n.], 2021. Disponível em: <https://api.semanticscholar.org/CorpusID:247280154>. Citado na página 35.

FAHIMIPIREHGALIN, M.; WEISS, I.; VOGEL-HEUSER, B. Causal inference in industrial alarm data by timely clustered alarms and transfer entropy. In: *2020 European Control Conference (ECC)*. [S.l.: s.n.], 2020. p. 2056–2061. Citado na página 17.

FUJIO, M.; MATSUMOTO, Y. Japanese dependency structure analysis based on lexicalized statistics. In: *Proceedings of the Third Conference on Empirical Methods for Natural Language Processing*. [S.l.: s.n.], 1998. p. 87–95. Citado na página 35.

GÉRON, A. *Mãos à obra: aprendizado de máquina com Scikit-Learn, Keras & TensorFlow: Conceitos, ferramentas e técnicas para a construção de sistemas inteligentes*. [S.l.]: Alta Books, 2021. v. 1. Citado 4 vezes nas páginas 29, 31, 32 e 34.

GIMÉNEZ, J.; MARQUEZ, L. Fast and accurate part-of-speech tagging: The svm approach revisited. In: *RANLP*. [S.l.: s.n.], 2003. p. 153–163. Citado na página 35.

GONZALEZ, E.; REDER, M.; MELERO, J. J. Scada alarms processing for wind turbine component failure detection. In: IOP PUBLISHING. *Journal of Physics: Conference Series*. [S.l.], 2016. v. 753, n. 7, p. 072019. Citado na página 22.

GURUSAMY, V.; KANNAN, S. Preprocessing techniques for text mining. In: . [S.l.: s.n.], 2014. Citado 2 vezes nas páginas 24 e 25.

HACIOGLU, K. et al. Semantic role labeling by tagging syntactic chunks. In: *Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL-2004) at HLT-NAACL 2004*. [S.l.: s.n.], 2004. p. 110–113. Citado na página 35.

HARTMANN, B.; KING, W. P.; NARAYANAN, S. Digital manufacturing: The revolution will be virtualized. *McKinsey & Company*, 2015. Citado na página 12.

HEEYAS Limited. <https://www.heeyas.co.uk/listings/729941-man-stx-18v32-40-hfo-or-mdo-generators-new>. Acesso em: 01 jul. 2024. Citado 2 vezes nas páginas 7 e 40.

ILIC, E.; GARCÍA-MARTÍNEZ, M.; PASTOR, M. S. A review of text classification models from bayesian to transformers. In: *Swiss Text Analytics Conference*. [s.n.], 2022. Disponível em: <https://api.semanticscholar.org/CorpusID:259121342>. Citado na página 36.

ISOZAKI, H.; KAZAWA, H. Efficient support vector classifiers for named entity recognition. In: *COLING 2002: The 19th International Conference on Computational Linguistics*. [S.l.: s.n.], 2002. Citado na página 35.

JAN, T. Impact of text representation techniques on clustering models. In: . [s.n.], 2022. Disponível em: <https://api.semanticscholar.org/CorpusID:264149082>. Citado na página 13.

KAESTNER, C. A. A. Support vector machines and kernel functions for text processing. *RITA*, v. 20, p. 130–154, 2013. Disponível em: <https://api.semanticscholar.org/CorpusID:17924205>. Citado na página 35.

KALRA, V.; AGGARWAL, R. Importance of text data preprocessing & implementation in rapidminer. In: *International Conference on Information Technology and Knowledge Management*. [s.n.], 2018. Disponível em: <https://api.semanticscholar.org/CorpusID:65238534>. Citado na página 24.

KUDO, T.; MATSUMOTO, Y. Use of support vector learning for chunk identification. In: *Fourth conference on computational natural language learning and the second learning language in logic workshop*. [S.l.: s.n.], 2000. Citado na página 35.

LEBRET, R. Word embeddings for natural language processing. In: . [s.n.], 2016. Disponível em: <https://api.semanticscholar.org/CorpusID:63947950>. Citado na página 28.

LEE, Y. K.; NG, H. T.; CHIA, T. K. Supervised word sense disambiguation with support vector machines and multiple knowledge sources. In: *Proceedings of SENSEVAL-3, the third international workshop on the evaluation of systems for the semantic analysis of text*. [S.l.: s.n.], 2004. p. 137–140. Citado na página 35.

LI, Y.; BONTCHEVA, K.; CUNNINGHAM, H. Svm based learning system for information extraction. In: SPRINGER. *International Workshop on Deterministic and Statistical Methods in Machine Learning*. [S.l.], 2004. p. 319–339. Citado na página 35.

LI, Y.; BONTCHEVA, K.; CUNNINGHAM, H. Adapting svm for natural language learning: A case study involving information extraction. *Natural Language Engineering*, v. 15, n. 2, p. 241–271, 2009. Citado na página 35.

MAATEN, L. Van der; HINTON, G. Visualizing data using t-sne. *Journal of machine learning research*, v. 9, n. 11, 2008. Citado na página 38.

MANCA, G.; DIX, M.; FAY, A. Clustering of similar historical alarm subsequences in industrial control systems using alarm series and characteristic coactivations. *IEEE Access*, v. 9, p. 154965–154974, 2021. Citado na página 17.

MARUGAN, A. P.; MÁRQUEZ, F. P. G. Advanced analytics for detection and diagnosis of false alarms and faults: A real case study. *Wind Energy*, v. 22, p. 1622–1635, 2019. Citado na página 23.

MEDIDA, S. Pocket guide on industrial automation. *IDC technologies*, 2008. Citado na página 20.

MEHTA, V.; BAWA, S.; SINGH, J. Weclustering: word embeddings based text clustering technique for large datasets. *Complex & Intelligent Systems*, v. 7, p. 3211 – 3224, 2021. Disponível em: <https://api.semanticscholar.org/CorpusID:237439952>. Citado na página 13.

MOSCHITTI, A. State-of-the-art kernels for natural language processing. In: *Annual Meeting of the Association for Computational Linguistics*. [s.n.], 2012. Disponível em: <https://api.semanticscholar.org/CorpusID:17316648>. Citado na página 35.

NAKAGAWA, T.; KUDO, T.; MATSUMOTO, Y. Unknown word guessing and part-of-speech tagging using support vector machines. In: *NLPRS*. [S.l.: s.n.], 2001. p. 325–331. Citado na página 35.

NEELIMA, A.; MEHROTRA, S. A comprehensive review on word embedding techniques. *2023 International Conference on Intelligent Systems for Communication, IoT and Security (ICISCoIS)*, p. 538–543, 2023. Disponível em: <https://api.semanticscholar.org/CorpusID:258219905>. Citado na página 28.

PETERS, M. E. et al. Deep contextualized word representations. In: WALKER, M.; JI, H.; STENT, A. (Ed.). *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, 2018. p. 2227–2237. Disponível em: <https://aclanthology.org/N18-1202>. Citado na página 26.

PETTA, V. de M. et al. Clusterizaçao de dados utilizando o algoritmo de enxame de vagalumes. In: . [s.n.], 2020. Disponível em: <https://api.semanticscholar.org/CorpusID:216361847>. Citado na página 29.

QIU, Y. et al. Wind turbine scada alarm analysis for improving reliability. *Wind Energy*, v. 15, p. 951–966, 2011. Citado 2 vezes nas páginas 22 e 23.

RADFORD, A. et al. Improving language understanding with unsupervised learning. Technical report, OpenAI, 2018. Citado na página 27.

RAMKUMAR, A. S.; NETHRAVATHY, R. Text document clustering using k-means algorithm. *Int. Res. J. Eng. Technol*, v. 6, p. 1164–1168, 2019. Citado na página 13.

RAVI, J.; KULKARNI, S. Text embedding techniques for efficient clustering of twitter data. *Evolutionary Intelligence*, Springer, v. 16, n. 5, p. 1667–1677, 2023. Citado na página 17.

ROTHENBERG, D. H. *Alarm management for process control: a best-practice guide for design, implementation, and use of industrial alarm systems*. [S.l.]: Momentum Press, 2009. Citado na página 22.

SHEN, Y.; LIU, J. Comparison of text sentiment analysis based on bert and word2vec. In: *2021 IEEE 3rd International Conference on Frontiers Technology of Information and Computer (ICFTIC)*. [S.l.: s.n.], 2021. p. 144–147. Citado na página 18.

SILVA, M. V. J. da et al. Preprocessing applied to legal text mining: analysis and evaluation of the main techniques used. *Anais do XX Encontro Nacional de Inteligência Artificial e Computacional (ENIAC 2023)*, 2023. Disponível em: <https://api.semanticscholar.org/CorpusID:264527717>. Citado na página 24.

SUBAKTI, A.; MURFI, H.; HARIADI, N. The performance of bert as data representation of text clustering. *Journal of big Data*, Springer, v. 9, n. 1, p. 15, 2022. Citado na página 29.

SULEIMAN, D.; AWAJAN, A. A. Comparative study of word embeddings models and their usage in arabic language applications. *2018 International Arab Conference on Information Technology (ACIT)*, p. 1–7, 2018. Disponível em: <https://api.semanticscholar.org/CorpusID:85496869>. Citado na página 28.

SVERKO, M.; GRBAC, T. G.; MIKUC, M. Scada systems with focus on continuous manufacturing and steel industry: A survey on architectures, standards, challenges and industry 5.0. *IEEE Access*, v. 10, p. 109395–109430, 2022. Citado na página 22.

TAKAHASHI, C. C. Mapeamento explícito como kernel em aprendizado de máquinas de vetores de suporte. In: . [s.n.], 2015. Disponível em: <https://api.semanticscholar.org/CorpusID:209992029>. Citado na página 36.

YAMADA, H.; MATSUMOTO, Y. Statistical dependency analysis with support vector machines. In: *Proceedings of the eighth international conference on parsing technologies*. [S.l.: s.n.], 2003. p. 195–206. Citado na página 35.

YANG, S. et al. Frequent alarm pattern mining of industrial alarm flood sequences by an improved prefixspan algorithm. *Processes*, MDPI, v. 11, n. 4, p. 1169, 2023. Citado na página 22.

YANG, X.; DZIEGIELEWSKI, B. Water use by thermoelectric power plants in the united states1. *JAWRA Journal of the American Water Resources Association*, v. 43, p. 160–169, 2007. Citado na página 12.

ZHANG, X. et al. Similarity analysis of industrial alarm floods based on word embedding and move-split-merge distance. In: *2023 IEEE 6th International Conference on Industrial Cyber-Physical Systems (ICPS)*. [S.l.: s.n.], 2023. p. 1–6. Citado na página 18.

ZHOU, G. et al. Exploring various knowledge in relation extraction. In: *Proceedings of the 43rd annual meeting of the association for computational linguistics (acl'05)*. [S.l.: s.n.], 2005. p. 427–434. Citado na página 35.