UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
CURSO DE GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Andressa Wickert Kreutz

MULTICLASSIFICADOR PARA DETECÇÃO DE *OUTLIERS* EM DADOS GERADOS POR SENSORES DE MONITORAMENTO AMBIENTAL

Santa Maria, RS
2022

Andressa Wickert Kreutz

MULTICLASSIFICADOR PARA DETECÇÃO DE *OUTLIERS* EM DADOS GERADOS POR SENSORES DE MONITORAMENTO AMBIENTAL

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia Elétrica da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para a obtenção do grau de **Bacharel em Engenharia Elétrica**

Orientador: Prof. Dr. Natanael Rodrigues Gomes

Santa Maria, RS

2022

Wickert Kreutz, Andressa

Multiclassificador para detecção de *outliers* em dados gerados por sensores de monitoramento ambiental / por Andressa Wickert Kreutz. – 2022.

70 f.: il.; 30 cm.

Orientador: Natanael Rodrigues Gomes

Trabalho de Conclusão de Curso - Universidade Federal de Santa Maria, Centro de Tecnologia, Curso de Graduação em Engenharia Elétrica, RS, 2022.

1. Cidades inteligentes. 2. Internet das coisas. 3. Outliers. 4. Multiclassificador. I. Rodrigues Gomes, Natanael. II.Multiclassificador para detecção de *outliers* em dados gerados por sensores de monitoramento ambiental.

E-mail: andressakreutz@gmail.com

Andressa Wickert Kreutz

MULTICLASSIFICADOR PARA DETECÇÃO DE *OUTLIERS* EM DADOS GERADOS POR SENSORES DE MONITORAMENTO AMBIENTAL

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia Elétrica da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para a obtenção do grau de **Bacharel em Engenharia Elétrica**

Aprovado em 17 de fevereiro de 2022:

Natanael Rodrigues Gomes, Dr. (UFSM)
(Presidente/Orientador)

Candice Müller, Dra. (UFSM)

Andrei Piccinini Legg, Dr. (UFSM)

Santa Maria, RS

2022

DEDICATÓRIA

Dedico este trabalho aos meus pais Luiz e Neiva,
que nunca medem esforços para apoiar os meus sonhos

AGRADECIMENTOS

Agradeço a UFSM, pela oportunidade de um ensino público de qualidade.

Agradeço ao meu professor orientador Natanael Rodrigues Gomes, por todos os ensinamentos e paciência.

Agradeço a minha família, sobretudo meus pais, que sempre me apoiam em tudo e são a minha fonte de motivação e suporte, e todos aqueles que de alguma forma fizeram parte desta história. A graduação é marcada pelas pessoas que nos rodeiam e eu com certeza tenho os melhores comigo.

“Anything that can be connected, will be connected.”

(JACOB MORGAN)

RESUMO

MULTICLASSIFICADOR PARA DETECÇÃO DE *OUTLIERS* EM DADOS GERADOS POR SENSORES DE MONITORAMENTO AMBIENTAL

AUTORA: ANDRESSA WICKERT KREUTZ
ORIENTADOR: NATANAEL RODRIGUES GOMES

No contexto de Internet das Coisas (IoT - *Internet of Things*), uma das aplicações são cidades inteligentes, nas quais a tecnologia é empregada de modo a melhorar a qualidade de vida dos seus cidadãos. Neste cenário, existe uma intensa geração de dados a partir de sensores, os quais fornecem informações importantes sobre o seu entorno e podem ser utilizados na tomada de decisões. Porém, também emergem novos desafios, tais como a presença de *outliers*, isto é, valores que se diferenciam drasticamente dos outros do mesmo conjunto de dados, o que pode implicar em interpretações equivocadas. Em vista disto, o presente trabalho apresenta a construção de um algoritmo multiclassificador para detecção de *outliers* em dados obtidos de três sensores IoT de monitoramento ambiental. O multiclassificador é composto pela resposta de três técnicas, sendo duas delas estatísticas (Zscore e Zscore Modificado) e uma de clusterização (K-Means). Os métodos são avaliados e comparados por meio das métricas de desempenho de sensibilidade, precisão, especificidade e acurácia. Constata-se que o Zscore e Zscore Modificado exibem melhor desempenho em identificar anormalidades nos dados de distribuição unimodal, enquanto que K-Means possui maior eficiência nos dados de distribuição bimodal. Portanto, ao reunir as respostas no multiclassificador, as mesmas se complementam, obtendo-se um sistema de detecção de *outliers* mais robusto, com melhores métricas de desempenho.

Palavras-chave: Cidades inteligentes. Internet das coisas. Outliers. Multiclassificador.

ABSTRACT

MULTICLASSIFIER FOR DETECTING OUTLIERS IN DATA GENERATED BY ENVIRONMENTAL MONITORING SENSORS

AUTHOR: ANDRESSA WICKERT KREUTZ
ADVISOR: NATANAEL RODRIGUES GOMES

In the context of the Internet of Things (IoT), one of the applications are smart cities, in which technology is employed in order to improve the quality of life of its citizens. In this scenario, there is an intense data generation from sensors, which provide important information about their surroundings and can be used in decision making. However, new challenges also emerge, such as the presence of outliers, that is, values that differ drastically from others in the same dataset, which can lead to misinterpretations. In view of this, the current work presents the construction of a multiclassifier algorithm for outlier detection in data obtained from three environmental monitoring IoT sensors. The multiclassifier is composed of two statistical techniques (Zscore and Modified Zscore) and one of clustering (K-Means). The methods are evaluated and compared using performance metrics of sensitivity, precision, specificity and accuracy. It is found that Zscore and Modified Zscore exhibit better performance in identifying abnormalities in unimodal distribution data, while K-Means has higher efficiency in bimodal distribution data. Therefore, by gathering the responses in the multiclassifier, they complement each other, yielding a more robust outlier detection system with better performance metrics.

Keywords: Smart cities. Internet of things. Outliers. Multiclassifier.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

De acordo com a Organização das Nações Unidas (ONU), 55% da população mundial vive atualmente em áreas urbanas e estima-se que este percentual atinja 70% até 2050 (ONU, 2019). Este aumento significa uma elevação no número de pessoas residindo em um mesmo espaço, evidenciando também alguns problemas já existentes, tais como mobilidade urbana, poluição ambiental e sonora, infraestrutura, saúde pública, entre outros.

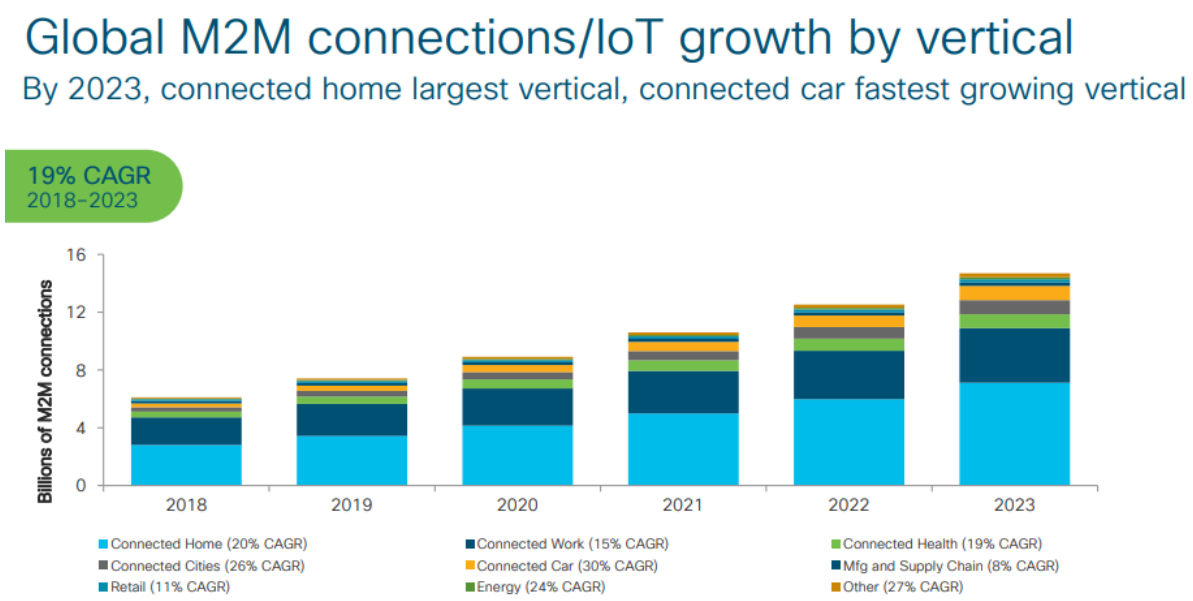
Paralelamente, as inovações em Tecnologia da Informação e Comunicação (TIC) também crescem em ritmo acelerado. Segundo dados da Associação Brasileira de Empresas do setor de TIC (Brasscom), este segmento cresceu 5,1% durante a pandemia, no ano de 2020 comparado a 2019 (Brasscom, 2021). O desenvolvimento desta área propicia algumas alternativas de soluções para os desafios citados anteriormente, sendo uma destas a denominada Internet das Coisas (IoT - *Internet of Things*).

Na literatura, são encontradas diversas definições para o termo IoT, existindo algumas divergências entre elas. Segundo Magrani (2018), este conceito pode ser explicitado como "um ambiente de objetos físicos interconectados com a internet por meio de sensores pequenos e embutidos, criando um ecossistema de computação onipresente (ubíqua), que busca facilitar o cotidiano das pessoas, introduzindo soluções funcionais nos processos do dia a dia". No geral, todas as definições retratam a IoT como sendo uma rede de "coisas", denominadas objetos inteligentes, que se comunicam entre si, coletando e processando dados e informações em um mundo hiperconectado.

A adesão aos dispositivos IoT está crescendo em ritmo acelerado, como apontado pelo Relatório Anual da Internet da Cisco (Cisco, 2020), a qual é um dos principais fabricantes mundiais de equipamentos para redes informáticas. Um dos gráficos deste relatório, retratado na Figura 1, evidencia a expectativa de crescimento global de conexões IoT, de 2018 a 2023. Analisando os resultados apresentados pela Cisco, estima-se que durante este período o número de tais conexões mais do que duplique, crescendo de cerca de 6 bilhões para aproximadamente 14 bilhões. Isto representa um aumento de 8 bilhões de conexões IoT em apenas 5 anos.

Ademais, estimativas da *Palo Alto Networks* (2020), empresa multinacional americana de segurança cibernética, apontam que aproximadamente 30% das interfaces finais de comunicações das empresas à internet são dispositivos IoT. Além disso, a empresa *International Data Corporation* (IDC), que desenvolve pesquisa, análise e consultoria no setor de TIC, prevê a

Figura 1 – Gráfico de crescimento da IoT



Fonte: (Cisco, 2020).

existência de 55,7 bilhões de dispositivos conectados em todo o mundo em 2025, sendo 75% destes vinculados a uma plataforma IoT. A IDC estima também que os 18,3 zettabytes de dados gerados em 2019 a partir de dispositivos IoT conectados cresçam para 73,1 zettabytes em 2025 (IDC, 2020), o que corresponde a $73,1 \cdot 10^{21}$ bytes.

Neste contexto, deve-se fazer uso deste elevado número de dispositivos e, consequentemente, de dados, tornando-os aliados em tomadas de decisões mais acertadas. Os recursos dos objetos inteligentes podem ser empregados para detectar informações úteis do seu entorno, a fim de analisá-las e, caso pretendido, tomar ações acerca das mesmas. Desse modo, provendo comunicação entre usuários e dispositivos e recursividade entre os dados obtidos pelos sensores e o ambiente a sua volta. Em vista disto, a partir da IoT, surge uma grande gama de novas aplicações, tais como as chamadas cidades inteligentes (*smart cities*).

Cidades inteligentes são aquelas onde a tecnologia é empregada de modo a otimizar a eficiência dos serviços, atendendo a demanda de seus habitantes em tempo real, por meio de uma massiva coleta e análise de dados obtidos por sensores e intensa comunicação entre estes. Essa coleta de dados proporciona informações relevantes sobre o meio urbano, as quais podem ser utilizadas de forma a melhorar a qualidade de vida dos cidadãos. Dentre as possíveis aplicações, pode-se citar as redes elétricas inteligentes (*smart grids*), o monitoramento ambiental inteligente e melhorias na mobilidade urbana (SANTOS et al., 2016), (SOUZA, 2020).

Além dos impactos positivos na qualidade de vida da população, a IoT deve apresentar implicações econômicas favoráveis. Segundo análise da empresa de consultoria de gestão McKinsey, em 2025 a IoT deve gerar, em nível mundial, receitas entre US$ 3,9 trilhões e US$ 11,1 trilhões, equivalendo, na extremidade superior, a cerca de 11% do PIB global (McKinsey Global Institute, 2015).

Porém, conjuntamente ao crescimento da IoT e aos benefícios atrelados a mesma, emergem novos desafios, como, por exemplo, regulamentações, privacidade, segurança, padronização e o alto volume de dados em trânsito (SANTOS et al., 2016), (BNDES, 2017), (SILVA FILHO, 2021). Para mais, os dados coletados pelos objetos IoT podem possuir imperfeições, advindas, por exemplo, da calibragem do sensor, e inconsistências, tais como bits fora de ordem, lacunas ou dados que se diferenciam drasticamente dos outros, denominados *outliers*.

1.1 JUSTIFICATIVA

Diversos estudos apresentaram propostas de implantações de cidades inteligentes com base em IoT focando em um objetivo específico, como, por exemplo, o monitoramento ambiental inteligente de áreas urbanas, caso do trabalho de Souza (2020) e tema escolhido para a presente dissertação. Conforme Mills (2007), este assunto representa um papel importante na identificação de tendências em mudanças no padrão de comportamento de variáveis ambientais.

Para mais, foram lançados vários projetos de *smart cities* em todo o mundo, tais como em Amsterdã (Holanda), Nice (França) e Padova (Itália), citadas por Talari (2017). Todos estes casos possuem uma melhora na qualidade de vida dos cidadãos e maior eficiência na infraestrutura e dia-dia do meio urbano, a partir do emprego da IoT e consequente monitoramento de parâmetros ambientais, como nível de monóxido de carbono (CO), temperatura e umidade do ar. Tais informações podem auxiliar na tomada de decisão dos gestores públicos, além de servir de alerta a população sobre possíveis fenômenos da natureza.

A IoT vem se tornando uma grande aliada na viabilização das cidades inteligentes, nas quais dispositivos como sensores e atuadores são componentes fundamentais para detectar e monitorar eventos vinculados ao meio ambiente, clima, energia, dentre outros (ZHANG et al., 2017). Neste contexto, destaca-se o monitoramento ambiental, visto que o acompanhamento do comportamento de parâmetros como gases poluentes, umidade e temperatura, dentre outros, podem propiciar informações essenciais para a saúde, segurança e qualidade de vida das pessoas, através da análise dos padrões de tais dados.

Nas cidades inteligentes, tem-se uma geração explosiva de dados, oriundos de diferentes fontes. Em vista disto, uma das principais dificuldades para a tomada de decisões e de ações a partir de dados obtidos pelos dispositivos IoT refere-se ao nível de confiança dos mesmos (SANTOS et al., 2016), devendo-se determinar quais são válidos e quais apresentam anormalidades. Talari (2017) concorda que um dos desafios no contexto de cidades inteligentes e monitoramento por sensores IoT é a confiabilidade dos dados obtidos. Se os dados não forem confiáveis, pode-se ter decisões e interpretações equivocadas com base nos mesmos.

Assim, tendo em vista que "a 'raison d'être' (razão de ser) da IoT é, justamente, a extração de conhecimento a partir dos dados coletados pelos seus sensores" (SANTOS et al., 2016), torna-se necessário que os algoritmos de processamento e análise dos dados tenham a capacidade de identificar dados discrepantes, isto é, que possivelmente apresentem erros. Neste cenário, um dos problemas encontrados, já citado anteriormente, é a presença de *outliers*.

Hawkins (1980) conceitua *outlier* como "uma observação que desvia tanto das outras observações que levanta suspeita de que foi gerada por um mecanismo diferenciado". De modo simplificado, *outliers* podem ser caracterizados como valores discrepantes ao se analisar a totalidade dos dados. Na maioria das vezes, estes são gerados por erros de medição ou de transmissão. Porém, segundo Souza (2020), *outliers* também podem indicar eventos de interesse, por exemplo, no contexto de monitoramento ambiental, altos níveis de poluição do ar, ruído ambiental e ilhas de calor, além de outras situações como fraude no cartão de crédito, intrusão cibernética ou atividade terrorista.

Na área da saúde, a detecção de anomalias nos dados de batimento cardíaco pode auxiliar a prever doenças cardíacas, sendo extremamente importante uma identificação correta. Ainda, pode-se ter uma previsão de acidentes de trânsito com base na observação de anormalidades nos padrões de tráfego. Outro motivo da necessidade de detecção de *outliers* refere-se a criação de modelos de aprendizado de máquina, os quais podem sofrer interferências devido a estas anomalias e gerar informações incorretas.

Os *outliers* podem ser o que mais dificulta a análise dos dados, mas ao mesmo tempo podem relatar acontecimentos que devem ser identificados. Para exemplificar possíveis consequências relacionadas a simples exclusão de *outliers* do conjunto de dados, sem uma análise mais detalhada dos mesmos, menciona-se um caso de grande impacto ecológico e relevância científica, citado por Rodrigues (2018). Este fato aconteceu no processamento dos dados oriundos do satélite Nimbus 7 da NASA (*National Aeronautics and Space Administration*), encarregado

de fornecer informações atmosféricas, dentre elas, monitorar dados relativos à camada de ozônio. Houve um atraso na descoberta de um buraco na camada de ozônio na Antártida por uma decisão inapropriada no tratamento de *outliers*. Apesar deste satélite ter iniciado suas atividades em 1978, o buraco só foi descoberto em 1985, através de dados coletados pelas bases de operação da *British Antarctic Survey*. Inclusive, os pesquisadores hesitaram em publicar seus resultados, visto que os dados do Nimbus 7, teoricamente mais precisos, não apontavam este problema. Com base neste relato, a NASA analisou mais minuciosamente seus dados, publicando, em 1986, evidências de observação do mesmo fenômeno. O buraco foi tão inesperado que durante vários anos os computadores que analisaram os dados do ozônio sistematicamente descartaram as leituras que deveriam ter apontado para o seu crescimento.

Consequentemente, as aplicações baseadas na detecção de anomalias são ilimitadas, sendo que esta torna-se essencial na observação e levantamento de dados. "Em um contexto no qual decisões são tomadas cada vez mais com base em dados, é de extrema importância garantir a veracidade dessas informações" (MAGRANI, 2018). Portanto, tendo em vista as diversas falhas de medição e de transmissão de dados, que podem levar a tomadas de decisões equivocadas, deve-se buscar a implementação de algoritmos que promovam uma maior confiabilidade dos dados obtidos pelos sensores IoT e que auxiliem na detecção de anormalidades, isto é, *outliers*.

1.2 OBJETIVOS

1.2.1 Objetivo Geral

Diante das informações até aqui dispostas, a proposta deste trabalho consiste na construção de um algoritmo multiclassificador para detecção de *outliers* em dados gerados por sensores IoT de monitoramento ambiental.

1.2.2 Objetivos específicos

Quanto aos objetivos específicos, pode-se citar:

– Realizar e expor uma pesquisa bibliográfica relativa aos principais conceitos utilizados;

– Implementar os algoritmos de detecção de *outliers* Zscore, Zscore Modificado e clusterização Kmeans em um mesmo conjunto de dados;

- Analisar os *outliers* detectados por cada um dos algoritmos, realizando uma comparação entre estes resultados;

- Calcular as métricas de desempenho individuais dos algoritmos;

- Construir um multiclassificador composto pela resposta dos três algoritmos;

- Calcular as métricas de desempenho do multiclassificador, comparando-as com as métricas individuais.

1.3 ORGANIZAÇÃO DO DOCUMENTO

Os capítulos posteriormente dispostos compõem a seguinte ordem: o Capítulo 2 expõe a fundamentação teórica essencial para embasar o trabalho proposto; o Capítulo 3 mostra os procedimentos metodológicos necessários para o desenvolvimento deste trabalho; o Capítulo 4 retrata as etapas de processamento e validação do sistema de detecção construído; o Capítulo 5 visa expor os resultados da implementação dos algoritmos e do multiclassificador, bem como a análise destes; e, por fim, o Capítulo 6 conclui este trabalho resumindo os resultados e trabalhos futuros relacionados.

2 FUNDAMENTAÇÃO TEÓRICA

Este Capítulo apresenta uma base de conhecimentos teóricos necessária para o entendimento deste trabalho. As seções iniciais apresentam conceitos acerca da Internet das Coisas (IoT - *Internet of Things*), com enfoque em cidades inteligentes, suas possíveis aplicações e o cenário brasileiro neste contexto. Em seguida, são elencadas as principais características de uma comunicação entre sensores IoT e os problemas mais usuais presentes nesta comunicação. Então, existe um maior detalhamento sobre os *outliers*, foco deste estudo, suas possíveis causas e os diferentes tipos existentes. Por fim, é descrito o funcionamento de algoritmos de detecção de *outliers*, enfatizando os utilizados neste trabalho, Zscore, Zscore Modificado e K-Means, além de descrição de métricas de avaliação destes algoritmos e a definição de um multiclassificador.

2.1 INTERNET DAS COISAS

O termo IoT foi proposto por Kevin Ashton em 1999, o qual, dez anos depois, escreveu o artigo "A Coisa da Internet das Coisas" para o RFID Journal (Techtudo, 2014). Ao longo do tempo, surgiram diversas definições para o termo. Conforme Chaudhary (2019), a IoT é um conceito de rápida evolução, sendo que sua ideia primária é interconectar múltiplos dispositivos eletrônicos analógicos e digitais, homogêneos e heterogêneos por natureza e com alcance de transmissão sobreposto entre si. Desse modo, os objetos IoT podem comunicar as informações de maneira eficiente, com a mensagem sendo entregue instantaneamente entre os nós vizinhos na rede.

A solução IoT é estruturada em três grandes componentes: *hardware*, *middleware* e interface. O *hardware* é composto por sensores e atuadores, responsáveis por, respectivamente, coletar informações do mundo físico e processá-las a fim de afetar este mesmo ambiente. Este primeiro componente se comunica com o *middleware*, que tem a função de salvar, processar e analisar os dados, para torná-los apresentáveis ao usuário final através da interface (visualização). Os três elementos se comunicam por meio de protocolos de comunicação (SILVA FILHO, 2021).

Tal estruturação propicia uma grande gama de novas aplicações, por exemplo, sensoriamento de ambientes de difícil acesso e inóspitos, monitoramento do trânsito e de variáveis ambientais, coleta de dados de pacientes em tempo real, entre outras. O principal objetivo da

IoT é facilitar o dia-a-dia e melhorar a qualidade de vida das pessoas, simplificando o uso de certos elementos e possibilitando a automação de atividades.

2.1.1 Cidades inteligentes

A IoT é fundamental ao conceito de cidades inteligentes, projetos urbanísticos inovadores, que tem o objetivo de aumentar a qualidade de vida dos cidadãos, baseando-se em preceitos de sustentabilidade e eficiência. Assim, busca-se corrigir problemas decorrentes da urbanização acelerada e melhorar processos atuais a fim de continuar em expansão e desenvolvimento sustentável.

2.1.1.1 Possíveis aplicações

Eis alguns exemplos de setores que a IoT pode contribuir em uma cidade inteligente, conforme Santos (2016) e Talari (2017):

- Redes elétricas inteligentes (*smart grids*): possibilitam maior controle e melhora no serviço de consumo de energia elétrica. Medidores inteligentes de energia coletam diversas leituras, as quais podem ser utilizadas pelas concessionárias de energia para controlar os recursos de modo proporcional ao consumo e prever possíveis falhas na rede elétrica, elevando a eficiência e qualidade dos serviços;

- Mobilidade urbana: semáforos inteligentes que ajustam o sinal conforme o tráfego; sistemas de trânsito com informações em tempo real sobre horários, localização e pontos de acesso de transporte público; automação de estacionamento, com reservas de vagas, pagamento automatizado e aplicativos de busca remota de vagas; informações sobre acidentes, engarrafamentos, entre outros acontecimentos para otimizar a fluidez de vias e melhorar a segurança;

- Gestão de recolhimento de resíduos: sensores em lixeiras que enviam notificação de onde existe resíduo a ser recolhido podem otimizar a coleta de lixo e propiciar economia de combustível do caminhão de recolhimento;

- Monitoramento ambiental: sensores podem avaliar a qualidade do ar e monitorar os índices de poluição ambiental, enviando os dados mais relevantes aos moradores e identificando fontes de poluição. Além disso, a observação de variáveis como temperatura, umidade, irradiação solar e velocidade do vento podem auxiliar na detecção de tempestades e disparar alertas para evacuação de áreas vulneráveis;

- Sistemas de emergência inteligentes: monitoramento em tempo real da segurança pú-

blica por câmeras e aplicativos de notificações coletivas, facilitando o acionamento e localização de viaturas de polícia, serviços de saúde, bombeiros e outros serviços de urgência.

2.1.1.2 Exemplos de cidades inteligentes

A primeira cidade a implementar o conceito de cidade inteligente foi Songdo, na Coreia do Sul, em 2001. Atualmente conhecida como a "cidade mais inteligente do mundo", seus edifícios são conectados a sistemas que possibilitam monitorar a energia e os alarmes de incêndio, minimizando os custos com manutenção e otimizando o uso. Também, há um sistema pneumático que envia o lixo das casas diretamente para um aterro subterrâneo, sem a necessidade de caminhões circulando pelas ruas. Os detritos são utilizados para abastecer incineradores que geram energia para a cidade.

Em Amsterdã, na Holanda, foram lançados vários projetos em 2006 visando torná-la uma cidade inteligente. Estes incluíram iluminação pública LED com controladores inteligentes de modo a reduzir o consumo, podendo gerar economia de energia de até 80%, além de relato automático e remoto de falhas de energia.

Na cidade de Nice, na França, quatro serviços inteligentes foram estabelecidos. Foram implantados iluminação e circulação inteligentes, gerenciamento inteligente de resíduos e monitoramento ambiental inteligente.

Um exemplo de cooperação privada e pública ocorre em Padova, na Itália. O município fornece a infraestrutura e orçamento necessários e a Universidade de Padova implementa o conceito de cidade inteligente. Sensores colocados em postes de iluminação pública e conectados à internet medem a intensidade da luz, ajustando a operação do sistema de iluminação, e coletam dados ambientais, como nível de CO, temperatura, umidade do ar, vibrações e ruído.

2.1.2 Cenário brasileiro

O mercado de Tecnologia da Informação e Comunicação (TIC) do Brasil apresenta relevância mundial, tendo movimentado US$ 158 bilhões em 2014, equivalendo ao posto de 5º maior do mundo, segundo a Associação Brasileira das Empresas de Software (ABES) e a *International Data Corporation* (IDC). Este segmento corresponde a cerca de 9,1% do PIB e, conforme estimativas da Brasscom, esse percentual irá se elevar para 10,7% até 2022 (BNDES, 2017).

Existem diversos fóruns e esforços brasileiros para discutir e compreender o tema IoT,

como a Câmara de Gestão e Acompanhamento do Desenvolvimento de Sistemas de Comunicação Máquina a Máquina (Câmara M2M), o Fórum Brasileiro IoT, o GT Interministerial de Cidades Inteligentes, entre outros (BNDES, 2017). Destas e outras discussões, resultou o Plano Nacional de Internet das Coisas, instituído pelo Decreto nº 9.854, em 25 de junho de 2019. Este Plano tem o objetivo de "implementar e desenvolver a Internet das Coisas no País e, com base na livre concorrência e na livre circulação de dados, observadas as diretrizes de segurança da informação e de proteção de dados pessoais" (BRASIL, 2019).

Além disso, no dia 8 de dezembro de 2020, durante o *Smart City Session 2020*, evento global online voltado a discussão das tendências sobre cidades inteligentes, foi lançada oficialmente a Carta Brasileira para Cidades Inteligentes. Este documento, elaborado de forma integrada pelo Governo Federal, sociedade civil, academia e setor privado, baseia-se nos princípios da Política Nacional de Desenvolvimento Urbano (PNDU) e visa nortear a elaboração de políticas públicas, a implementação e o financiamento de projetos de cidades inteligentes pelos municípios (BRASIL, 2020).

Neste contexto, as cidades inteligentes no Brasil estão avançando aos poucos. Em ranking das 165 principais cidades inteligentes do mundo, divulgado em 2020 pelo IESE Bussiness School, escola de negócios da Universidade de Navarra (Espanha), o Brasil ocupa seis posições (IESE Business School, 2020). As cidades brasileiras presentes no ranking e suas respectivas posições são: São Paulo (123ª), Rio de Janeiro (132ª), Brasília (135ª), Curitiba (138ª), Belo Horizonte (156ª) e Salvador (157ª).

Exemplificando alguns casos brasileiros desta lista, a capital paulista é destaque em virtude dos investimentos em mobilidade urbana, com a construção de mais ciclofaixas e corredores de ônibus. Na mesma perspectiva, Curitiba inovou através da criação do Ecoelétrico, frota de carros elétricos que prestam serviços públicos, além da implementação de uma frota de ônibus híbridos, movidos à eletricidade e biocombustível. Também, existe um aplicativo pelo qual é possível solicitar a coleta de entulhos, manutenções em ruas e calçadas e consultar em tempo real horários, rotas e localizações de ônibus. Em Salvador, além dos investimentos para melhorar a mobilidade urbana, tem-se uma atenção especial voltada ao monitoramento da iluminação de locais públicos, existindo uma redução no consumo de energia e mais rapidez na manutenção de equipamentos.

Destaca-se ainda outra localidade brasileira, que, apesar de não estar no ranking citado, possui grande relevância devido principalmente ao seu propósito. Em Croatá, no Ceará, está

sendo desenvolvido um projeto pelo grupo italiano Planet Idea de construção da primeira cidade inteligente do mundo voltada para a habitação social. Criada para moradores de baixa renda, a cidade vai possuir coleta de lixo inteligente, sistemas de reaproveitamento da água das chuvas, fiação elétrica subterrânea, irrigação automatizada conforme o clima e sistemas de compartilhamento de bicicletas e de carros (Planet Smart City, 2017).

2.2 COMUNICAÇÃO ENTRE SENSORES IOT

No contexto de IoT, deve-se priorizar a comunicação sem fio entre os sensores, dado os custos elevados de cabeamento de milhões de (MAGRANI, 2018), além da maior flexibilidade proporcionada, sendo possível a alocação de sensores em locais de difícil acesso e o sensoriamento remoto em larga escala. Uma Rede de Sensores Sem Fio (RSSF) é composta por nós de sensores sem fio que englobam uma interface de rádio, um conversor analógico-digital (ADC), vários sensores, memória e uma fonte de alimentação. Os nós sensores de uma RSSF devem ser extremamente pequenos, de custo reduzido e com baixo consumo de energia, podendo ser aplicados em qualquer ambiente (MAGRANI, 2018).

As RSSF são uma tecnologia notável para o desenvolvimento de aplicações nas mais diversas áreas, podendo fornecer recursos de medição, inferência e compreensão de índices ambientais, logísticos, de saúde, dentre outros. Para tanto, o principal desafio citado por Talari (2017) é "como processar as informações em larga escala dos sensores sobre as restrições de energia e rede e diferentes tipos de incerteza".

2.2.1 Principais problemas na comunicação IoT

Os sensores usados na maior parte de infraestruturas IoT são de baixo custo e, como resultado, são suscetíveis a falhas. Alguns dos principais problemas envolvidos na comunicação IoT elencados por (SANTOS et al., 2016), os quais podem dificultar a análise dos dados obtidos, são descritos brevemente a seguir:

- Lacunas nos dados: correspondem a vazios, isto é, dados faltantes no conjunto de dados, ocorridos por diferentes fatores. O caso mais simples é a lacuna originada por uma falha esporádica na operação dos sensores, por motivos desvinculados do fenômeno monitorado, causando uma lacuna aleatória (MAR - *missing at random*). Outros casos são as lacunas que não são geradas aleatoriamente (MNAR - *missing not at random*). Estas são geradas, por exemplo,

quando um sensor deixa de coletar dados no momento que determinada variável monitorada atinge certo valor ou quando um usuário desliga o sensor por algum motivo.

- Diferença de granularidade: em razão da grande quantidade de sensores heterogêneos, surgem diferenças de amostragem entre cada sensor. Por exemplo, para um mesmo fenômeno em um mesmo intervalo de tempo, é possível ter um sensor coletando dados a cada segundo e outro a cada minuto.

- Imprecisões: podem surgir algumas inconsistências no conjunto de dados, como, por exemplo, bits fora de ordem.

- *Outliers*: problema foco deste trabalho e que será melhor detalhado a seguir.

2.3 OUTLIERS

"Um *outlier* é um elemento que desvia de um padrão do conjunto de dados ao qual ele pertence" (SILVA FILHO, 2021). Esta é uma definição bem simplificada para esta inconsistência, sendo necessário realizar algumas considerações. Um *outlier* sempre pertence a um conjunto e podem existir outros *outliers* neste mesmo conjunto. Além disso, um elemento é considerado *outlier* sempre em relação a um padrão. Sendo assim, um mesmo elemento pode ser dito *outlier* em relação ao padrão A e não ser um *outlier* em relação ao padrão B. Também, estes valores discrepantes aparecem em pequena quantidade no conjunto ao qual pertencem, pois se existissem muitos *outliers* seria impossível estabelecer um padrão (SILVA FILHO, 2021).

2.3.1 Possíveis causas

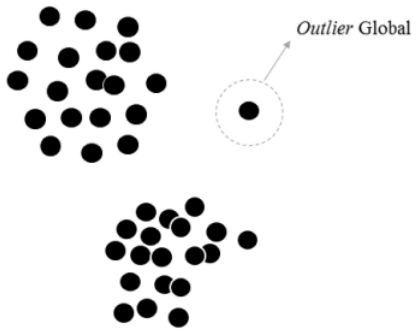
As anomalias encontradas nos dados obtidos pelos sensores IoT são originadas de diversas fontes. Algumas das causas são uma grande quantidade de ruído, erro de calibração, conexão ou hardware, bateria fraca ou ambiente fora da faixa de medição do sensor (USBERT et al., 2021). Outros possíveis motivos da presença de *outliers* são erros de amostragem, dados misturados com outros, erros de medição causados pelo aplicativo ou sistema, além de desvios naturais nos dados, os quais indicam fraude ou alguma outra anormalidade que se pretende detectar.

2.3.2 Tipos de outliers

Em relação aos tipos de *outliers*, Souza (2020) expõe a seguinte categorização:

- *Outlier* global (ou pontual): caso em que um único elemento difere da normalidade dos dados, como apresentado na Figura 2.

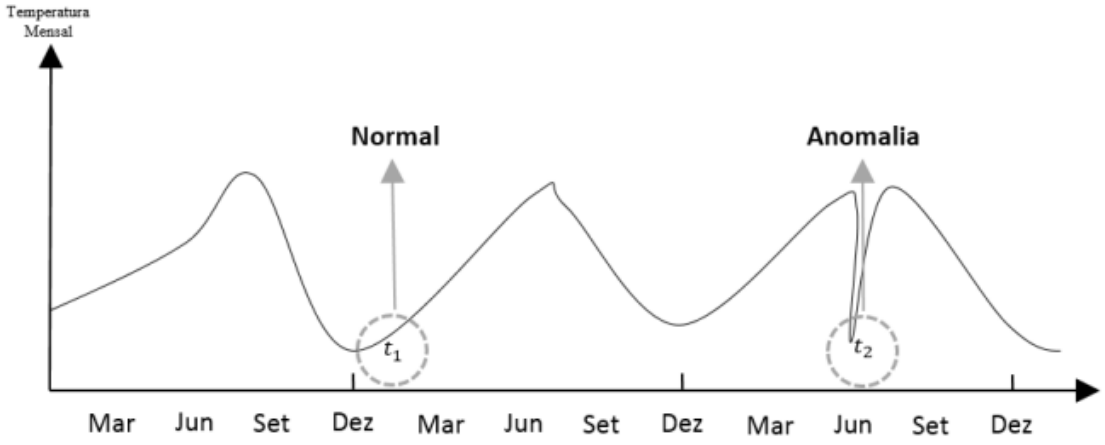
Figura 2 – Ilustração de um *outlier* global



Fonte: (FREITAS, 2019).

- *Outlier* contextual (ou condicional): quando uma instância de dados é anômala apenas em um determinado contexto. Conforme pode ser visualizado no exemplo da Figura 3, que retrata um local do Hemisfério Norte, uma temperatura próxima de zero é normal no inverno (instante t_1), mas trata-se um valor anormal no verão (instante t_2).

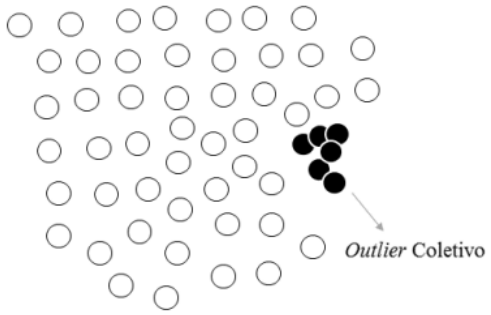
Figura 3 – Ilustração de um *outlier* contextual



Fonte: (FREITAS, 2019).

- *Outliers* coletivos: ocasião em que um subconjunto dos dados é considerado anormal em relação ao conjunto inteiro. A Figura 4 retrata um exemplo deste tipo de *outlier*.

Figura 4 – Ilustração de um *outlier* coletivo



Fonte: (FREITAS, 2019).

2.4 ALGORITMOS DE DETECÇÃO DE OUTLIERS

Um detector de anomalias padrão recebe como entrada um conjunto de dados e classifica-os em normais ou anormais, conforme o comportamento esperado. Os métodos de detecção de *outliers* podem ser divididos em três tipos de aprendizagem: supervisionada, semi-supervisionada e não supervisionada (FREITAS, 2019), (SILVA FILHO, 2021), (SOUZA, 2020).

- Aprendizagem supervisionada: o algoritmo tem como entrada um conjunto de dados já rotulados, isto é, definidos em normais e anormais. Tem como objetivo treinar um classificador a fim de torná-lo capaz de classificar novos dados não rotulados.

- Aprendizagem semi-supervisionada: o classificador recebe um conjunto de dados que contém somente dados rotulados como normais, sem a necessidade de rótulos para a classe dos *outliers*.

- Aprendizagem não supervisionada: o algoritmo conta apenas com dados não rotulados e busca extrair padrões destes dados, necessitando de análise para perceber o significado de tais padrões. Assume-se que os dados normais são mais frequentes que as anormalidades. Sem precisar de dados de treinamento, tal aprendizagem apresenta uma maior aplicabilidade.

2.4.1 Natureza dos dados

Na escolha das técnicas de detecção de *outliers* a serem empregadas em determinado conjunto de dados, o principal fator que deve ser considerado é a natureza dos dados. Os dados obtidos podem apresentar natureza univariada, multivariada ou multidimensional. Dados univariados correspondem a amostras de um mesmo fenômeno escalar, por exemplo, monitoramento apenas de umidade. Dados multivariados representam amostras referentes à diferentes fenô-

menos, como o monitoramento ao mesmo tempo de umidade, temperatura, iluminação, dentre outros. Dados multidimensionais caracterizam amostras relacionadas a diversos fenômenos e situados, por exemplo, no espaço, isto é, podem representar mais de duas dimensões. Estes dados podem ser retratados por arranjos multidimensionais, denominados tensores (SOUZA, 2020).

2.4.2 Saída dos classificadores

Na detecção de *outliers*, as saídas produzidas pelos classificadores normalmente podem ser de dois tipos descritos a seguir (FREITAS, 2019), (SOUZA, 2020):

- Rótulos (*Labels*): rótulos binários, indicando a qual classe (normal ou anormal) pertence cada amostra;

- Pontuações (*Scores*): pontuações que indicam quão anormais são as amostras daquele conjunto. Classificador retorna uma lista com *outliers* ranqueados.

As técnicas baseadas em *scores* possibilitam ao analista de dados maior liberdade na definição dos *outliers*, tornando possível o uso de um limite específico a fim de selecionar as anomalias mais representativas à pesquisa. Apesar das técnicas baseadas em *labels* poderem ser adaptadas para tal, as com base em *scores* são mais flexíveis.

2.4.3 Métodos estatísticos

De acordo com Usbert (2021), as técnicas de detecção de *outliers* baseadas em estatística são consideradas melhores em redes de sensores de baixa capacidade em razão da sua escalabilidade, baixo custo computacional e o fato de não necessitarem conhecimento prévio sobre os dados. Métodos estatísticos presumem que os dados de entrada seguem um modelo estocástico, isto é, as variáveis respondem a uma distribuição específica. Desse modo, para cada parâmetro, é aplicado um teste de inferência estatística com o objetivo de determinar se o mesmo pertence ao modelo, ou seja, se é normal ou não (SOUZA, 2020).

A seguir, tem-se uma explicação mais detalhada sobre os dois métodos estatísticos utilizados neste estudo, Zscore e Zscore Modificado, além de descrição sobre uma distribuição normal dos dados, hipótese considerada pelo Zscore.

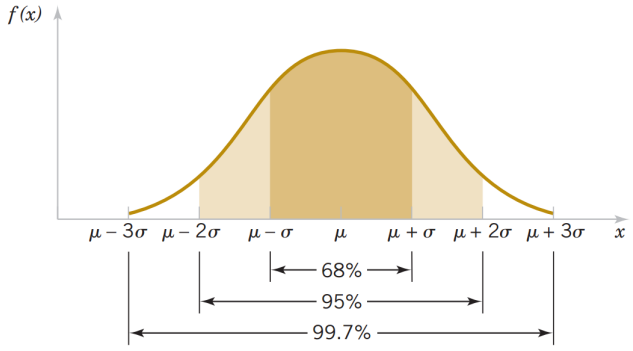
2.4.3.1 Distribuição normal (gaussiana)

A distribuição normal, ou gaussiana, é a mais conhecida distribuição de probabilidade e uma das mais importantes em estatística. Ela possui dois parâmetros: a média (μ), onde está centralizada, e a variância ($\sigma^2 > 0$), que denota o seu grau de dispersão. A dispersão também pode ser indicada em termos de unidades padrão, através do desvio padrão (σ). Dependendo destes parâmetros, tem-se diferentes distribuições normais, cuja curva é traçada conforme a função $g(x)$, dada pela Equação 2.1 (UFSC, 2018):

$$g(x) = \frac{e^{(\frac{-0.5 \cdot (x-\mu)^2}{\sigma^2})}}{\sigma \cdot \sqrt{2\pi}} \tag{2.1}$$

Um exemplo do gráfico da função $g(x)$ pode ser visualizado na Figura 5.

Figura 5 – Exemplo de curva de distribuição normal (gaussiana)



Fonte: (UFSC, 2018).

A distribuição normal apresenta um formato simétrico em torno da média, o que significa que a média, a mediana e a moda são todas coincidentes. Ela tem a forma similar a "curva de um sino", rapidamente decaindo para $\pm\infty$. O pico da curva corresponde ao valor médio dos dados e a área total abaixo da curva é unitária. Pode-se observar também na Figura 5, a marcação de pontos no eixo x e o quanto desviados eles estão em relação a média, além do percentual correspondente da área total da curva englobada por estes pontos.

2.4.3.2 Zscore

O Zscore é um método simples que detecta *outliers* a partir do cálculo de quantos desvios padrões um determinado dado está em relação ao valor médio do conjunto. Este algoritmo se baseia na propriedade de que o conjunto de dados segue uma distribuição normal, sendo mais

eficiente nestes casos. Sua métrica é definida como a diferença absoluta entre o valor do dado x_i, em que $i = 1, 2, 3, ...N$, sendo N o número total de vetores no conjunto, e a média do conjunto, normalizada pelo desvio padrão, conforme Equação 2.2 (FETTERMANN et al., 2015):

$$Z_{score} = \frac{x_i - \mu}{\sigma} \tag{2.2}$$

Um limite de corte sugerido pela literatura para que um dado seja considerado *outlier* é de um valor absoluto de Z_{score} maior do que 3 (CAMPOS, 2015), (FETTERMANN et al., 2015). Entretanto, dependendo das características do conjunto de dados ou dos critérios de decisão estabelecidos, tal valor pode ser alterado (FETTERMANN et al., 2015).

Este classificador traz como principal vantagem a sua fácil implementação, visto que necessita apenas da média e do desvio padrão do conjunto de dados. Porém, possui alguns problemas, tais como perda de eficiência em dados que não são normalmente distribuídos e um comportamento impreciso em bases de dados pequenas. Outra limitação do algoritmo, citada por Fettermann (2015), é o desvio padrão elevado devido aos dados com valores extremos. Em vista disto, os *outliers* menos extremos podem não ser detectados em razão dos *outliers* mais extremos.

2.4.3.3 Zscore Modificado

Para minimizar as limitações do algoritmo Zscore, foi desenvolvido o método denominado Zscore Modificado. A média e o desvio padrão são substituídos por indicadores mais robustos, isto é, menos sensíveis a *outliers*: a mediana M_x e o desvio absoluto da mediana (MAD), calculado conforme Equação 2.3 (NASCIMENTO et al., 2012), em que *median* representa a função mediana. Nas equações 2.3 e 2.4 a seguir, x_i representa o conjunto de dados, sendo $i = 1, 2, 3, ...N$, onde N é o número total de vetores no conjunto.

$$MAD = median\{|x_i - M_x|\} \tag{2.3}$$

Os valores são multiplicados por uma constante igual a $0,6745$ (NASCIMENTO et al., 2012), obtendo-se a métrica M_i do Zscore Modificado, dada pela Equação 2.4.

$$M_i = \frac{0,6745}{MAD}(x_i - M_x) \tag{2.4}$$

Os dados serão identificados como outliers quando $|M_i| > K$ (NASCIMENTO et al.,

2012), sendo K um valor selecionado pelo pesquisador (FETTERMANN et al., 2015). Iglewics e Hoaglin (1993) sugerem que os pontos considerados *outliers* apresentem $|M_i| > 3,5$.

2.4.4 Clusterização

As técnicas de clusterização, também chamadas de técnicas de agrupamento, buscam dividir um determinado conjunto de dados em *clusters* (aglomerados), de forma que as distâncias dentro de um mesmo grupo sejam minimizadas e as distâncias entre *clusters* sejam maximizadas. Além disso, cada um dos dados é classificado exclusivamente em um único *cluster*. Trata-se de um método de aprendizagem não-supervisionado, uma vez que os rótulos dos dados não são utilizados.

Para a detecção de *outliers*, pode-se observar os dados destoantes, que não se classificam em nenhum dos *clusters*. Isto é feito após a divisão de todo o conjunto de dados em *clusters*, por meio da determinação de um limiar máximo do quão distantes os dados devem estar dos centroides, da densidade máxima, ou de outra característica referente ao método de clusterização, dos *clusters* de modo a serem considerados *outliers*.

2.4.4.1 K-Means

Existem vários algoritmos de clusterização, sendo que o mais utilizado é o K-Means. Este algoritmo tem o objetivo de dividir os dados em k clusters, sendo k o número de clusters definido pelo pesquisador. Esta divisão é feita a partir da determinação de centroides, locais imaginários ou reais, que representam o centro de cada *cluster*. Inicialmente, a localização dos centroides é gerada aleatoriamente e os k agrupamentos são criados associando cada um dos dados ao centroide mais próximo. Então, tem início um processo iterativo, no qual o novo centroide de cada *cluster* se torna a média dos dados atribuídos para aquele *cluster* repetidas vezes, até convergir.

O algoritmo básico do K-Means pode ser resumido em:

1. Definir k centroides iniciais de modo aleatório;
2. Criar k *clusters*, atribuindo cada dado ao centroide mais próximo;
3. Recalcular o centroide de cada *cluster*;
4. Repetir até que os centroides não se alterem mais ou se alterem pouco.

Para determinar a proximidade dos dados aos centroides, existem diferentes métricas possíveis. A mais utilizada é a distância euclidiana. A distância d entre um ponto $P = (p_1, p_2)$

e um ponto $Q = (q_1, q_2)$ é dada pela Equação 2.5 (PARKER, 2011):

$$d = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2} \tag{2.5}$$

Para pontos em um espaço com N dimensões, a fórmula da distância euclidiana é generalizada conforme a Equação 2.6 (PARKER, 2011), na qual $i = 1, 2, 3, ...N$.

$$d = \sqrt{\sum_{i=1}^{N} (p_i - q_i)^2} \tag{2.6}$$

2.4.5 Efeitos de *masking* e *swamping*

Na presença de mais que um *outlier* no conjunto analisado, os algoritmos de detecção de *outliers* estão sujeitos aos efeitos de *masking* e *swamping*. O efeito de *masking* (camuflagem) ocorre quando um *outlier* é mascarado pela presença de outros, isto é, são identificados menos *outliers* do que existem realmente. O efeito de *swamping*, problema oposto ao anterior, acontece quando os *outliers* fazem com que algumas observações normais sejam consideradas anormalidades, identificando-se mais *outliers* do que os realmente existentes (MARQUES, 2015).

2.4.6 Métricas de desempenho dos algoritmos

Ao empregar um algoritmo de detecção de *outliers*, são obtidos quatro tipos de resultados (COSTA, 2014), (SILVA FILHO, 2021), (SOUZA, 2020):

- Verdadeiros Positivos (VP): número de *outliers* corretamente identificados;

- Falsos Positivos (FP): número de dados normais incorretamente identificados como *outliers*;

- Verdadeiros Negativos (VN): número de dados normais corretamente identificados;

- Falsos Negativos (FN): número de *outliers* incorretamente identificados como dados normais.

A partir destas características, é possível calcular várias métricas de desempenho usadas para avaliar os algoritmos. Algumas das mais empregadas e que serão calculadas neste estudo são: sensibilidade, precisão, especificidade e acurácia.

A sensibilidade ou taxa de verdadeiros positivos (TVP) mede a porcentagem de acerto do algoritmo na identificação dos *outliers*. É dada pela Equação 2.7 (COSTA, 2014), (SOUZA,

2020):

$$TVP = \frac{VP}{VP + FN} \tag{2.7}$$

A precisão (P) estabelece a quantidade de acertos de uma determinada classe com base nas predições feitas. Resulta em dois indicadores: P1, calculado conforme a Equação 2.8, define a porcentagem de anomalias preditas corretamente em relação ao total de anomalias preditas; e P2, obtido pela Equação 2.9, que denota a porcentagem de dados normais preditos corretamente em relação ao total de dados normais preditos (COSTA, 2014).

$$P_1 = \frac{VP}{VP + FP} \tag{2.8}$$

$$P_2 = \frac{VN}{VN + FN} \tag{2.9}$$

A especificidade (E) indica a porcentagem de acerto do algoritmo em relação aos casos negativos, isto é, na classificação dos dados normais, sendo estabelecida pela Equação 2.10 (COSTA, 2014):

$$E = \frac{VN}{VN + FP} \tag{2.10}$$

Por fim, a acurácia (A), definida pela Equação 2.11, denota a porcentagem total de predições corretas, tanto de *outliers* quanto de dados normais (COSTA, 2014), (SILVA FILHO, 2021).

$$A = \frac{VP + VN}{VP + VN + FP + FN} \tag{2.11}$$

2.4.7 Multiclassificador

Existe uma grande variedade de métodos de detecção de *outliers*, não sendo possível determinar um único algoritmo como sendo o melhor em todas as circunstâncias. Cada classificador tem seus pontos fortes e fracos. Uma maneira de aproveitar esta diversidade é aplicar mais de um método ao mesmo conjunto de dados e construir um algoritmo que mescle os resultados, denominado de multiclassificador. Isto deve apresentar resultados mais satisfatórios do que qualquer método individual, pois trata-se de um sistema mais robusto. Em uma situação

ideal, as fraquezas devem mais ou menos se anular ao invés de se reforçar, fornecendo altas taxas de detecção (PARKER, 2011).

Um multiclassificador deve lidar com três importantes questões (PARKER, 2011):

- A resposta do multiclassificador deve ser a melhor dados os resultados dos classificadores individuais. É necessário representar, de alguma maneira lógica, a classificação verdadeira mais provável, ainda que se tenham classificações individuais contraditórias;

- Os possíveis diferentes tipos de resposta gerados pelos classificadores individuais devem ser unidos em uma única resposta coerente;

- O multiclassificador deve fornecer o resultado correto mais frequentemente do que qualquer um dos classificadores individuais, caso contrário, a mesclagem não faria sentido.

3 METODOLOGIA

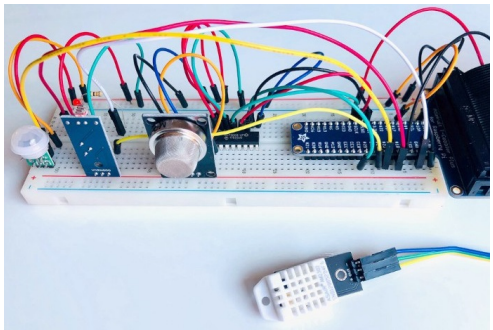
Este estudo teve início com uma revisão bibliográfica sobre a IoT, com foco em cidades inteligentes, a fim de se obter um maior entendimento acerca do assunto, além de analisar trabalhos relacionados já realizados e, assim, obter ideias do que poderia ser feito. Neste Capítulo, será abordado o conjunto de dados utilizado para análise e a lógica de implementação das técnicas de detecção de *outliers* estabelecidas.

3.1 CONJUNTO DE DADOS

O conjunto de dados utilizado para as análises foi obtido da plataforma do Google denominada Kaggle (Kaggle, 2021), a qual foi criada em 2010 e permite a realização de várias atividades relacionadas as áreas de ciência de dados e aprendizado de máquina (*machine learning*). Através dela, o usuário tem acesso a um ambiente online gratuito em que são disponibilizados diversos conjuntos de dados, além de fóruns de discussão e códigos nas linguagens Phyton e R.

De acordo com a aplicação definida para este estudo, escolheu-se um conjunto que reúne dados de telemetria de sensores ambientais, publicado na plataforma Kaggle por Gary Stafford em 20 de julho de 2020 (Gary Stafford, 2020a). Os dados foram coletados a partir de três dispositivos IoT reais, cada um composto por quatro sensores conectados a um microcontrolador Raspberry Pi e montado sobre uma placa de ensaio (protoboard), conforme apresentado na Figura 6.

Figura 6 – Exemplo de dispositivo IoT utilizado na obtenção dos dados



Fonte: (Gary Stafford, 2020a).

Cada dispositivo inclui os seguintes sensores:

1. Sensor de qualidade do ar e de detecção de gases perigosos, neste caso, monóxido de

carbono (CO), gás de petróleo liquefeito (LPG - *liquid petroleum gas*) e fumaça;

2. Sensor digital de temperatura e umidade;

3. Sensor de movimento infravermelho piroelétrico;

4. Sensor fotossensível de detecção de intensidade de luz.

Os nomes dos dispositivos IoT, designados por Stafford (2020) conforme segue, foram encurtados:

- "b8:27:eb:bf:9d:51": denominado de D1;

- "00:0f:00:70:91:0a": chamado de D2;

- "1c:bf:ce:15:ec:4d": intitulado de D3.

Estes dispositivos foram colocados propositalmente em ambientes físicos que variam em temperatura, umidade e outras condições ambientais. O dispositivo D1 foi alocado em um local de condições estáveis, mais quente e seco, enquanto que D2 foi localizado em um ambiente também de condições estáveis, mas mais frio e úmido. Já o dispositivo D3 foi posicionado em um meio de temperatura e umidade altamente variáveis.

Cada dispositivo IoT coletou sete leituras diferentes dos quatro sensores em intervalos regulares. As leituras dos sensores abrangem temperatura, umidade, CO, LPG, fumaça, luz e movimento. Estas foram realizadas de 12/07/2020 00:00:00 UTC (*Coordinated Universal Time* - Tempo Universal Coordenado) a 19/07/2020 23:59:59 UTC. As leituras do sensor, juntamente com um ID do dispositivo e carimbo de data/hora, foram publicados como uma única mensagem, utilizando o protocolo de rede MQTT (*Message Queuing Telemetry Transport* - Transporte de telemetria de enfileiramento de mensagens). O arquivo de leituras é composto por 405.184 linhas de dados e 9 colunas, cuja descrição pode ser visualizada na Tabela 1.

Tabela 1 – Descrição das colunas do conjunto de dados

Coluna	**Descrição**	**Unidade de medida**
ts	momento da leitura	data/hora
device	nome do dispositivo	*string* (frase)
co	monóxido de carbono	partes por milhão (%)
humidity	umidade	percentual (%)
light	detecção ("*true*") ou não ("*false*") de luz	*boolean* (variável booleana)
lpg	gás de petróleo liquefeito	partes por milhão (%).
motion	detecção ("*true*") ou não ("*false*") de movimento	*boolean* (variável booleana)
smoke	concentração de fumaça	partes por milhão (%)
temp	temperatura	Fahrenheit (F)

Fonte: Adaptado de (Gary Stafford, 2020b)

Para as análises, foram consideradas somente cinco das sete características: CO, umidade, LPG, fumaça e temperatura. Isso porque os parâmetros de luz e movimento só retornam "verdadeiro" (*true*) ou "falso" (*false*), para quando existe ou não luz ou movimento, respectivamente.

3.1.1 Separação dos dados de cada dispositivo

Do Kaggle, foi obtido um arquivo de extensão csv tabulado por vírgulas denominado "iot_telemetry_data". No Excel, essa tabulação foi convertida para separação por colunas e os valores das variáveis foram multiplicados por 1 para serem reconhecidos como valores numéricos, e não como caracteres. Também, conforme já comentado, os nomes dos dispositivos foram substituídos por D1, D2 e D3, a fim de facilitar a referência destes. O novo arquivo de extensão csv foi chamado de "iot_teste2".

Os dados deste novo arquivo foram lidos em uma tabela e, assim, foi possível comparar as *strings* da coluna *device* da tabela com os nomes D1, D2 e D3, armazenando os dados de cada dispositivo nas matrizes "dados_D1", "dados_D2" e "dados_D3", respectivamente. Também, foram excluídas as colunas desnecessárias às análises, deixando somente aquelas correspondentes as cinco características que se deseja observar a presença de *outliers*.

3.1.2 Análise da distribuição dos dados

Buscou-se obter um entendimento sobre a distribuição dos dados para determinar uma metodologia de análise. Para tal, foram determinados a média μ e o desvio padrão σ de cada característica para os três dispositivos. Os resultados estão resumidos na Tabela 2.

Tabela 2 – Média e desvio padrão das características dos dispositivos D1, D2 e D3

Característica	μ D1	μ D2	μ D3	σ D1	σ D2	σ D3
CO	0.0056	0.0035	0.0042	0.00056	0.0015	0.00032
Umidade	50.8141	75.4444	61.9102	1.8889	1.9758	8.9448
LPG	0.0083	0.0059	0.0068	0.0006	0.0017	0.00037
Fumaça	0.0223	0.0155	0.0179	0.0017	0.0048	0.0011
Temperatura	22.2800	19.3626	26.0255	0.4819	0.6438	2.0264

Fonte: elaborado pela autora (2021).

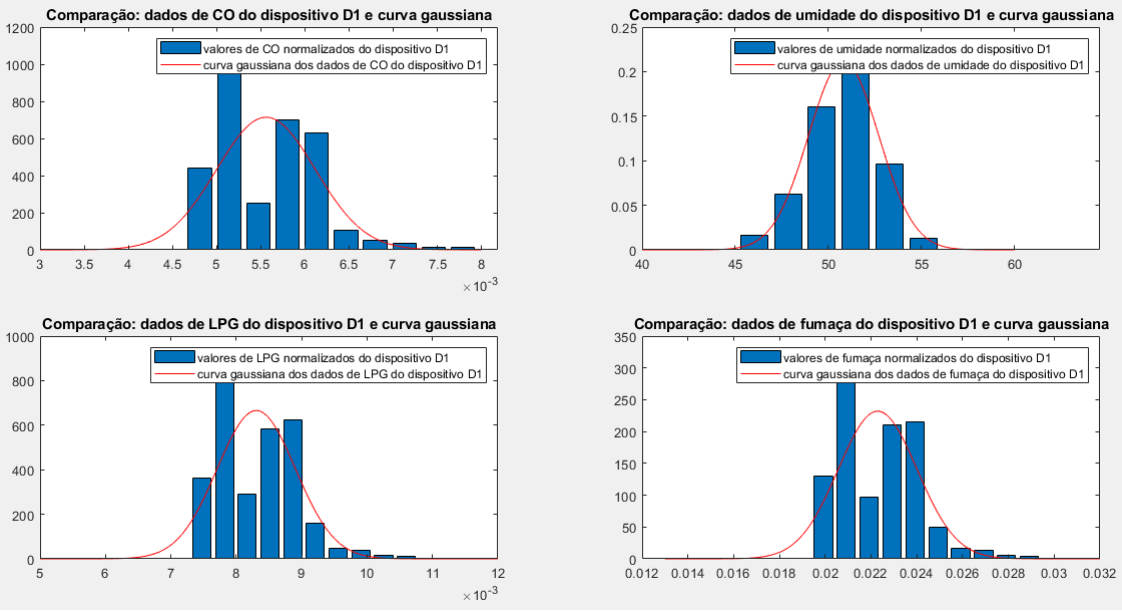
Analisando os valores da Tabela 2, constata-se que o dispositivo D1 apresenta as maiores médias de concentração de gases poluentes (CO, LPG e fumaça), enquanto que o dispositivo D2

possui os maiores desvios padrões destas características. Quanto aos parâmetros de temperatura e umidade, é possível observar que o dispositivo D1 está inserido em um ambiente mais quente e mais seco do que D2, conforme os locais que os mesmos foram alocados. Já o dispositivo D3 exibe desvios padrões bem maiores do que os outros dispositivos nestas duas características, confirmando que está posicionado em um local de condições altamente variáveis.

Para complementar a análise, foram gerados gráficos de comparação entre o histograma normalizado dos dados e a curva gaussiana destes dados, obtida a partir da média e do desvio padrão dos mesmos, para cada uma das cinco características e para os três dispositivos. Nos gráficos, o eixo x representa o valor da característica. Já o eixo y retorna, para o histograma, o número de vezes que tal valor se repete (neste caso, uma quantidade normalizada) e, para a curva gaussiana, retorna a função $g(x)$, dada pela Equação 2.1.

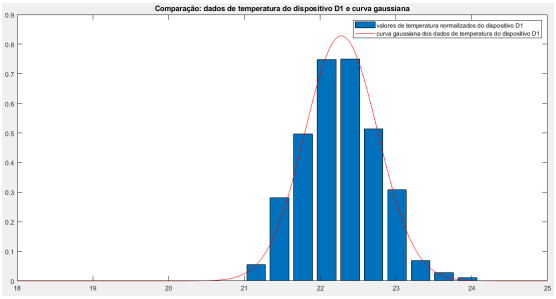
A comparação obtida pode ser vista nas Figuras 7 e 8 para o dispositivo D1, nas Figuras 9 e 10 para o dispositivo D2 e nas Figuras 11 e 12 para o dispositivo D3.

Figura 7 – Comparação dos dados normalizados de CO, umidade, LPG e fumaça do dispositivo D1 com a sua curva gaussiana



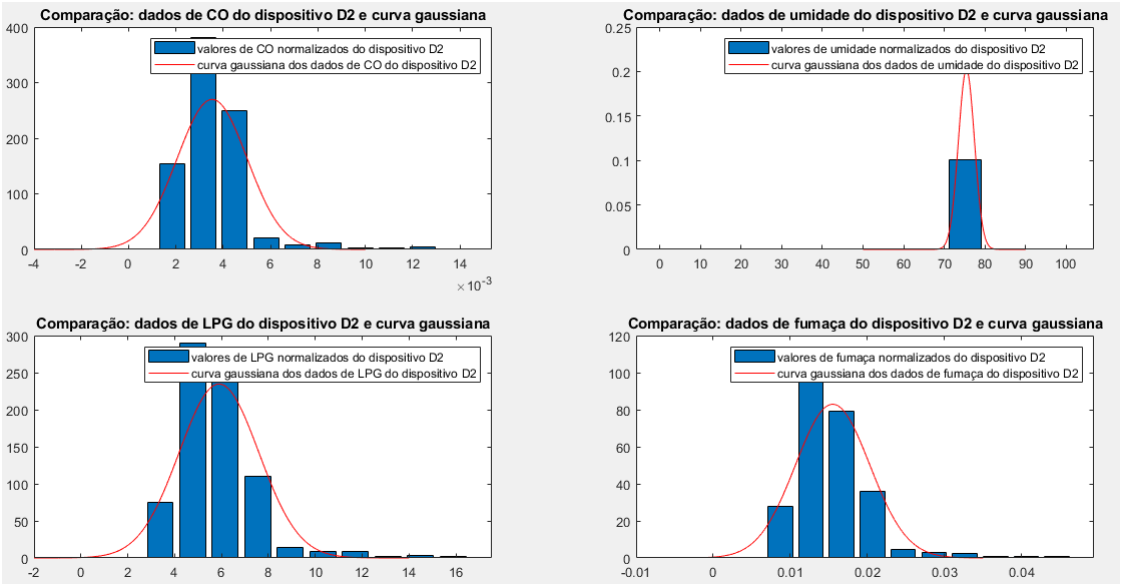
Fonte: elaborado pela autora (2021).

Figura 8 – Comparação dos dados normalizados de temperatura de D1 com a gaussiana



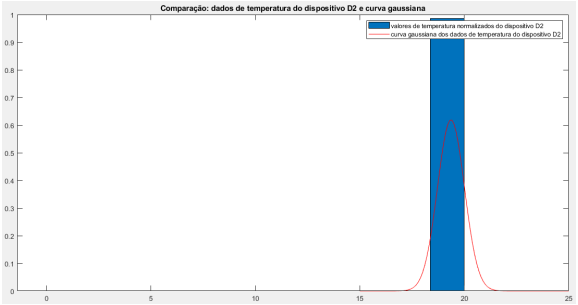
Fonte: elaborado pela autora (2021).

Figura 9 – Comparação dos dados normalizados de CO, umidade, LPG e fumaça do dispositivo D2 com a sua curva gaussiana



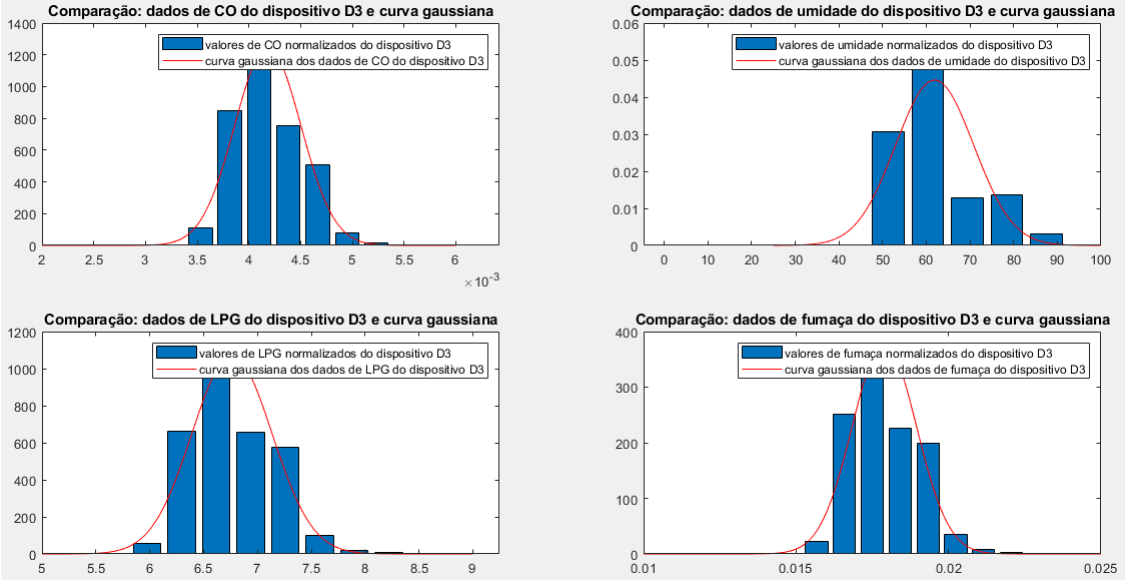
Fonte: elaborado pela autora (2021).

Figura 10 – Comparação dos dados normalizados de temperatura de D2 com a gaussiana



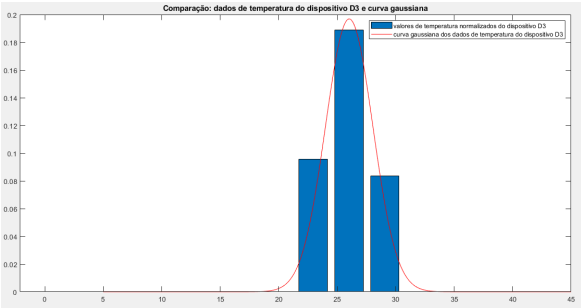
Fonte: elaborado pela autora (2021).

Figura 11 – Comparação dos dados normalizados de CO, umidade, LPG e fumaça do dispositivo D3 com a sua curva gaussiana



Fonte: elaborado pela autora (2021).

Figura 12 – Comparação dos dados normalizados de temperatura de D3 com a gaussiana



Fonte: elaborado pela autora (2021).

Observando-se os gráficos de comparação das Figuras 7 a 12, foi constatado que o dispositivo D1 apresenta alguns valores distantes além de 3σ do valor médio nas características CO, LPG, fumaça e temperatura; enquanto que o dispositivo D2 possui tal comportamento, um pouco mais acentuado, nas características CO, LPG e fumaça; já o dispositivo D3 exibe valores destoantes além de 3σ nos parâmetros de CO, umidade, LPG e fumaça. Ressalta-se que na característica de umidade para D3, esta discrepância ocorre mais vezes para um mesmo valor, isto é, o tamanho da barra do histograma é maior no eixo y.

Também foi possível verificar que os dispositivos D1 e D2 apresentam pouca variação nos valores de umidade e de temperatura, ao mesmo tempo que o dispositivo D3 exibe grande

variação nos mesmos. A umidade varia em torno de 45% a 57% em D1 e de 70% a 80% em D2, enquanto que em D3 esta variação vai de cerca de 48% a 92%. De igual forma, a temperatura medida vai aproximadamente de 21F a 24F no dispositivo D1, de 18F a 20F em D2 e de 22F a 31F em D3.

3.1.3 Rótulos dos dados

O conjunto de dados do dispositivo D1 possui 187541 elementos, sendo destes 2665 rotulados como *outliers* (1,42%) e 184876 como normais (98,58%). Já o dispositivo D2 possui 111815 dados, sendo 2846 *outliers* (2,545%) e 108969 normais (97,455%). Além disso, o dispositivo D3 possui 105918 dados, sendo 2805 rotulados *outliers* (2,65%) e 103113 classificados como normais (97,35%).

3.2 OUTLIERS E TÉCNICAS DE DETECÇÃO

Neste trabalho, detecta-se a presença de *outliers*, cuja relevância é descrita no Capítulo 1. Foram estabelecidas as técnicas de detecção a serem implementadas: Zscore, Zscore Modificado e K-Means.

A escolha pelos métodos estatísticos Zscore e Zscore Modificado se deu com base no descrito por Fetterman (2015), o qual afirma que entre os métodos univariados mais utilizados estão estes dois algoritmos. Além disso, tais técnicas são bastante empregadas em dados que seguem uma distribuição unimodal, sendo que grande parte das características do conjunto de dados escolhido segue esta distribuição, conforme pode-se observar nos gráficos de histograma e curva gaussiana. Sendo assim, escolheu-se tais técnicas de detecção em virtude de sua eficácia e ampla aceitação, esperando-se que estes algoritmos apresentem maior eficiência nos dados cuja distribuição se assemelhe mais a distribuição normal, visto que se baseiam nesta consideração.

Para fins de comparação, análise e aprendizado, também foi determinado o uso de uma técnica de detecção que segue os princípios da clusterização. Dentre estas, foi escolhida a técnica K-Means, em razão de ser a mais utilizada. É esperado que este método apresente melhor desempenho na detecção de *outliers* em dados de distribuição bimodal em relação aos métodos estatísticos.

3.3 IMPLEMENTAÇÃO DAS TÉCNICAS

De acordo com Souza (2020), pode-se ter modelos vetoriais (unidimensional), matriciais (multivariada) ou tensoriais (multidimensional), sendo que os modelos vetoriais são as soluções mais básicas para analisar dados de monitoramento ambiental. Em vista disto, as cinco variáveis ambientais analisadas foram organizadas em vetores, resultando em uma programação mais simples e rápida. Tratam-se de dados univariados, sendo independentes uns dos outros.

Inicialmente, buscou-se um entendimento geral sobre a técnica de detecção, isto é, como o algoritmo funciona e suas limitações. Então, foram desenvolvidos os códigos, identificando e corrigindo possíveis falhas. Em seguida, teve-se como objetivo melhorar os códigos, deixando-os mais simples e velozes. Além disso, primeiro os métodos foram testados com um conjunto reduzido dos dados, a fim de avaliar o seu comportamento, depois expandindo-os para o conjunto inteiro. Após a implementação dos algoritmos, foi realizada a validação dos mesmos, por meio da inserção de valores *outliers* conhecidos e verificação se os códigos detectam estes valores.

3.3.1 Zscore

O algoritmo Zscore desenvolvido segue uma lógica bem simples, resumida no pseudo-código a seguir, conforme o raciocínio exposto no Capítulo 2. Isto é, são considerados *outliers* os vetores em que a média das métricas Z_{score} das cinco características é maior ou igual a 3. Não foram analisados os parâmetros de modo individual em virtude de que a forma estabelecida para rotular o conjunto de dados em normais e *outliers* seguiu este princípio, conforme detalhado na Seção 4.4

1. Algoritmo: "Zscore".

2. Entradas: matrizes de dados "dados_D1", "dados_D2" e "dados_D3".

3. Parâmetros: média e desvio padrão dos dados.

4. Saídas: vetores *outliers* e vetores de dados normais.

5. Processamento:

5.1. Ler os dados nas matrizes;

5.2. Calcular a média dos dados;

5.3. Calcular o desvio padrão dos dados;

5.4. Determinar a métrica Z_{score} de cada característica a partir da Equação 2.2;

5.5. Calcular a média dos módulos das métricas Z_{score} de cada vetor;

5.6. Verificar se a média Z_{score} do vetor é $\leq$ (dado normal) ou $>$ que 3 (*outlier*).

6. Fim do algoritmo

Após a implementação do algoritmo, verificou-se o funcionamento do mesmo. Esta parte de validação mostrou-se muito importante, pois permite a identificação de erros na lógica do algoritmo.

3.3.2 Zscore Modificado

O método Zscore Modificado segue uma lógica semelhante a do Zscore, porém com a utilização da mediana e do desvio absoluto da mediana (MAD) em lugar da média e do desvio padrão. Sendo assim, o algoritmo desenvolvido apresenta a estrutura descrita no pseudocódigo a seguir. Dessa forma, são definidos como *outliers* os vetores em que M_i é maior ou igual do que determinado valor limite estabelecido.

1. Algoritmo: "Zscore Modificado".

2. Entradas: matrizes de dados "dados_D1", "dados_D2" e "dados_D3".

3. Parâmetros: mediana e desvio absoluto da mediana (MAD) dos dados.

4. Saídas: vetores *outliers* e vetores de dados normais.

5. Processamento:

5.1. Ler os dados nas matrizes;

5.2. Calcular a mediana dos dados;

5.3. Calcular MAD de cada coluna, referente a cada uma das cinco características, conforme a Equação 2.3;

5.4. Determinar a métrica M_i de cada característica a partir da Equação 2.4;

5.5. Calcular a média dos módulos das métricas M_i de cada vetor;

5.6. Estabelecer valor limite para a média M_i do vetor para ele ser considerado *outlier*;

5.7. Verificar se a média M_i do vetor é $\leq$ (dado normal) ou $>$ que o valor limite estabelecido (*outlier*).

6. Fim do algoritmo

Inicialmente, os limites de M_i foram estabelecidos iguais a 3,5, conforme recomendado por Iglewics e Hoaglin (1993).

3.3.3 K-Means

Para a implementação do K-Means, foi utilizado um código baseado na lógica descrita por Parker (2011). Este algoritmo solicita como entrada o número de clusters n_c desejado e, para um conjunto de dados x, retorna uma matriz denominada codebook. Esta matriz possui n_c linhas e o mesmo número de colunas que x, cada linha contendo um dos centros dos clusters, isto é, o vetor médio de cada cluster. O algoritmo define estes clusters a partir da determinação aleatória de n_c centros iniciais e cálculo da distância euclidiana entre todos os pontos de x e cada centro, classificando-os como pertencentes ao cluster mais próximo. Então, recalcula os novos centros com os pontos classificados em cada região, por meio da média dos dados, até convergir.

No caso em estudo, determinou-se x como sendo a união das matrizes "dados_D1", "dados_D2" e "dados_D3", nesta ordem, uma abaixo da outra. Foi estabelecido n_c = 3, pretendendo-se que o algoritmo classificasse os dados de cada dispositivo em um cluster diferente. Para a determinação de *outliers* foi definida uma distância euclidiana limite de 3 desvios padrões em relação aos centros dos clusters. Os dados que estão fora destes limites para os 3 centros são considerados anormalidades.

Para tal, foi calculado o desvio padrão das 5 características de cada um dos 3 dispositivos, sendo este multiplicado por 3, visto que se deseja um limite de 3 desvios padrões. Então, estes valores foram somados aos centros dos clusters, armazenados em codebook, o que resultou em 3 vetores limites de classificação. A seguir, determinou-se a distância euclidiana destes vetores de fronteira em relação ao respectivo vetor médio. Finalmente, é calculada a distância dos vetores presentes em x que se deseja classificar em relação ao vetor médio e observado se a mesma fica menor (dado normal) ou maior (*outlier*) do que a distância do vetor limite calculada anteriormente. Caso seja um dado normal, verifica-se, dentre os clusters que o dado está dentro dos limites de fronteira, qual o centro do cluster mais próximo, classificando o dado como pertencente aquele cluster.

O algoritmo construído, que faz uso dos centros armazenados na matriz codebook, pode ser resumido conforme descrito a seguir. As distâncias são calculadas em um espaço de cinco dimensões, considerando cada vetor composto pelas cinco características analisadas como um ponto deste espaço.

1. Algoritmo: "K-Means".

2. Entradas: matrizes de dados "dados_D1", "dados_D2" e "dados_D3" e matriz codebook.

3. Parâmetros: desvio padrão dos dados, vetores limites e distâncias euclidianas.

4. Saídas: vetores *outliers* e vetores classificados em cada cluster.

5. Processamento:

5.1. Ler os dados nas matrizes;

5.2. Receber a matriz codebook com os centros dos 3 clusters;

5.3. Calcular o desvio padrão dos dados;

5.4. Determinar os vetores limites equivalentes a soma de 3 desvios padrões nos centros armazenados por codebook;

5.5. Calcular a distância máxima dist_max para um dado ser considerado normal, correspondente a distância euclidiana entre os vetores limites e os centros dos clusters;

5.6. Calcular a distância dist_dados dos dados em relação aos centros;

5.7. Verificar se dist_dados é maior que os 3 valores de dist_max (um de cada centro de cluster) e, se for, classificar o dado como *outlier*;

5.8. Caso contrário, verificar dentre as dist_max que dist_dados é menor, qual o cluster correspondente mais próximo do dado e classificar o dado como pertencente aquele cluster.

6. Fim do algoritmo

4 PROCESSAMENTO E VALIDAÇÃO DO SISTEMA DE DETECÇÃO

Neste Capítulo, são demonstradas as etapas de processamento e validação das três técnicas de detecção de *outliers* implementadas, bem como a definição de rótulos para o conjunto de dados (classificação em dados normais e *outliers*) e a implementação do multiclassificador.

4.1 ZSCORE

Para verificação do funcionamento do algoritmo Zscore, foram inseridos no conjunto de dados, de cada um dos dispositivos, 10 dados pertencentes a cada um dos outros dois dispositivos, obtendo-se as detecções a seguir:

- Conjunto "dados_D1": método detectou os dados de D2 e D3 inseridos como *outliers*.

- Conjunto "dados_D2": foram identificados como *outliers* os dados de D1 e D3.

- Conjunto "dados_D3": não identificou-se os dados de nenhum dos dois dispositivos D1 e D2 como *outliers*. Isso ocorre, possivelmente, pelo fato de que os dados de D3 apresentam grande desvio padrão, possuindo valores bem mais destoantes no próprio conjunto do que os dados de D1 e de D2 inseridos.

Também, foram inseridos vetores com valores bem discrepantes, representados na Figura 13, para melhor validação do funcionamento do algoritmo. As colunas das tabelas apresentadas nesta Figura e nas Figuras seguintes correspondem às cinco variáveis ambientais analisadas, nesta ordem:

- Coluna 1: monóxido de carbono (partes por milhão - %);

- Coluna 2: umidade (%);

- Coluna 3: gás de petróleo liquefeito (partes por milhão - %);

- Coluna 4: concentração de fumaça (partes por milhão - %);

- Coluna 5: temperatura (F).

O algoritmo Zscore identifica todos os vetores *outliers* da Figura 13 para os dispositivos D1 e D2. Para o dispositivo D3, o método não identifica os *outliers* das linhas 4 e 6, sendo que o vetor médio dos dados deste dispositivo é exibido na Figura 14.

Logo, constata-se que os vetores das linhas 4 e 6 da Figura 13 apresentam grande discrepância principalmente na característica de umidade em relação ao vetor médio de D3 e deveriam ter sido detectados como *outliers*. Desse modo, é verificada uma limitação do algoritmo Zscore.

Figura 13 – Vetores *outliers* inseridos para validar os algoritmos Zscore e Zscore Modificado

	1	2	3	4	5
1	0.0200	50	0.0078	0.0202	22.6000
2	0.0200	75	0.0050	0.0132	19.7000
3	0.0200	77	0.0070	0.0185	27
4	0.0050	22	0.0078	0.0202	22.6000
5	0.0029	43	0.0050	0.0132	19.7000
6	0.0044	17	0.0070	0.0185	27
7	0.0050	50	0.0780	0.0202	22.6000
8	0.0029	77	0.0800	0.0132	19.7000
9	0.0200	50	0.0078	0.0202	71
10	0.0200	75	0.0050	0.0132	2
11	0.0200	77	0.0070	0.0185	54

Fonte: elaborado pela autora (2021).

Figura 14 – Vetor médio do dispositivo D3 com a inserção de *outliers*

	1	2	3	4	5
1	0.0042	62.2761	0.0067	0.0178	25.6466

Fonte: elaborado pela autora (2021).

No caso, quando só uma das características é bastante discrepante, os Z_{score} das outras características acabam se aproximando de zero e alteram para baixo a média dos Z_{score} das cinco características, resultando na não detecção.

4.2 ZSCORE MODIFICADO

Para a determinação dos limites de M_i, comparou-se a detecção de *outliers* pelo Zscore Modificado com a do Zscore. Ao rodar o algoritmo Zscore para todo o conjunto de dados, ele classifica os dados do dispositivo D1 como tendo 0 *outliers*, D2 com 1206 *outliers* e D3 com 69 *outliers*. Implementou-se o Zscore Modificado com diferentes limites de M_i para detecção de *outliers*, para se ter uma ideia melhor do seu comportamento:

- Todos os vetores com $M_i \geq 3,5$ são considerados *outliers*: detecta-se 0 *outliers* em D1, 1673 *outliers* em D2 e 17 *outliers* em D3;

- Todos os vetores com $M_i \geq 4,5$ são identificados como *outliers* (menos rigoroso): são detectados 0 *outliers* em D1, 1076 *outliers* em D2 e 0 *outliers* em D3;

- Todos os vetores com $M_i \geq 3$ são classificados como *outliers* (mais rigoroso): identifica-se 0 *outliers* em D1, 2961 *outliers* em D2 e 84 *outliers* em D3.

Teoricamente, o método Zscore Modificado deve identificar mais *outliers* que o Zscore

simples, em virtude do efeito de *masking*, em que *outliers* mais extremos mascaram os menos extremos na detecção pelo Zscore. Em vista disto, analisando a quantidade de *outliers* detectados nos testes, para os dispositivos D1 e D2, o limite $M_i \geq 3,5$, conforme sugerido na literatura, seria plausível, visto que, em D1, ambos os algoritmos não detectam nenhum *outlier* e, em D2, Zscore Modificado identifica mais *outliers* do que o Zscore simples. Porém, para o dispositivo D3, deve-se considerar, possivelmente, um limite mais rigoroso para M_i, dado que o Zscore Modificado detecta menos *outliers* do que o Zscore simples para $M_i \geq 3,5$.

Para analisar se $M_i \geq 3,5$ é realmente um valor adequado para os dados do dispositivo D2, foi realizado um teste com os primeiros mil dados de D2 aplicando as duas técnicas estatísticas. O vetor médio dos primeiros mil dados do dispositivo D2 é retratado na Figura 15, enquanto que as Figuras 16 e 17 apresentam os *outliers* identificados pelos métodos Zscore simples e Zscore Modificado, respectivamente.

Figura 15 – Vetor médio dos primeiros mil dados do dispositivo D2

	1	2	3	4	5
1	0.0026	75.7997	0.0048	0.0123	19.6747

Fonte: elaborado pela autora (2021).

Figura 16 – *Outliers* identificados pelo Zscore simples nos primeiros mil dados de D2

	1	2	3	4	5
1	0.0027	63	0.0049	0.0128	6.8000
2	0.0026	76.3000	0.0048	0.0124	7

Fonte: elaborado pela autora (2021).

Figura 17 – *Outliers* identificados pelo Zscore Modificado nos primeiros mil dados de D2

	1	2	3	4	5
1	0.0028	62.3000	0.0051	0.0133	19.5000
2	0.0027	63	0.0049	0.0128	6.8000
3	0.0026	76.3000	0.0048	0.0124	7

Fonte: elaborado pela autora (2021).

Observou-se que o Zscore Modificado identificou um *outlier* a mais do que o método Zscore, no caso, o vetor da linha 1 da Figura 17. Isso ocorreu em virtude do *outlier* mencionado não ser tão extremo, sendo mascarado pelos mais extremos na aplicação de Zscore.

Outro teste realizado utilizou os primeiros cinco mil dados do dispositivo D2. Neste caso, o Zscore identificou 12 *outliers*, todos corretos. Já o Zscore Modificado detectou 19 anormalidades, sendo as 12 identificadas pelo Zscore mais outros 7 vetores, todos também corretos. Estes 7 *outliers* apresentam valores de umidade entre 62% a 63%, aproximadamente, sendo que o valor médio é igual a 75,7%. Logo, tratam-se novamente de *outliers* menos extremos mascarados pelos mais extremos em Zscore. Portanto, por meio destes testes, foi possível constatar que $M_i \geq 3,5$ é um limite aceitável para os dados do dispositivo D2 na aplicação do Zscore Modificado, visto que se provou eficiente.

A fim de determinar o limite adequado de M_i para os dados do dispositivo D3, também comparou-se a detecção de *outliers* pelos algoritmos Zscore e Zscore Modificado. Para isso, utilizou-se todo o conjunto de dados de D3, que resulta no vetor médio exibido na Figura 18. Conforme já comentado, o Zscore identifica 69 *outliers* e o Zscore Modificado 17. A Figura 19 retrata os *outliers* identificados por Zscore Modificado. Para resumir as análises, foram observados apenas os 34 primeiros *outliers* detectados por Zscore, apresentados na Figura 20. Nestas Figuras, as setas vermelhas representam os *outliers* identificados somente pelo Zscore; as setas azuis denotam os *outliers* detectados apenas pelo Zscore Modificado; e as setas verdes retratam os *outliers* identificados por ambos.

Figura 18 – Vetor médio de todos os dados do dispositivo D3

	1	2	3	4	5
1	0.0042	61.9102	0.0068	0.0179	26.0255

Fonte: elaborado pela autora (2021).

Figura 19 – *Outliers* identificados pelo Zscore Modificado nos dados de D3

	1	2	3	4	5
1	0.0046	48.1000	0.0073	0.0193	1.6000
2	0.0058	56.4000	0.0086	0.0231	29.5000
3	0.0053	5.7000	0.0080	0.0215	29.5000
4	0.0058	57.1000	0.0086	0.0231	29.4000
5	0.0058	57.2000	0.0086	0.0231	29.4000
6	0.0059	57.3000	0.0086	0.0232	29.3000
7	0.0060	57.4000	0.0088	0.0238	29.4000
8	0.0058	57.4000	0.0086	0.0230	29.4000
9	0.0062	57.4000	0.0090	0.0243	29.3000
10	0.0062	57.4000	0.0090	0.0243	29.3000
11	0.0058	57.5000	0.0086	0.0232	29.3000
12	0.0061	57.4000	0.0089	0.0240	29.3000
13	0.0061	57.5000	0.0089	0.0239	29.2000
14	0.0060	57.6000	0.0088	0.0236	29.2000
15	0.0058	57.6000	0.0086	0.0232	29.2000
16	0.0060	57.7000	0.0088	0.0238	29.2000
17	0.0048	2.7000	0.0075	0.0200	30

Fonte: elaborado pela autora (2021).

Figura 20 – *Outliers* identificados pelo Zscore nos dados de D3

	1	2	3	4	5
1	0.0039	56.5000	0.0064	0.0168	2.1000
2	0.0038	47.9000	0.0063	0.0167	3
3	0.0038	53.8000	0.0063	0.0165	4.4000
4	0.0040	55.6000	0.0065	0.0171	0.2000
5	0.0040	55.7000	0.0065	0.0171	0.1000
6	0.0039	56.6000	0.0065	0.0170	0
7	0.0037	62.3000	0.0061	0.0161	1.1000
8	0.0037	62.4000	0.0062	0.0163	0.6000
9	0.0037	62.1000	0.0061	0.0162	0.5000
10	0.0038	67	0.0063	0.0167	1.4000
11	0.0038	67	0.0063	0.0166	1.6000
12	0.0057	55.8000	0.0085	0.0228	29.5000
13	0.0057	56.2000	0.0085	0.0228	29.5000
14	0.0058	56.4000	0.0086	0.0231	29.5000
15	0.0057	56.2000	0.0085	0.0228	29.5000
16	0.0058	57.1000	0.0086	0.0231	29.4000
17	0.0058	57.2000	0.0085	0.0229	29.4000
18	0.0058	57.2000	0.0086	0.0231	29.4000
19	0.0057	57.4000	0.0085	0.0228	29.4000
20	0.0059	57.3000	0.0086	0.0232	29.3000
21	0.0060	57.4000	0.0088	0.0238	29.4000
22	0.0058	57.4000	0.0086	0.0230	29.4000
23	0.0062	57.4000	0.0090	0.0243	29.3000
24	0.0062	57.4000	0.0090	0.0243	29.3000
25	0.0058	57.5000	0.0086	0.0232	29.3000
26	0.0061	57.4000	0.0089	0.0240	29.3000
27	0.0057	57.5000	0.0085	0.0229	29.2000
28	0.0061	57.5000	0.0089	0.0239	29.2000
29	0.0060	57.6000	0.0088	0.0236	29.2000
30	0.0058	57.6000	0.0086	0.0232	29.2000
31	0.0060	57.7000	0.0088	0.0238	29.2000
32	0.0058	57.7000	0.0086	0.0230	29.1000
33	0.0058	57.7000	0.0086	0.0230	29.1000
34	0.0057	57.6000	0.0085	0.0228	29

Fonte: elaborado pela autora (2021).

Então, foram observados os dados que o Zscore Modificado deveria ter necessariamente identificado como *outliers* e não identificou e seus respectivos M_i. Assim, foi constatado que dentre os vetores que o algoritmo não detectou como *outlier* o menor M_i é igual a 2,67. Sendo assim, rodou-se o algoritmo para todos os dados de D3, considerando como *outliers* os vetores em que $M_i \geq 2,67$, obtendo como resultado a detecção de 334 *outliers*.

Para analisar se este novo limite é adequado, foi realizado um teste com os primeiros 90 mil dados de D3, pois um conjunto reduzido de dados facilita a análise. Neste caso, o algoritmo Zscore detectou 37 *outliers*, sendo 36 deles corretamente identificados e um dado normal classificado incorretamente como anormal, o qual possui Z_{score} = 3,0787, isto é, bem próximo do limite igual a 3. Já o método Zscore Modificado detectou 57 *outliers*, sendo estes iguais aos detectados por Zscore mais outros 20 vetores. 48 vetores foram corretamente classificados como *outliers* pelo Zscore Modificado e 9 vetores de dados normais foram incorretamente classificados como anormalidades, em virtude do efeito de *swamping*. Os índices M_i destes 9 vetores são iguais a 3,0155, 3,1952, 3,2813, 2,73, 2,9048, 3,1673, 2,8305 e 2,682. Porém, apesar disto, é necessário um limite de M_i igual a 2,67 para identificar outros *outliers*, conforme observado.

4.2.1 Testes de validação para $M_i \geq 3,5$ (dados de D1 e D2) e $M_i \geq 2,67$ (dados de D3)

A fim de validar o funcionamento desta técnica, seguiu-se os mesmos procedimentos realizados com o algoritmo Zscore. Ou seja, inicialmente, foram inseridos no conjunto de dados, de cada um dos dispositivos, 10 dados pertencentes a cada um dos outros dois dispositivos, resultando nas seguintes detecções:

- Conjunto "dados_D1": método detectou os dados de D2 e D3 como *outliers*.

- Conjunto "dados_D2": identificou-se como *outliers* os dados de D1 e D3.

- Conjunto "dados_D3": algoritmo não identificou os dados de nenhum dos dois dispositivos D1 e D2 como *outliers*, em virtude, provavelmente, da grande discrepância de valores no próprio conjunto de dados do dispositivo D3.

Além disso, foram inseridos os mesmos vetores de valores bem discrepantes usados na validação do Zscore, apresentados na Figura 13. O método Zscore Modificado identifica todos os *outliers* inseridos para os dispositivos D1 e D2, enquanto que para D3, o algoritmo deixa de identificar o vetor *outlier* da linha 6 da Figura 13. O vetor mediana dos dados de D3 pode ser visualizado na Figura 21. Logo, observa-se que o vetor da linha 6 da Figura 13 não identificado apresenta elevada discrepância em relação ao vetor mediana na característica de umidade, enquanto que as outras características são bem semelhantes.

Figura 21 – Vetor mediana do dispositivo D3 com a inserção de *outliers*

	1	2	3	4	5
1	0.0041	59.8000	0.0067	0.0176	25.6000

Fonte: elaborado pela autora (2021).

Em vista disso, verifica-se uma limitação do Zscore Modificado: quando só uma das características é bastante destoante, os M_i das outras características acabam se aproximando de zero e puxam para baixo a média dos M_i das cinco características, resultando na não detecção.

4.3 K-MEANS

Inicialmente, rodou-se o algoritmo que retorna o vetor codebook, com um número de clusters igual a 3, a fim de analisar se estes centros são condizentes aos vetores médios dos três dispositivos. Rodando com um conjunto reduzido de dados, composto pelos primeiros três mil dados de cada um dos dispositivos, obteve-se o vetor codebook apresentado na Figura 25, sendo

que os vetores médios do conjunto de dados testado dos dispositivos D1, D2 e D3 podem ser visualizados nas Figuras 22, 23 e 24, nesta ordem.

Figura 22 – Vetor médio dos primeiros três mil dados de D1

	1	2	3	4	5
1	0.0050	50.6378	0.0077	0.0206	22.4471

Fonte: elaborado pela autora (2021).

Figura 23 – Vetor médio dos primeiros três mil dados de D2

	1	2	3	4	5
1	0.0026	75.7997	0.0048	0.0123	19.6747

Fonte: elaborado pela autora (2021).

Figura 24 – Vetor médio dos primeiros três mil dados de D3

	1	2	3	4	5
1	0.0042	77.5443	0.0068	0.0181	26.8165

Fonte: elaborado pela autora (2021).

Figura 25 – Vetor codebook para os primeiros três mil dados de cada dispositivo

	1	2	3	4	5
1	0.0034	76.6988	0.0058	0.0152	23.2540
2	0.0027	63	0.0049	0.0128	6.8000
3	0.0050	50.6500	0.0077	0.0206	22.4484

Fonte: elaborado pela autora (2021).

Assim, constatou-se que os *outliers* acabam por deslocar os centros dos clusters, dado que o vetor da linha 2 de codebook (Figura 25) apresenta temperatura igual a 6,8°C, valor que não é próximo do valor médio desta característica em nenhum dos dispositivos. Esta constatação foi reforçada ao obter os centros exibidos na Figura 26, condizentes aos vetores médios dos dispositivos, excluindo-se os *outliers* conhecidos de cada dispositivo (linhas 160, 244 e 275 de D2 e linha 320 de D3) e rodando o algoritmo para o mesmo conjunto reduzido de dados.

A fim de verificar se este algoritmo poderia realmente ser utilizado, rodou-se o código com todo o conjunto de dados, para observar se neste caso os *outliers* também deslocam os

Figura 26 – Vetor codebook para os primeiros três mil dados de cada dispositivo sem *outliers*

	1	2	3	4	5
1	0.0034	76.6972	0.0058	0.0152	23.2659
2	0.0023	25.1000	0.0045	0.0115	19.7000
3	0.0050	50.6384	0.0077	0.0206	22.4513

Fonte: elaborado pela autora (2021).

centros. Obteve-se os vetores médios de D1, D2 e D3 exibidos nas Figuras 27, 28 e 29 e os centros apresentados na Figura 30.

Figura 27 – Vetor médio de todo o conjunto de dados de D1

	1	2	3	4	5
1	0.0056	50.8141	0.0083	0.0223	22.2800

Fonte: elaborado pela autora (2021).

Figura 28 – Vetor médio de todo o conjunto de dados de D2

	1	2	3	4	5
1	0.0035	75.4444	0.0059	0.0155	19.3626

Fonte: elaborado pela autora (2021).

Figura 29 – Vetor médio de todo o conjunto de dados de D3

	1	2	3	4	5
1	0.0042	61.9102	0.0068	0.0179	26.0255

Fonte: elaborado pela autora (2021).

Figura 30 – Vetor codebook para todo o conjunto de dados

	1	2	3	4	5
1	0.0036	75.5941	0.0060	0.0158	20.3276
2	0.0042	62.1999	0.0068	0.0179	26.3057
3	0.0053	51.9843	0.0080	0.0213	23.2222

Fonte: elaborado pela autora (2021).

Em vista disto, verificou-se que prevalece a lei dos grandes números, pois os *outliers* praticamente não interferem na média e na determinação dos centros, quando analisado o con-

junto inteiro de dados. Também foi possível estabelecer a ordem dos centros em codebook respectiva a cada dispositivo: D2 (cluster 1), D3 (cluster 2), D1 (cluster 3).

Então, prosseguiu-se com a validação do algoritmo de detecção dos *outliers* K-Means, descrito no Capítulo 3. Testando o mesmo para todo o conjunto de dados, foi verificado que o método classifica vários dados do dispositivo D3 como sendo pertencentes aos clusters dos dispositivos D1 e D2, em virtude, dentre outras razões, dos diferentes picos de umidade dos dados de D3 (Figura 11).

Além disso, observou-se que alguns *outliers* do dispositivo D2 são classificados como sendo normais e pertencentes ao cluster de D3. Isso ocorre em virtude da distância máxima em relação ao centro do cluster 2 (referente aos dados do dispositivo D3), para um dado poder ser pertencente ao cluster, ficou muito elevada, enquanto que as distâncias máximas dos clusters 1 (dispositivo D2) e 3 (dispositivo D1) ficaram bem mais restritivas, como exibido na Figura 31.

Figura 31 – Distâncias máximas em relação aos centros dos clusters para um dado poder ser pertencente ao cluster

dist_max1	6.2341
dist_max2	27.5144
dist_max3	5.8483

Fonte: elaborado pela autora (2021).

Pensou-se como solução rodar o algoritmo que retorna codebook definindo o número de clusters igual a 4, esperando-se que isto dividisse os dados de D3 em dois clusters, um para cada pico de umidade, e diminuísse as distâncias máximas. Porém, como pode ser visualizado na Figura 32, o quarto centro adicionado tem umidade bem parecida aos outros centros e temperatura igual a 12,2°C, provavelmente para contemplar os *outliers* dos dispositivos.

Figura 32 – Vetor codebook com a definição de 4 clusters

	1	2	3	4	5
1	0.0036	75.5947	0.0060	0.0158	20.3283
2	0.0042	62.1792	0.0068	0.0179	26.3180
3	0.0040	61.2747	0.0065	0.0172	12.2115
4	0.0053	51.9747	0.0080	0.0213	23.2266

Fonte: elaborado pela autora (2021).

Assim sendo, não foi encontrada outra alternativa para resolver o impasse de que o limite do cluster 2 (dispositivos D3) deve ser menor para não incluir *outliers* de D2 como dados

normais e, por outro lado, deve ser maior para não identificar como *outliers* alguns dados normais de D3. Portanto, os limites de 3 desvios padrões em relação aos centros dos clusters não foram alterados, esperando-se que o multiclassificador a ser construído compense esta limitação do K-Means.

Por fim, foram inseridos vetores *outliers* conhecidos, com valores bem diferentes dos vetores médios dos três dispositivos. Todos estes vetores foram corretamente detectados como anormalidades pelo algoritmo.

4.4 DEFINIÇÃO DE RÓTULOS DO CONJUNTO DE DADOS

Para calcular as métricas de desempenho dos algoritmos, é necessária a definição de rótulos do conjunto de dados, uma vez que não se tem conhecimento de quais dados são *outliers* e quais são normais. Para tal, inicialmente, buscou-se definir um vetor *threshold* (limiar), para cada dispositivo, que representasse um padrão dos *outliers* identificados pelas três técnicas. A partir da distância deste vetor em relação ao vetor médio do dispositivo, todos os dados acima desta distância seriam rotulados como *outliers*.

Várias tentativas foram feitas, tais como um vetor *threshold* formado pela média dos *outliers* detectados por uma ou mais técnicas ou composto pela mediana destes *outliers*. Outra alternativa testada foi a definição do *threshold* a partir da média dos *outliers* menos extremos identificados pelas três técnicas. Porém, em todos estes casos, foi constatado que o *threshold* deixa de classificar como *outliers* alguns vetores anormais conhecidos do conjunto, mostrando-se, assim, ineficaz.

Em vista disto, esta rotulagem foi feita de outra maneira. Foram classificados como *outliers* todos os vetores em que ao menos uma das características desvia em mais do que 3 desvios padrões do seu valor médio. Esta classificação é mais robusta pois independe das respostas dos algoritmos.

4.5 IMPLEMENTAÇÃO DO MULTICLASSIFICADOR

Para a determinação do multiclassificador, algoritmo que reúne as respostas das três técnicas em um único sistema, inicialmente, atribuiu-se percentuais do quanto os dados detectados como *outliers* são extremos em relação aos outros. Isto é, para o *outlier* mais extremo identificado por cada uma das técnicas, foi atribuído um percentual igual a 100%, referente a um score

igual a 1. Os outros *outliers* detectados recebem um percentual proporcional ao mais extremo. Assim, os algoritmos retornam um vetor coluna com scores de 0 a 1 de quanto os dados devem ser considerados *outliers*.

Além disso, foram estabelecidos pesos a cada um dos algoritmos, de acordo com a sua eficiência demonstrada pelas métricas de desempenho individuais. Esta análise foi feita em separado para cada um dos três dispositivos, visto que uma técnica pode identificar melhor os *outliers* em um dos dispositivos e pior em outro, pois as distribuições dos dados dos mesmos são diferentes. Para um algoritmo que teve melhor desempenho identificando corretamente as anormalidades em um dado dispositivo, foi atribuído um peso maior, enquanto que estabeleceu-se pesos menores as técnicas menos eficazes.

Desse modo, a resposta M do multiclassificador é dada conforme a Equação 4.1, em que P_Z, P_{ZMod} e P_K representam os pesos atribuídos aos algoritmos Zscore, Zscore Modificado e K-Means, respectivamente, enquanto que $Poutliers_Z$, $Poutliers_{Zmod}$ e $Poutliers_K$ correspondem ao percentual de *outlier* detectado por cada uma das técnicas. Isto resulta em um vetor coluna com scores de 0 a 1 de quanto os dados são *outliers*.

$$M = P_Z \cdot Poutliers_Z + P_{ZMod} \cdot Poutliers_{ZMod} + P_K \cdot Poutliers_K \tag{4.1}$$

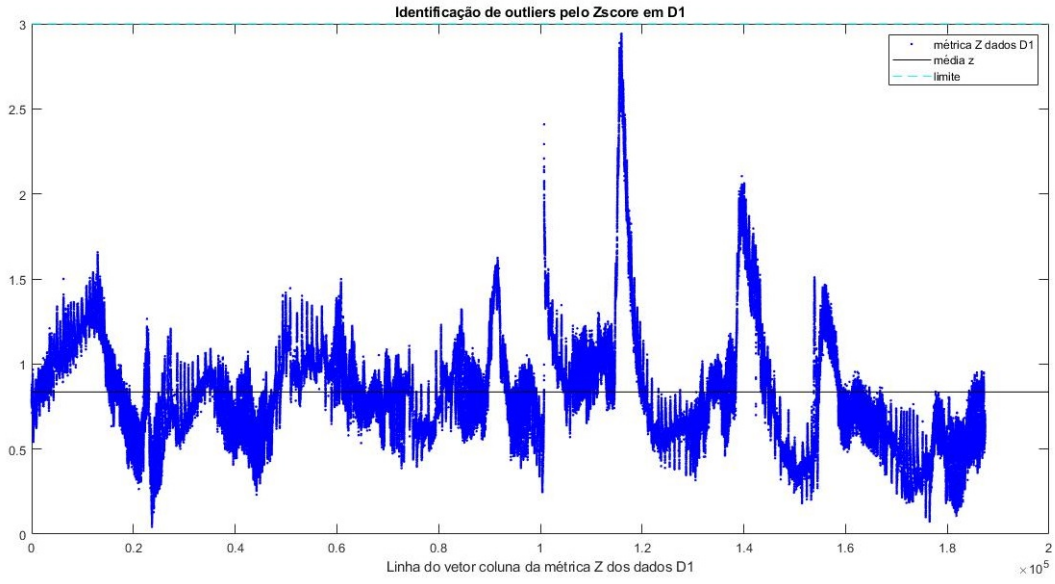
5 RESULTADOS E ANÁLISES

Neste Capítulo são apresentados os resultados de detecção de *outliers* pelas três técnicas implementadas e as análises dos mesmos, além dos cálculos das métricas de desempenho individuais e do multiclassificador.

5.1 DETECÇÃO DE OUTLIERS PELAS TÉCNICAS

As três técnicas, Zscore, Zscore Modificado e K-Means não identificaram nenhum *outlier* para o conjunto de dados do dispositivo D1. As Figuras 33 e 34 apresentam, respectivamente, as métricas Z_{score} e M_i destes vetores, sendo possível visualizar que nenhuma delas ultrapassa os limites estabelecidos para detecção de *outliers*. Para o algoritmo K-Means, não foi possível gerar gráfico em virtude de serem vetores que compõem um espaço de cinco dimensões, visto que existem cinco características sendo analisadas.

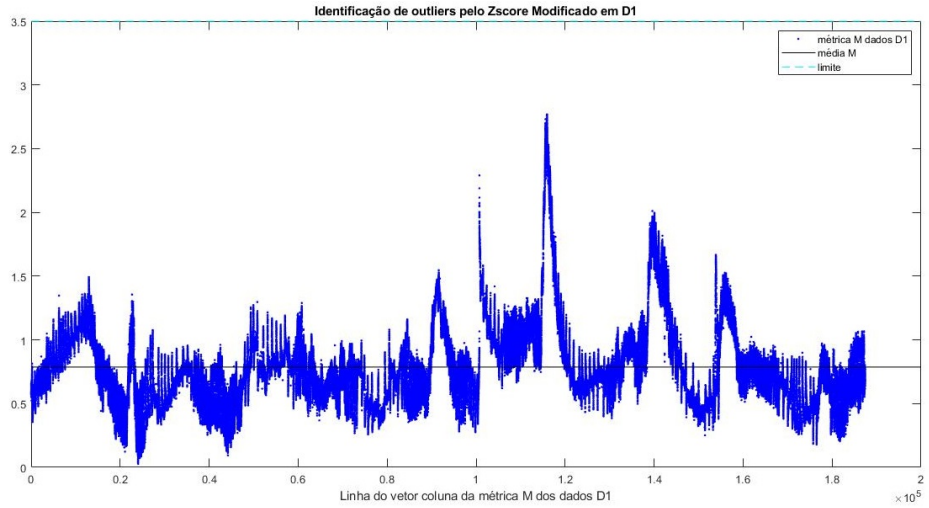
Figura 33 – Identificação de *outliers* pelo Zscore nos dados do dispositivo D1



Fonte: elaborado pela autora (2021).

Ressalta-se que a maioria dos *outliers* presentes em D1 apresentam variações pequenas, como é possível visualizar nos histogramas e curvas gaussianas das Figuras 7 e 8. Além disso, muitas vezes, apenas uma das características ultrapassa o limiar estabelecido, enquanto que as outras quatro características permanecem dentro dos seus limites. Isto traz um impacto insig-

Figura 34 – Identificação de *outliers* pelo Zscore Modificado nos dados do dispositivo D1

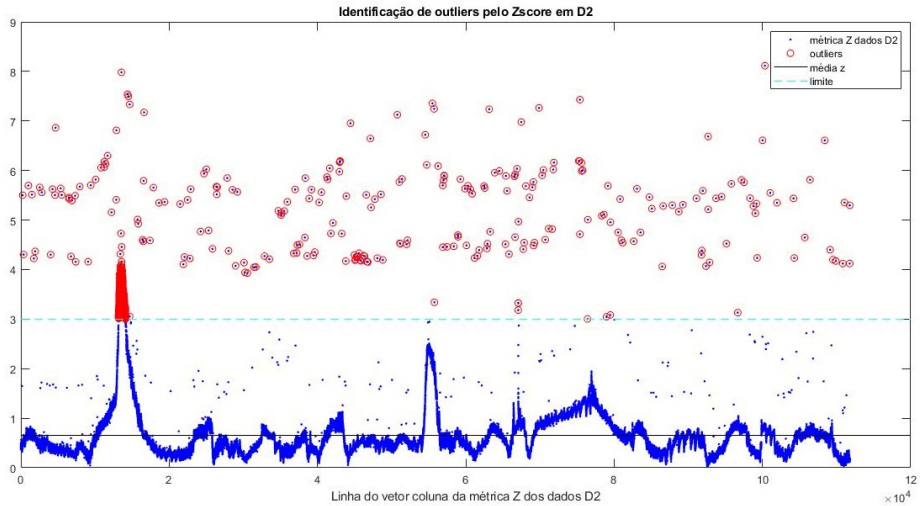


Fonte: elaborado pela autora (2021).

nificante na média das métricas Z_{score} e M_i de cada vetor, na aplicação das técnicas Zscore e Zscore Modificado, ou na distância euclidiana em relação ao centro do cluster na implementação de K-Means, resultando na não detecção.

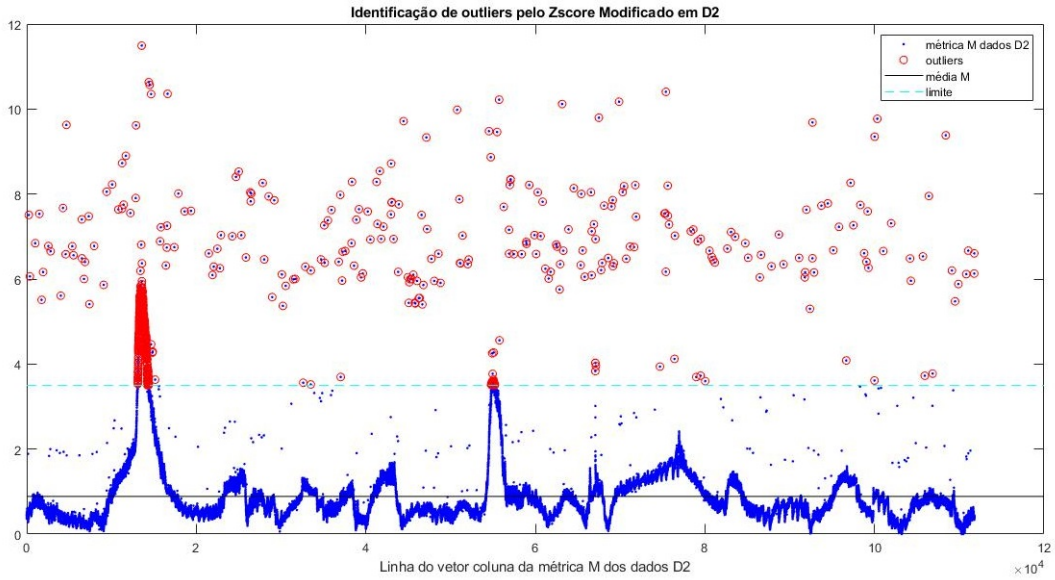
Para os dados do dispositivo D2, o algoritmo Zscore identificou 1206 *outliers* (1,08%) e 110609 dados normais (98,95%), apresentados na Figura 35. A técnica Zscore Modificado detectou 1673 *outliers* (1,5%) e 110142 dados normais (98,5%), exibidos na Figura 36. Pelo método K-Means foram identificados 69 *outliers* (0,062%) e 111746 dados normais (99,938%).

Figura 35 – Identificação de *outliers* pelo Zscore nos dados do dispositivo D2



Fonte: elaborado pela autora (2021).

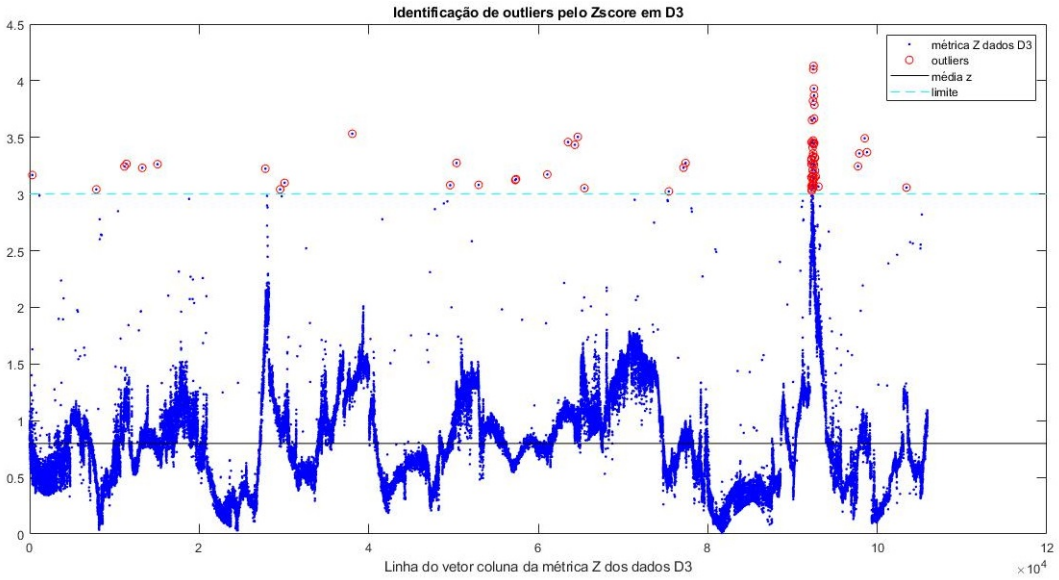
Figura 36 – Identificação de *outliers* pelo Zscore Modificado nos dados do dispositivo D2



Fonte: elaborado pela autora (2021).

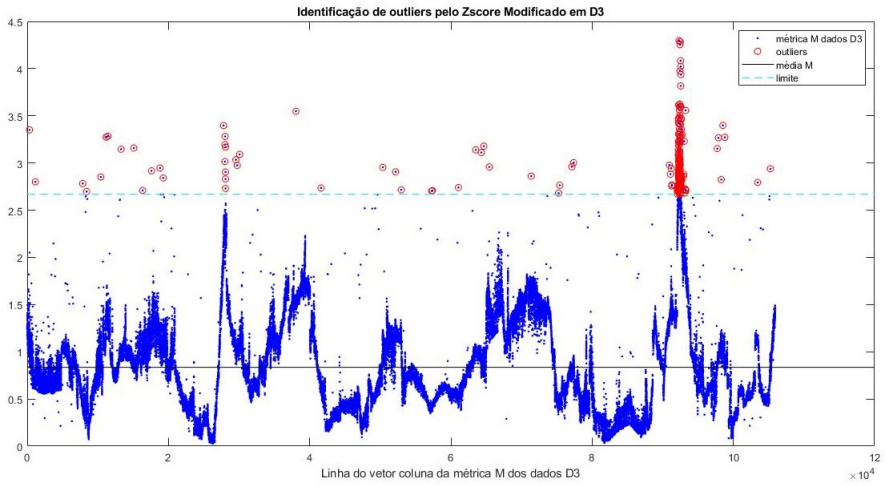
Nos dados do dispositivo D3, Zscore identificou 69 *outliers* (0,065%) e 105849 dados normais (99,935%), conforme mostrado na Figura 37. Zscore Modificado detectou 334 *outliers* (0,315%) e 105584 dados normais (99,685%), como pode ser visualizado na Figura 38. K-Means identificou 1899 *outliers* (1,793%) e 104019 dados normais (98,207%).

Figura 37 – Identificação de *outliers* pelo Zscore nos dados do dispositivo D3



Fonte: elaborado pela autora (2021).

Figura 38 – Identificação de *outliers* pelo Zscore Modificado nos dados do dispositivo D3



Fonte: elaborado pela autora (2021).

Analisando os resultados obtidos, constata-se a presença do efeito de *masking* na aplicação do algoritmo Zscore. Isso pode ser visualizado ao comparar as Figuras 35 e 36 de detecção de *outliers* nos dados do dispositivo D2, e as Figuras 37 e 38 que traz os *outliers* identificados nos dados do dispositivo D3, pelo Zscore e Zscore Modificado, nesta ordem. Alguns vetores que estavam próximos do limite de três desvios padrões de Zscore, mas não foram classificados como *outliers* em virtude do mascaramento por *outliers* mais extremos, são detectados como anormalidades pelo Zscore Modificado, comprovando que o mesmo é mais robusto.

5.2 MÉTRICAS DE DESEMPENHO INDIVIDUAIS

A partir das respostas dos algoritmos, calculou-se as métricas de desempenho individuais. As Tabelas 3, 4, 5 e 6 exibem, respectivamente, o número de verdadeiros positivos (VP), falsos positivos (FP), verdadeiros negativos (VN) e falsos negativos (FN) de cada técnica de detecção de *outliers*. Estes valores estão separados para cada dispositivo e também somados em um valor total.

Tabela 3 – Verdadeiros positivos (VP) de cada método de detecção

VP Zscore				VP Zscore Modificado				VP K-Means			
D1	D2	D3	Total	D1	D2	D3	Total	D1	D2	D3	Total
0	1206	69	1275	0	1673	334	2007	0	69	1899	1968

Fonte: elaborado pela autora (2022).

Tabela 4 – Falsos positivos (FP) de cada método de detecção

FP Zscore				FP Zscore Modificado				FP K-Means			
D1	D2	D3	Total	D1	D2	D3	Total	D1	D2	D3	Total
0	0	0	0	0	0	0	0	0	0	0	0

Fonte: elaborado pela autora (2022).

Tabela 5 – Verdadeiros negativos (VN) de cada método de detecção

VN Zscore				VN Zscore Modificado				VN K-Means			
D1	D2	D3	Total	D1	D2	D3	Total	D1	D2	D3	Total
184796	108969	103113	396878	184796	108969	103113	396878	184185	108969	64055	357209

Fonte: elaborado pela autora (2022).

Tabela 6 – Falsos negativos (FN) de cada método de detecção

FN Zscore				FN Zscore Modificado				FN K-Means			
D1	D2	D3	Total	D1	D2	D3	Total	D1	D2	D3	Total
2655	1640	2736	7031	2655	1173	2471	6299	2655	2777	906	6338

Fonte: elaborado pela autora (2022).

A partir destes valores, determinou-se a taxa de verdadeiros positivos (TVP), a precisão (P1 e P2), a especificidade (E) e a acurácia (A) de cada método de detecção. Estas métricas podem ser visualizadas nas Tabelas 7, 8, 9, 10 e 11, nesta ordem. Elas foram calculadas em relação ao conjunto de dados de cada dispositivo e também para todo o conjunto de dados, na coluna "Total", utilizando os valores totais de VP, FP, VN e FN, representando assim o desempenho geral de cada algoritmo.

Tabela 7 – Taxa de verdadeiros positivos (TVP) de cada método de detecção

TVP Zscore (%)				TVP Zscore Modificado (%)				TVP K-Means (%)			
D1	D2	D3	Total	D1	D2	D3	Total	D1	D2	D3	Total
0	42,38	2,46	15,35	0	58,78	11,91	24,16	0	2,42	67,7	23,69

Fonte: elaborado pela autora (2022).

Tabela 8 – Precisão (P1) de cada método de detecção

P1 Zscore (%)				P1 Zscore Modificado (%)				P1 K-Means (%)			
D1	D2	D3	Total	D1	D2	D3	Total	D1	D2	D3	Total
-	100	100	100	-	100	100	100	-	100	100	100

Fonte: elaborado pela autora (2022).

Tabela 9 – Precisão (P2) de cada método de detecção

P2 Zscore (%)				P2 Zscore Modificado (%)				P2 K-Means (%)			
D1	D2	D3	Total	D1	D2	D3	Total	D1	D2	D3	Total
98,58	98,52	97,42	98,26	98,58	98,94	97,66	98,44	98,58	97,51	98,61	98,26

Fonte: elaborado pela autora (2022).

Tabela 10 – Especificidade (E) de cada método de detecção

E Zscore (%)				E Zscore Modificado (%)				E K-Means (%)			
D1	D2	D3	Total	D1	D2	D3	Total	D1	D2	D3	Total
100	100	100	100	100	100	100	100	100	100	100	100

Fonte: elaborado pela autora (2022).

Tabela 11 – Acurácia (A) de cada método de detecção

A Zscore (%)				A Zscore Modificado (%)				A K-Means (%)			
D1	D2	D3	Total	D1	D2	D3	Total	D1	D2	D3	Total
98,58	98,53	97,42	98,26	98,58	98,95	97,67	98,45	98,58	97,52	98,64	98,27

Fonte: elaborado pela autora (2022).

Foi possível verificar que as técnicas Zscore e Zscore Modificado apresentam desempenho superior a K-Means na detecção de *outliers* nos dados do dispositivo D2, visto que a taxa de verdadeiros positivos (TVP) destas duas técnicas é igual a 42,38% e 58,78%, respectivamente, contra apenas 2,42% do método K-Means. Isto se deve, principalmente, a distribuição dos dados de D2 ser semelhante a uma distribuição normal unimodal, conforme visto nas Figuras 9 e 10, na qual os métodos estatísticos implementados se baseiam. Ainda, o método K-Means é mais ineficiente para este conjunto de dados devido ao limite do cluster respectivo aos dados do dispositivo D3 ser muito elevado, o que resulta na inclusão de *outliers* de D2 como dados normais deste cluster.

Por outro lado, para os dados do dispositivo D3, o método K-Means é o que possui

melhor desempenho, apresentando TVP igual a 67,7% contra meros 2,46% de Zscore e 11,91% de Zscore Modificado. Possivelmente, isto ocorre em razão da distribuição dos dados de D3 se aproximar a uma distribuição bimodal, sobretudo os dados de umidade, de acordo com o apresentado nas Figuras 11 e 12. Desse modo, o desvio padrão e o desvio absoluto da mediana (MAD), calculados para os algoritmos Zscore e Zscore Modificado, são determinados para um valor que não é a média ou mediana real dos dados, sendo a distância dos vetores obtida em relação a um vetor padrão que não é verdadeiro para todo o conjunto. Isto implica em um desempenho ruim destas duas técnicas.

Em vista disto, comprovou-se a necessidade da criação de um multiclassificador, uma vez que pode-se fazer uso do melhor das três técnicas, uma complementando a outra.

5.3 MÉTRICAS DE DESEMPENHO DO MULTICLASSIFICADOR

Com base nas métricas de desempenho individuais, definiu-se pesos para cada método:

- Conjunto "dados_D1": peso igual a $\frac{1}{3}$ para todos os algoritmos;
- Conjunto "dados_D2": Zscore = 0,41, Zscore Modificado = 0,56 e K-Means = 0,03.
- Conjunto "dados_D3": Zscore = 0,03, Zscore Modificado = 0,145 e K-Means = 0,825.

Assim sendo, foram obtidos para o multiclassificador o número de verdadeiros positivos (VP), falsos positivos (FP), verdadeiros negativos (VN) e falsos negativos (FN) apresentados nas Tabelas 12, 13, 14 e 15, nesta ordem. Estes valores são referentes ao conjunto de dados de cada dispositivo e a soma de todos, correspondente a coluna denominada "Total".

Tabela 12 – Verdadeiros positivos (VP) do multiclassificador

VP Multiclassificador			
D1	D2	D3	Total
0	1674	2229	3903

Fonte: elaborado pela autora (2022).

Tabela 13 – Falsos positivos (FP) do multiclassificador

FP Multiclassificador			
D1	D2	D3	Total
0	0	0	0

Fonte: elaborado pela autora (2022).

Tabela 14 – Verdadeiros negativos (VN) do multiclassificador

VN Multiclassificador			
D1	D2	D3	Total
184796	108969	103113	396878

Fonte: elaborado pela autora (2022).

Tabela 15 – Falsos negativos (FN) do multiclassificador

FN Multiclassificador			
D1	D2	D3	Total
2655	1172	576	4403

Fonte: elaborado pela autora (2022).

Assim, foi possível obter a taxa de verdadeiros positivos (TVP), a precisão (P1 e P2), a especificidade (E) e a acurácia (A) do multiclassificador, as quais são apresentadas, respectivamente, nas Tabelas 16, 17, 18, 19 e 20. Estas métricas foram obtidas em relação ao conjunto de dados de cada dispositivo e ainda para todo o conjunto de dados, na coluna "Total", utilizando os valores totais de VP, FP, VN e FN, correspondendo ao desempenho geral do multiclassificador.

Tabela 16 – Taxa de verdadeiros positivos (TVP) do multiclassificador

TVP Multiclassificador (%)			
D1	D2	D3	Total
0	58,82	79,47	46,99

Fonte: elaborado pela autora (2022).

Tabela 17 – Precisão (P1) do multiclassificador

P1 Multiclassificador (%)			
D1	D2	D3	Total
-	100	100	100

Fonte: elaborado pela autora (2022).

Tabela 18 – Precisão (P2) do multiclassificador

P2 Multiclassificador (%)			
D1	D2	D3	Total
98,58	98,94	99,44	98,90

Fonte: elaborado pela autora (2022).

Tabela 19 – Especificidade (E) do multiclassificador

E Multiclassificador (%)			
D1	D2	D3	Total
100	100	100	100

Fonte: elaborado pela autora (2022).

Tabela 20 – Acurácia (A) do multiclassificador

A Multiclassificador (%)			
D1	D2	D3	Total
98,58	98,95	99,46	98,91

Fonte: elaborado pela autora (2022).

Os resultados obtidos confirmaram que o multiclassificador apresenta uma detecção mais robusta de *outliers*, com a elevação de TVP para 58,82% nos dados de D2 e 79,47% nos dados de D3, além de reunir estes dois percentuais em uma mesma resposta. Além disso, os índices de precisão e acurácia também melhoraram.

6 CONCLUSÃO

Por meio dos resultados obtidos, constatou-se que o método Zscore Modificado é realmente mais robusto do que o Zscore, uma vez que este último possui o efeito de *masking* na detecção de *outliers*, o que leva a métricas de desempenho inferiores. Foi possível observar também que nenhuma das três técnicas detectou *outliers* no conjunto de dados do dispositivo D1, os quais apresentam anormalidades pouco destoantes ou apenas uma das cinco características analisadas ultrapassa os limites definidos.

Além disso, foi verificado que as técnicas estatísticas apresentam maior eficiência do que a clusterização em identificar anomalias do conjunto de dados de distribuição normal unimodal, referente ao dispositivo D2. Por outro lado, K-Means possui melhor performance em detectar *outliers* no conjunto de dados do dispositivo D3, cuja distribuição é mais semelhante a bimodal. Assim sendo, o multiclassificador construído elevou as métricas de desempenho, uma vez que fez uso do melhor das três técnicas.

Quanto às limitações deste estudo, pode-se citar as restrições nos resultados de Zscore e Zscore Modificado, devido ao fato destes classificadores considerarem que os dados seguem uma distribuição estatística normal, além da necessidade de ajuste manual do limite M_i do Zscore Modificado de acordo com a distribuição dos dados. Ademais, evidencia-se o fato de que os rótulos do conjunto de dados também se basearam em uma distribuição normal dos dados e, em razão disto, podem não representar adequadamente os dados analisados.

Portanto, neste trabalho foram estudadas e empregadas três técnicas de identificação de *outliers*, além da construção de um multiclassificador que une as respostas individuais. Os métodos Zscore e Zscore Modificado são técnicas simples baseadas na estatística e permitem grande velocidade computacional para o sistema de detecção. O algoritmo K-Means é uma técnica de clusterização que possibilita uma outra forma de classificação de *outliers*, baseando-se na distância aos centróides dos clusters obtidos. A combinação destas técnicas em um algoritmo multiclassificador mostrou-se, de fato, um sistema mais robusto para detectar *outliers*, tarefa muito importante no contexto IoT.

6.1 TRABALHOS FUTUROS

Como trabalhos futuros, é possível a implementação dos algoritmos individuais e do multiclassificador construído em um conjunto de dados já rotulado. Também, sugere-se o teste de outras técnicas de detecção de *outliers* e outros conjuntos de dados, comparando-se os resultados obtidos.

Além disso, para este trabalho, foi considerada a distribuição gaussiana nos métodos estatísticos implementados e na rotulação dos dados. Sugere-se que sejam testadas outras possibilidades de distribuições, como a distribuição bimodal, caso, por exemplo, da característica de umidade dos dados do dispositivo D3.

Para mais, propõe-se testar o multiclassificador com pares dos três algoritmos e analisar se as métricas melhoram ou pioram. Uma outra ideia é testar outras regras e pesos para compor o multiclassificador.

REFERÊNCIAS

BNDES. **Internet das coisas**: estimando impactos na economia. Acesso em 17 dez. 2021.

BRASIL. **Decreto nº 9.854, de 25 de junho de 2019.** Acesso em 12 jan. 2022.

BRASIL. **Carta Brasileira para Cidades Inteligentes norteará soluções tecnológicas em todo o Brasil**. Acesso em 14 jan. 2022.

Brasscom. **Relatório Setorial 2020 Macrossetor de TIC**. Acesso em 17 dez. 2021.

CAMPOS, G. O. **Estudo, avaliação e comparação de técnicas de detecção não supervisionada de outliers**. 2015. 87p. Mestrado em Ciências de Computação e Matemática Computacional — Universidade de São Paulo, São Carlos.

CHAUDHARY, S. et al. CRAIoT: concept, review and application(s) of iot. In: **2019 4th International Conference on Internet of Things**: smart innovation and usages (iot-siu). Ghaziabad, India: IEEE Xplore, 2019. p.1–4.

Cisco. **Cisco Annual Internet Report Complete Forecast Update, 2018–2023**. Acesso em 16 dez. 2021.

COSTA, G. de Barros Paranho da. **Detecção de anomalias utilizando métodos paramétricos e múltiplos classificadores**. 2014. 78p. Mestrado em Ciências de Computação e Matemática Computacional — Universidade de São Paulo, São Carlos.

FETTERMANN, C. et al. Uma sistemática para detecção de fraudes em empresas de abastecimento de água. **Asociacíon Interciencia**, [S.l.], v.40, n.2, p.114–120, 2015.

FREITAS, I. W. S. de. **Um estudo comparativo de técnicas de detecção de *outliers* no contexto de classificação de dados**. 2019. 99p. Mestrado em Ciência da Computação — Universidade do Estado do Rio Grande do Norte, Mossoró.

Gary Stafford. **Environmental Sensor Telemetry Data**. Acesso em 11 jan. 2022.

Gary Stafford. **Getting Started with IoT Analytics on AWS**. Acesso em 11 jan. 2022.

HAWKINS, D. **Identification of Outliers**. Dordrecht: Springer, 1980.

IDC. **IoT Growth Demands Rethink of Long-Term Storage Strategies, says IDC**. Acesso em 10 dez. 2021.

IESE Business School. **Cities em motion**. Acesso em 13 jan. 2022.

IGLEWICS, B.; HOAGLIN, D. **How to Detect and Handle Outliers**. Milwaukee: ASQC Quality Press, 1993.

Kaggle. **Kaggle**: your machine learning and data science community. Acesso em 5 nov. 2021.

MAGRANI, E. **A Internet das Coisas**. Rio de Janeiro: FGV Editora, 2018. 20-23p.

MARQUES, H. O. **Avaliação e seleção de modelos em detecção não supervisionada de outliers**. 2015. 84p. Mestrado em Ciências de Computação e Matemática Computacional — Universidade de São Paulo, São Carlos.

McKinsey Global Institute. **Unlocking the potential of the Internet of Things**. Acesso em 16 dez. 2021.

MILLS, G. Cities as agents of global change. **International Journal of Climatology**, [S.l.], v.27, p.1849–1857, 2007.

NASCIMENTO, R. M. do et al. Algoritmo de Detecção e Correção de *Outliers* para Previsão de Carga. In: IV SIMPÓSIO BRASILEIRO DE SISTEMAS ELÉTRICOS (SBSE). **Anais...** SWGE, 2012. Acesso em: 18/01/2022.

ONU. **ONU prevê que cidades abriguem 70% da população mundial até 2050**. Acesso em 15 dez. 2021.

Palo Alto Networks. **2020 Unit 42 IoT Threat Report**. Acesso em 16 dez. 2021.

PARKER, J. R. **Algorithms for Image Processing and Computer Vision**. Indianapolis: Wiley Publishing, 2011.

Planet Smart City. **Smart City Laguna**: o projeto piloto de croatá. Acessado em 20 out. 2021.

RODRIGUES, R. D. **Detecção de *outliers* baseada em caminhada determinística do turista**. 2018. 99p. Mestrado em Computação Aplicada — Universidade de São Paulo, Ribeirão Preto.

SANTOS, B. P. et al. Internet das Coisas: da teoria à prática. In: **Minicursos / XXXIV Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos**. Porto Alegre: Sociedade Brasileira de Computação (SBC), 2016. p.1–50.

SILVA FILHO, J. G. da. **Detecção de anomalias no tráfego MQTT de redes IOT utilizando técnicas de aprendizado de máquina**. Fortaleza, 2021. 68p.

SOUZA, T. I. A. de. **Métodos de detecção de *outliers* para o monitoramento ambiental de espaços urbanos inteligentes via análise multivariada e multidimensional**. 2020. 102p. Doutorado em Engenharia de Teleinformática — Universidade Federal do Ceará, Fortaleza.

TALARI, S. et al. A Review of Smart Cities Based on the Internet of Things Concept. **Energies**, [S.l.], 2017.

Techtudo. **'Internet das Coisas'**: entenda o conceito e o que muda com a tecnologia. Acesso em 12 jan. 2021.

UFSC. **Distribuição Normal (Gaussiana)**. Acesso em 16 jan. 2022.

USBERT, E. E. et al. Aplicação de técnicas de qualidade da informação em sensores na internet das coisas (IoT). **Revista Ifes Ciência**, [S.l.], v.7, n.1, p.01–18, 2021.

ZHANG, K. et al. Security and privacy in smart city applications: challenges and solutions. **IEEE Communications Magazine**, [S.l.], v.55, p.122–129, 2017.

NUP: 23081.008418/2022-21 **Prioridade:** Normal

Homologação de ata de defesa de TCC e estágio de graduação

125.322 - Bancas examinadoras de TCC: indicação e atuação

COMPONENTE

Ordem	Descrição	Nome do arquivo
11	Trabalho de conclusão de curso (TCC) (125.32)	Trabalho de Conclusão de Curso - Andressa Kreutz - Versao Final.pdf

Assinaturas

26/02/2022 11:22:30

NATANAEL RODRIGUES GOMES (PROFESSOR DO MAGISTÉRIO SUPERIOR)

07.09.14.00.0.0 - CURSO DE ENGENHARIA EM TELECOMUNICAÇÕES - CETEL