LEVI DA COSTA PIMENTEL

DETECÇÃO E CORREÇÃO DE OUTLIERS EM CURVAS DE DEMANDA DE ENERGIA UTILIZANDO REDES NEURAIS ARTIFICIAIS AUTOENCODERS

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica - PPGEE, da Universidade Federal da Paraíba - UFPB, como requisito parcial para a obtenção do título de Mestre em Engenharia Elétrica.

Orientador: Dr. Juan Maurício Moises Villanueva

JOÃO PESSOA

2023

LEVI DA COSTA PIMENTEL

DETECÇÃO E CORREÇÃO DE OUTLIERS EM CURVAS DE DEMANDA DE ENERGIA UTILIZANDO REDES NEURAIS ARTIFICIAIS AUTOENCODERS

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica - PPGEE, da Universidade Federal da Paraíba - UFPB, como requisito parcial para a obtenção do título de Mestre em Engenharia Elétrica.

Orientador: Dr. Juan Maurício Moises Villanueva

JOÃO PESSOA

2023

UNIVERSIDADE FEDERAL DA PARAÍBA – UFPB

CENTRO DE ENERGIAS ALTERNATIVAS E RENOVÁVEIS – CEAR

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA – PPGEE

A Comissão Examinadora, abaixo assinada, aprova a Dissertação

DETECÇÃO E CORREÇÃO DE OUTLIERS EM CURVAS DE DEMANDA DE ENERGIA UTILIZANDO REDES NEURAIS ARTIFICIAIS AUTOENCODERS

Elaborada por

LEVI DA COSTA PIMENTEL

como requisito parcial para obtenção do grau de

Mestre em Engenharia Elétrica.

COMISSÃO EXAMINADORA

PROF. DR. JUAN MOISÉS MAURICIO VILLANUEVA
Orientador – UFPB

Documento assinado digitalmente
gov.br YURI PERCY MOLINA RODRIGUEZ
Data: 31/03/2023 08:01:31-0300
Verifique em https://validar.iti.gov.br

PROF. DR. YURI PERCY MOLINA RODRIGUEZ
Examinador Interno – UFPB

Documento assinado digitalmente
gov.br IVANOVITCH MEDEIROS DANTAS DA SILVA
Data: 31/03/2023 05:45:15-0300
Verifique em https://validar.iti.gov.br

PROF. DR. IVANOVITCH MEDEIROS DANTAS DA SILVA
Examinador Externo – UFRN

João Pessoa/PB, 30 de março de 2023

SUMÁRIO

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

RESUMO

DETECÇÃO E CORREÇÃO DE OUTLIERS EM CURVAS DE DEMANDA DE ENERGIA UTILIZANDO REDES NEURAIS ARTIFICIAIS AUTOENCODERS

Um dos principais problemas encontrados em *Smart Grids* é a ocorrência de outliers, que podem corromper dados, modificando então as informações trazidas por eles, dificultando a tomada de decisão com base nestas informações por parte dos operadores do sistema elétrico. Portanto, este trabalho propõe uma metodologia integrada de detecção e correção de outliers, baseada em redes neurais artificiais. Mais especificamente, foi desenvolvido um sistema de detecção baseado em Autoencoders, com auxílio de uma camada softmax, e um sistema de correção baseado em Autoencoders. A metodologia proposta foi submetida a diversos cenários, utilizando dados de uma subestação real, onde avalia-se a influência da variação do número de outliers presentes no banco de dados, assim como da variação da amplitude destes, sobre o funcionamento dos algoritmos. Nos testes conduzidos, a técnica de detecção chegou a alcançar Acurácia e F-score superiores a 99,7% e 97,4%, respectivamente. A técnica de correção chegou a obter erro percentual absoluto médio MAPE de 1,42%, enquanto a raiz do erro médio quadrático se manteve, na maioria dos cenários avaliados, inferior a 0,15 MW, valor que representa cerca de 1,7% do valor máximo de potência disponível no banco de dados.

Palavras-chave: Redes Elétricas Inteligentes, Medidores Inteligentes, Valores Discrepantes, Detecção e correção de outliers, Inteligência Artificial, Redes Neurais Artificiais, RNA, Autoencoders.

ABSTRACT

DETECTION AND CORRECTION OF OUTLIERS IN ENERGY DEMAND CURVES USING ARTIFICIAL NEURAL NETWORKS AUTOENCODERS

One of the main problems encountered in Smart Grids is the occurrence of outliers, which can corrupt data, thus modifying the information brought by them, making it difficult for electrical system operators to make decisions based on this information. Therefore, this work proposes an integrated outlier detection and correction methodology, based on artificial neural networks. More specifically, a detection system based on Autoencoders was developed, with the aid of a softmax layer, and a correction system based on Autoencoders. The proposed methodology was contemplated in several scenarios, using data from a real substation, where the influence of the variation in the number of outliers present in the database, as well as the variation of their amplitude, on the functioning of the algorithms, is evaluated. In the tests performed, the detection technique achieved Accuracy and F-scores greater than 99.7% and 97.4%, respectively. The correction technique obtained MAPE mean absolute percentage error of 1.42%, while the root mean square error remained, in most of the evaluated scenarios, below 0.15 MW, a value that represents about 1.7% of the maximum power value available in the database.

Keywords: Smart Grids, Smart Meters, Outliers, Outlier detection and correction, Artificial Intelligence, Artificial Neural Networks, RNA, Autoencoders.

1 INTRODUÇÃO

1 INTRODUÇÃO

1.1 MOTIVAÇÃO E RELEVÂNCIA DO TRABALHO

Na maioria das aplicações, os dados são criados por um ou mais processos geradores, que podem refletir a atividade do sistema ou pode nos fornecer observações sobre componentes deste sistema. Ou seja, através da análise dos dados coletados de determinada aplicação, é possível extrair informações importantes que podem, por exemplo nos auxiliar a tomar decisões sobre a mesma (AGGARWAL, 2017).

Não é diferente no âmbito dos sistemas de geração e distribuição de energia elétrica. O constante aumento de demanda por energia elétrica, incentivado pelo crescimento natural da população mundial, faz com que usinas de geração de energia elétrica busquem aprimorar os processos de geração de energias a partir de fontes usuais (como, por exemplo a hídrica, proveniente das águas e rios), assim como buscam incluir em seus repertórios a geração de energia elétrica por meio de fontes renováveis (como, por exemplo, solar, proveniente do sol, eólica, fornecida pelos ventos e biomassa, extraída da matéria orgânica) (NETO, 2018).

.

Porém, para atender às crescentes e novas demandas de energia elétrica, com seus pré-requisitos intrínsecos como, por exemplo, de eficiência e confiabilidade, faz-se necessária a implantação de uma infraestrutura tecnológica e inteligente, a fim de transformar a atual estrutura em uma rede elétrica inteligente, popularmente conhecidas como Redes Elétricas inteligentes (ou Smart Grids, do inglês) (NETO, 2018; REZA et al., 2015). As redes elétricas inteligentes são compostas por uma infraestrutura de equipamentos eletrônicos dotados de uma tecnologia de comunicação digital com fluxo de dados bidirecional com o intuito de aperfeiçoar a eficiência energética da rede elétrica. Alguns desses equipamentos são denominados de medidores inteligentes, ou smart meters, os quais são capazes de medir, processar e transmitir dados referentes ao consumo de energia elétrica (NETO, 2018).

As distribuidoras de energia têm grande interesse nos dados coletados nas subestações como as curvas de demanda e potência de sua região de distribuição, tendo em vista que, com os dados adquiridos é possível construir um banco de dados

através do qual são desenvolvidas análises sobre a subestação e sobre a área de cobertura da mesma. De posse desses estudos, é possível desenvolver ou aprimorar os métodos de operação, ter maior controle do sistema elétrico além de facilitar a manutenção corretiva e preventiva. Estas medidas por parte das concessionárias são base para um planejamento estratégico e uso mais racional na alocação de recursos (ANDRADE, 2018)

Em qualquer sistema onde há fluxo de dados, há também a possibilidade de ocorrência de eventos que acarretam em alterações desses dados, fenômeno que pode ser indesejado. Nos sistemas de geração e distribuição de energia elétrica, os dados de demanda e potência podem ser afetados por diversos fatores, como por exemplo, erros de comunicação, manobras de chaves, fenômenos da natureza como a queda de um raio, queda de energia ou problemas de instabilidade na transdução da medida pelo sensor. Eventos assim podem gerar valores discrepantes se comparados com o comportamento padrão da curva de demanda (ANDRADE, 2018). Esses valores atípicos são conhecidos na literatura como outliers, anormalidades, discordantes, desviantes ou anomalias na literatura de mineração de dados e de estatística e constituem uma observação que se desvia tanto das outras observações a ponto de levantar suspeitas de que foi gerado por um mecanismo diferente (AGGARWAL, 2017).

Quando o sistema não se comporta da maneira usual, resultando na geração de outliers, estes podem carregar informações úteis sobre características anormais do sistema e entidades que impactam no processo de geração de dados (AGGARWAL, 2017).

Alguns autores diferenciam ruído e anomalia pelo fato de que o ruído, geralmente, parece se encaixar em algum outro padrão, que o não o dos dados de interesse, mas que está representado por outros dados distribuídos aleatoriamente, diferentemente de uma anomalia, que não parece se encaixar em nenhum padrão. Outros autores modelam o ruído como uma forma fraca de outlier (AGGARWAL, 2017).

Há semelhanças entre outliers e ruído e autores distintos podem defini-los diferentemente. Geralmente, ruído é algo que deve ser removido para facilitar a análise dos dados, e o outlier é um dado valioso, que pode conter alguma característica (FREITAS, 2019). Portanto, é o interesse de quem analisa os dados que

deve definir, de acordo com a aplicação, a diferença entre ruído e outlier (AGGARWAL, 2017).

Como a distinção entre ruído e outlier é definida pelo interesse do analista e a grande massa de algoritmos de detecção de outliers pode ser usada também para ruídos, vemos que a diferenciação é, então, muito mais semântica que técnica. Sendo assim, a melhor maneira de definir essa diferença na prática, é usar o feedback de exemplos previamente conhecidos de pontos discrepantes. Outro ponto importante é que as técnicas de detecção supervisionadas, em geral, têm desempenho bem superiores se comparadas com as técnicas não-supervisionadas, visto que usam as informações dos exemplos prévios de outlier para identificar novos exemplos (AGGARWAL, 2017). O presente trabalho usa um algoritmo supervisionado assim como não faz distinção entre outlier e ruído.

Diante desta conjuntura, é de grande valia o desenvolvimento de metodologias que tornem possível tanto a detecção quanto a substituição de outliers, seja por quê deseja-se obter informações dos dados em si, na sua forma mais pura possível, sem perder ou deturpar informações, ou por quê deseja-se isolar os outliers a fim de analisá-los para obter informações sobre comportamentos anormais do sistema.

Dada a relevância do tema, diversos trabalhos têm sido desenvolvidos a fim de atacar a problemática. A Tabela 1 resume alguns dos principais trabalhos desenvolvidos na área, que pode ser vista como um check list, na qual a segunda coluna indica se o trabalho fez uso de técnicas estatísticas, a terceira coluna indica se o trabalho fez uso de Sistemas de Inferência Fuzzy, a quarta coluna indica se o trabalho fez uso de Autoencoders, a quinta coluna informa se o trabalho utilizou modelos autorregressivos ARIMA, as sexta e sétima colunas indicam se o trabalho abordou técnicas de detecção e correção, respectivamente, e a oitava e última coluna informa se o trabalho contém uma análise quantitativa sobre o tema.

Nota-se, portanto, que a maioria dos trabalhos desenvolvidos focam, principalmente, em uma metodologia, (ou detecção ou de correção de outiliers) mas não em ambas; alguns fazem testes específicos (por exemplo, apenas com outliers do tipo zero e/ou pico), acabando por não fornecer testes e métricas suficientes de forma a verificar a robustez do método; há ainda possibilidade de exploração do tema,

por exemplo, na tentativa de melhorar alguns índices de desempenho obtidos por trabalhos anteriores, dentre outras questões.

TABELA 1: COMPARAÇÃO ENTRE ALGUNS DOS PRINCIPAIS TRABALHOS DESENVOLVIDOS NA ÁREA.

Ano/Título	EST	FUZZY	RNA	AE	ARIMA	DET	COR	QUANT
2022 / Time series anomaly detection in power electronics signals with recurrent and ConvLSTM autoencoders (RADAIDEH et. al., 2022).			✓	✓		✓		✓
2020 / An Outliers Processing Module Based on Artificial Intelligence for Substations Metering System (ANDRADE; VILLANUEVA; MACEDO, 2020).	✓	✓	✓		✓	✓	✓	✓
2020 / Probabilistic Deep Autoencoder for Power System Measurement Outlier Detection and Reconstruction (LIN; WANG, 2020).	✓		✓	✓		✓	✓	✓
2018 / *Big Data Analytics of Smart Grids using Artificial Intelligence for the Outliers Correction at Demand Measurements* (NETO; ANDRADE; VILLANUEVA; SANTOS, 2018)		✓	✓				✓	✓
2018 / *Outlier Data Treatment Methods Toward Smart Grid Applications* (SUN; ZHOU; ZHANG; YANG, 2018)	✓	✓	✓			✓		
2017 / *Outliers Discovery from Smart Meters Data Using a Statistical Based Data Mining Approach* (NEAGU; GRIGORAS; SCARLATACHE, 2017).	✓					✓		✓
2012 / *Outliers' Detection and Filling Algorithms for Smart Metering Centers* (NASCIMENTO et. al. 2012)	✓					✓	✓	✓

Fonte: Elaborada pelo autor

Neste cenário, este trabalho de pesquisa tem como objetivo o desenvolvimento de uma metodologia baseada em técnicas de inteligência artificial, principalmente Autoencoders, com a finalidade de detectar e corrigir outliers sob curvas de demanda de energia elétrica com alto percentual de acertos e baixa taxa de erro na fase de detecção, baixo erro percentual absoluto médio e baixo erro quadrático na fase de correção, levando em conta os resultados obtidos em trabalhos anteriores.

1.2 OBJETIVOS

Desenvolver uma metodologia baseada em técnicas de inteligência artificial, principalmente Autoencoders, com a finalidade de detectar e corrigir outliers sob curvas de demanda de energia elétrica.

1.2.1 Objetivos Específicos

- Avaliar o uso de Autoencoders para o desenvolvimento de técnicas de detecção e correção de outliers;
- O desenvolvimento de uma metodologia que auxilia na seleção de parâmetros do autoencoder de forma a selecionar o modelo com melhor desempenho, etapa que por muitas vezes é desenvolvida de forma empírica;
- O desenvolvimento de um sistema de pontuação capaz de eleger, dentro de um conjunto de configurações de um modelo estimativo, a que apresenta menores índices erro relativo máximo, erro absoluto máximo, a raiz quadrada do erro quadrático médio e o erro percentual absoluto médio;
- Melhorar alguns índices de desempenho em relação a alguns trabalhos anteriores: (ANDRADE et al. 2020), (LIN; WANG, 2020) e (RADAIDEH et al.; 2022).

1.3 ORGANIZAÇÃO DO TRABALHO

Além do capitulo de introdução, este trabalho está composto por mais 4 capítulos, descritos sucintamente a seguir:

No cap 2, será apresentada a fundamentação teórica, dos principais pontos da pesquisa, abordando temas como Smart Grids, Medidores Inteligentes, Outliers, Autoencoders, Métricas para Avaliação de Sistemas, dentre outros.

No cap 3, Metodologia, serão apresentados, individualmente, as técnicas detecção e correção de outliers desenvolvidas a partir de autoencoders, a metodologia de treinamento das redes neurais artificiais utilizadas, assim como os cenários de testes aos quais o sistema foram submetidos.

No capítulo 4, serão apresentados comparações, discussões e os principais resultados obtidos pelas técnicas desenvolvidas pelo trabalho.

Por fim, no capítulo 5, serão feitas as considerações finais com base nos resultados, e proposições sobre possíveis trabalhos futuros com base na metodologia proposta neste trabalho.

2 FUNDAMENTAÇÃO TEÓRICA

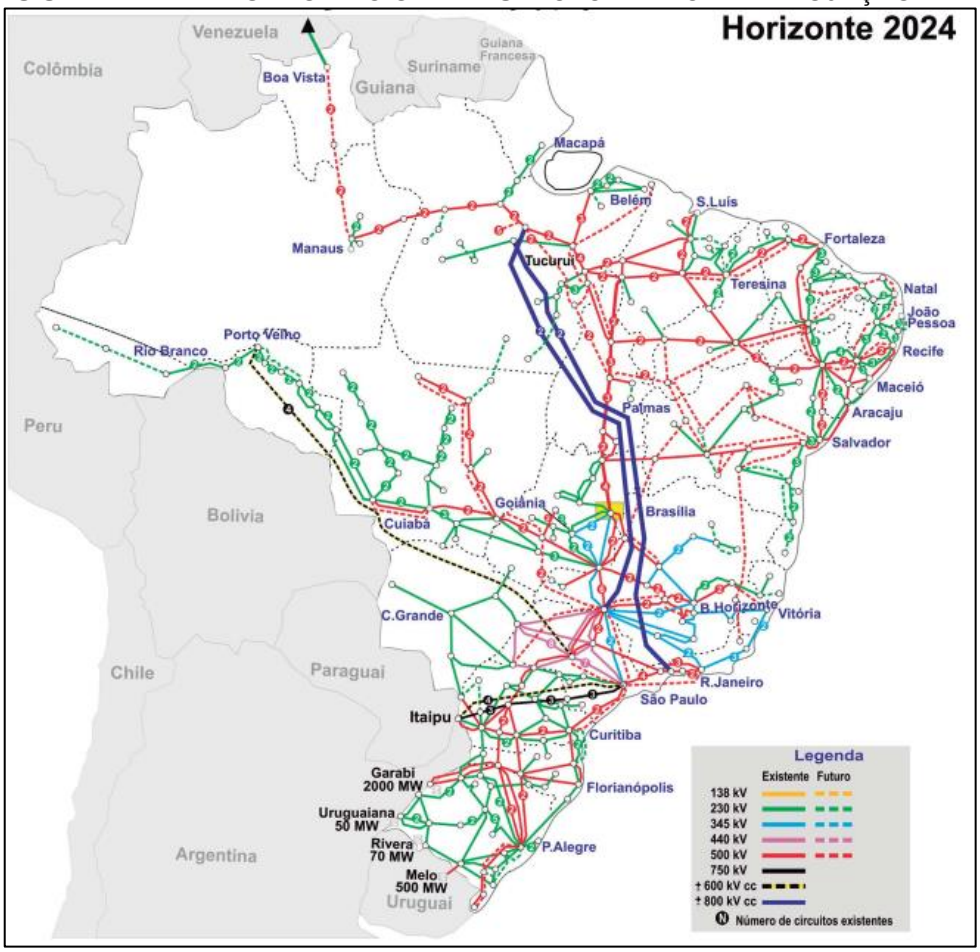
2 FUNDAMENTAÇÃO TEÓRICA

Nesta seção serão estudadas as definições fundamentais relacionadas ao tema da pesquisa, iniciando-se com algumas informações sobre o sistema interligado nacional. Em seguida, são apresentados alguns conceitos sobre redes elétricas inteligente, outliers e como estes ocorrem em smart grids. Para finalizar, são apresentados alguns conceitos sobre autoencoders, assim como são apresentadas as métricas que foram utilizadas para avaliação dos resultados da metodologia desenvolvida.

2.1 SISTEMA INTERLIGADO NACIONAL – SIN

Sistema Interligado Nacional (SIN) é o termo adotado para designar a rede elétrica básica brasileira em toda sua extensão, composto por uma matriz elétrica com 166,76 GW de capacidade instalada, sendo que 85% são de fontes renováveis. Em termos de geração de energia elétrica, em 2018, foi gerado um total de 601.396 GWh. Citando as quatro fontes com maior participação, observa-se 388.971 GWh (64,68%) provenientes de hidrelétricas, seguidas de térmicas a gás natural, com 54.622 GWh (9,08%), térmicas a biomassa, com 52.267 GWh (8,69%), e 48.475 GWh (8,06%) de usinas eólicas. Como resultado da participação predominante de fontes renováveis, tem-se uma intensidade de emissões de 88 kgCO2/MWh (SILVA, 2020).

FIGURA 1:SISTEMA INTERLIGADO NACIONAL: ESTRUTURA ATUAL E PROJEÇÃO PARA 2024.



Fonte: Silva (2020).

Os 141.756 km de linhas de transmissão que percorrem, praticamente, todo o Brasil, é um dos fatores que torna o SIN um dos maiores sistemas de transmissão do mundo, percorrendo o território nacional quase que em sua totalidade, excetuando-se apenas o estado de Roraima, como pode ser observado na Figura 1. É importante observar que a Figura 1 não representa apenas o cenário atual, considerando que a o levantamento fora feito no ano de 2020, mas também retrata a projeção para o ano de 2024 (SILVA, 2020).

O Operador Nacional do Sistema (ONS), órgão responsável pela coordenação e controle da operação das instalações de geração e transmissão de energia elétrica no Sistema Interligado Nacional (SIN) e pelo planejamento da operação dos sistemas isolados do país, sob a fiscalização e regulação da Agência Nacional de Energia Elétrica (Aneel) (ONS, 2022), prevê um aumento de extensão de 28% do SIN até 2024 (SILVA, 2020).

2.2 SMART GRIDS

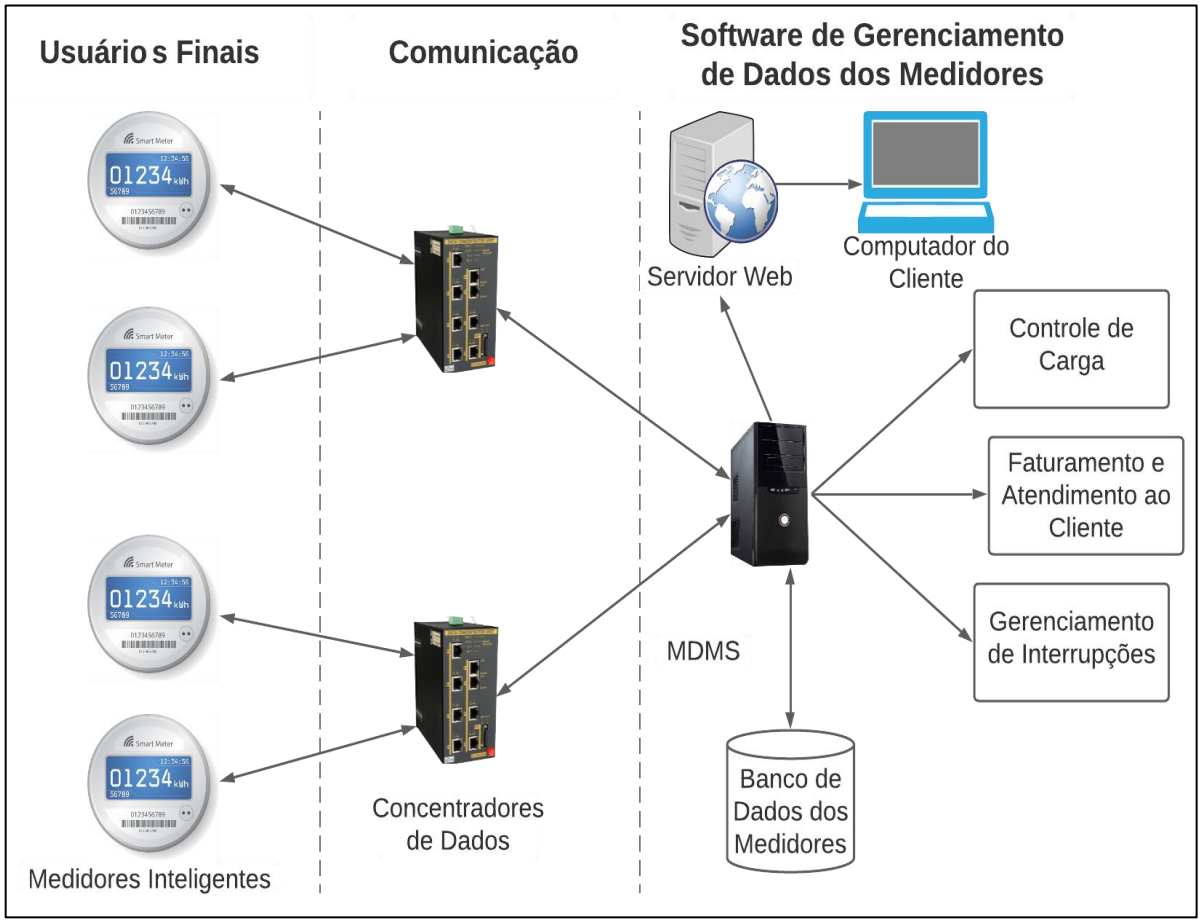
Dado o aumento da população mundial e da complexidade dos sistemas de geração distribuição de energia elétrica, como por exemplo o SIN representado na Figura 1, dentre outros fatores, cada vez mais se faz necessária a implementação de redes inteligente que sejam capazes, por exemplo, de transportar não só a energia em si, mas também informação de forma bidirecional.

Uma rede inteligente (ou smart grid) pode ser entendida como um ecossistema onde vários tipos de fontes de energia renováveis são conectados. Casas e edifícios inteligentes são equipados com instalações capazes de gerar energia elétrica para consumo próprio e compartilhar a energia excedente com a concessionária. (SUN; ZHOU; ZHANG; YANG, 2018)

Smart grid é um sistema ciber-físico com sistema de comunicação com a estrutura de fluxo de energia, com o intuito de dar inteligência e controle automatizado ao próprio sistema o que permite que não apenas o fluxo de energia, mas também o fluxo de informações. Os esquemas de suporte à comunicação e as técnicas de medição em tempo real da smart grid aumentam a robustez e a capacidade de previsão, além de oferecer proteção contra ameaças internas e externas. A rede inteligente usa infraestrutura de medição avançada (Advanced Metering Infrastructure - AMI) para coletar e processar informações de medidores inteligentes. Uma AMI consiste em três componentes básicos:

- dispositivos de medição inteligentes no usuário final (os smart meters);
- caminho de comunicação bidirecional entre o usuário final e a concessionária, onde encontramos os concentradores de dados, e
- software automatizado e centro de operação para processamento de dados (CHOI et al, 2021; BARAI et al, 2015; GUNGOR et al, 2013).

FIGURA 2: ESTRUTURA TÍPICA DE UMA AMI.



Fonte: Adaptado de Lisowski et al. (2019) e Zhou et al. (2012).

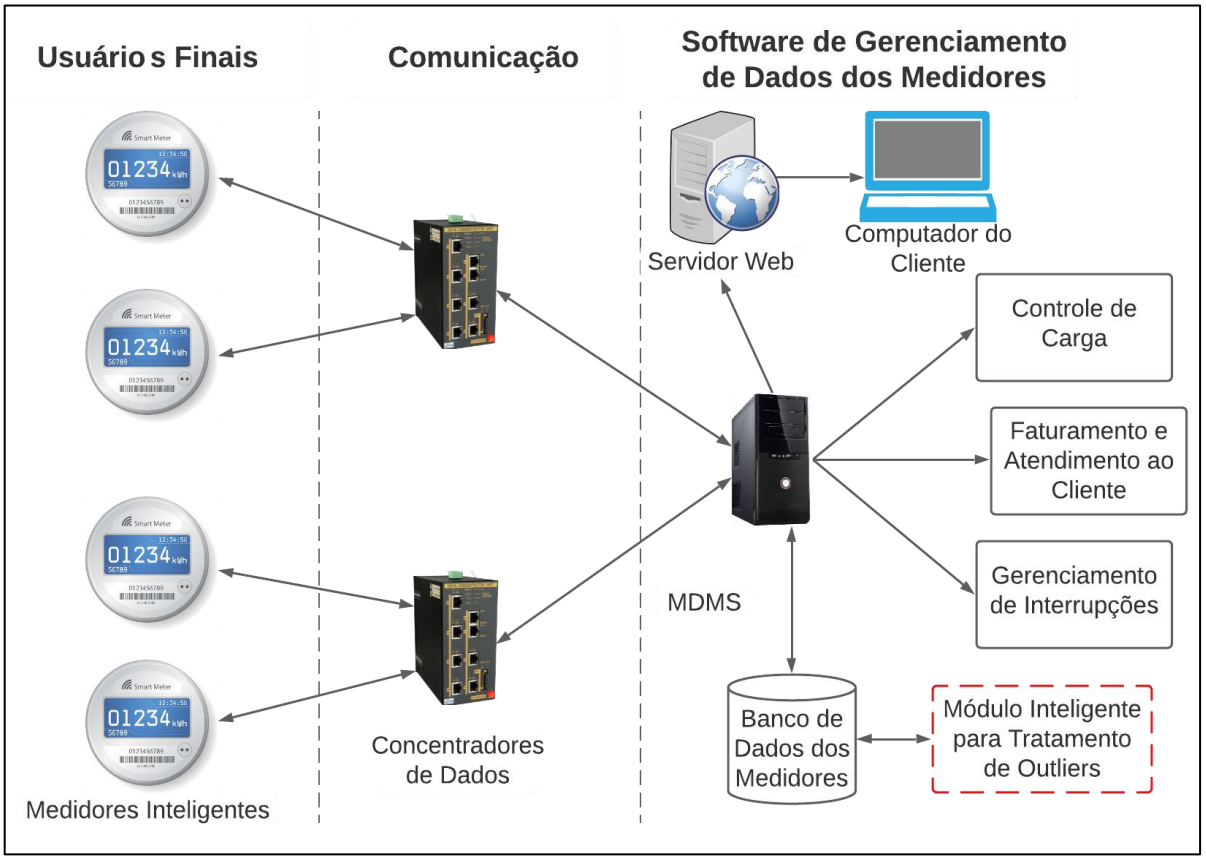
Uma AMI requer um sistema de gerenciamento de dados dos medidores (Meter Data Management System - MDMS) que, geralmente, tem arquitetura centralizada (CHOI et al, 2021; BARAI et al, 2015; GUNGOR et al, 2013).

Na Figura 2 pode ser vista a estrutura geral de uma AMI, onde foi destacada, na parte superior, o concentrador de dados que constitui um nó importante da AMI, visto que este concentra dados de vários smart meters de forma segura e os envia para o MDMS. Os MDMS são responsáveis pela coleta dos dados que entram no centro de operação a partir dos concentradores, pelo processamento e armazenamento desses dados, dispondo de ferramentas analíticas que permitem interação entre o MDMS, que constitui o coração de uma AMI, e as diferentes seções de operação e sistemas, como por exemplo o gerenciamento de interrupções, controle de carga, que fornece gerenciamento de qualidade de energia e previsão de carga, sistema de informações do consumidor que gerencia o faturamento de serviços e

informações do cliente, permitindo por exemplo, disponibilização de informações ao cliente assim como atendimento ao consumidor (CHOI et al, 2021; BARAI et al, 2015; GUNGOR et al, 2013).

Este trabalho propõe uma alteração na topologia do sistema de gerenciamento de dados dos medidores – MDMS, com a adição de um módulo inteligente para tratamento de outliers, atuando junto ao repositório de dados dos medidores, como indicado na Figura 3. Esta posição é estratégica visto que permite ao MDMS ter acesso aos dados originais, isto é, aos dados medidos, assim como aos dados corrigidos pela metodologia proposta, que pode então decidir qual dado usar para com determinada seção de operação ou sistema com o qual interage.

FIGURA 3: TOPOLOGIA PROPOSTA COM INSERÇÃO DO MÓDULO INTELIGENTE PARA TRATAMENTO DE OUTLIERS.



Fonte: Adaptado de Lisowski et al. (2019) e Zhou et al. (2012).

O módulo inteligente para tratamento de outliers consiste na metodologia para detecção e correção de outliers baseada em inteligência artificial, principalmente autoencoders, desenvolvida neste trabalho.

2.3 SMART METERS

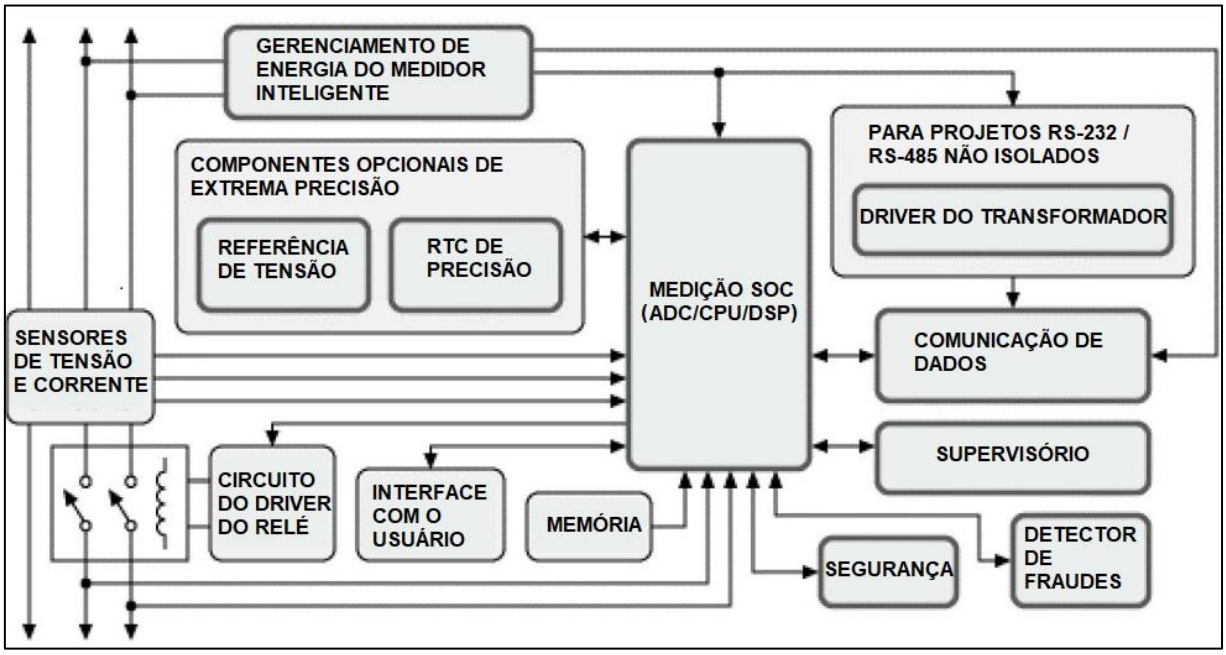
Uma Smart Meter é um dispositivo eletroeletrônico usado não apenas para medição, mas também para fornecer informações de faturamento dos clientes (para as concessionárias e para os próprios clientes), para a operação de seu sistema elétrico, constituindo parte fundamental de uma infraestrutura de uma Smart Grid (CHREN; ROSSI; PITNER, 2016).

Os medidores inteligentes devem possuir monitoramento confiável em tempo real, coleta automática de informações, devem permitir a interação do usuário e possuir dispositivo de controle de energia, ou seja, sempre que a demanda máxima de carga dos clientes ultrapassar seu valor de pico, o fornecimento de energia elétrica para o cliente poderá ser desconectado com auxílio do smart meter (BARMAN; YADAV; KUMAR; GOPE, 2018).

Os smart meters podem ser vistos como uma rede de sensores espalhados por toda extensão da rede elétrica cujas informações podem ser usadas para:

- Detecção, resposta e restauração de interrupções mais rápidas, fornecendo dados para as operações de campo em tempo hábil;
- Manter os clientes melhor informados sobre o status da rede elétrica, fornecendo, por exemplo, dados sobre a causa da interrupção, estimativa tempo para reestabelecimento do fornecimento;
- Melhorando a robustez contra interrupções, evitando possíveis interrupções, reduzindo a frequência e a duração das interrupções, aumentando a precisão do planejamento e gerenciamento da rede (BARAI, KRISHNAN, VENKATESH, 2015).

FIGURA 4: DIAGRAMA DE BLOCO DE UM TÍPICO SMART METER.



Fonte: Barai, Krishnan e Venkatesh (2015).

É possível ver o diagrama de blocos de um medidor inteligente na Figura 4 que, de uma forma geral, pode conter: Relógio em tempo real de alta precisão (RTC), módulo de comunicação de dados, sistema de medição no chip (System-on-chip - SoC), módulo de segurança, sistema de gerenciamento de energia, módulo de supervisão, detecção desvios fraudulentos, driver de transformador e referência de tensão (VREF). O centro de um hardware de medidor inteligente é baseado no processador SoC, incluindo a arquitetura para suportar as medições. O front-end analógico de um medidor consiste em conversores analógicos digitais (ADCs) que suportam entradas diferenciais. Um estágio de ganho adicionado em função de sensores com saída de baixa intensidade de sinal. Um multiplicador de hardware no chip SoC pode ser usado para acelerar as operações matemáticas intensivas durante o cálculo de energia. O software é capaz de calcular vários parâmetros, como por exemplo: corrente e tensão RMS, potência ativa e reativa, fator de potência, frequência, dentre outros (BARAI, KRISHNAN, VENKATESH, 2015).

2.4 OUTLIERS EM SMART GRIDS E ALGUNS ALGORITMOS

Diversos tipos de outliers com considerável complexidade podem ser gerados em redes de energia elétrica, o que se estende para as Smart-Grids. As principais

causas de dados discrepantes, segundo (SUN; ZHOU; ZHANG; YANG, 2018) são as seguintes:

- Capacidade de aquisição de dados. Os dispositivos de aquisição de dados, como medidores e sensores inteligentes, possuem desempenhos diferentes em frequência e diferentes níveis de precisão de aquisição de dados, o que pode levar a erros de medição. Ademais, ruídos podem ser gerados quando os dispositivos possuem baixa capacidade de filtrar interferências.
- Falhas no sistema de energia, como por exemplo, falhas no sistema de transmissão de dados, falhas nos equipamentos de transmissão de energia e falta de energia podem levar à geração de outliers.
- Fatores humanos. Algumas ações em sistemas de energia, como por exemplo, controle de desligamento manual, resposta a contingências são intervencionadas por humanos. Adicionalmente, humanos também estão envolvidos em alguns processos de coleta de dados, o que pode levar a geração de outliers.

Dado que os outliers estão presentes em smart-grid, é de extrema importância o desenvolvimento de técnicas que consigam identificá-los e substituí-los com a melhor precisão possível. Primeiramente, identificá-los é interessante visto que podem carregar informações sobre o comportamento anormal do sistema. Em segundo lugar, pode ser interessante apenas substituí-los por valores mais próximos do real possível, para que não impactem na análise dos dados, seja pela presença das anomalias ou pela ausência de dados (visto que em algumas abordagens, os outliers são apenas removidos).

As abordagens primárias podem ser caracterizadas como baseadas em distância (pontos mais distantes são considerados mais anômalos), baseadas em densidade (pontos que estão em regiões de densidade mais baixa são considerados mais anômalos) e baseadas em classificação (pontos mais anômalos são aqueles cujos vizinhos mais próximos têm outros como vizinhos mais próximos.) (MEHROTRA; MOHAN; HUANG, 2017)

Na literatura, existem alguns algoritmos que foram desenvolvidos para detecção de outliers, como por exemplo, Extreme Studentized Deviate – ESD, Z-Score, Test Box Plot, Thompson, Exponential smoothing (ExpSM). Algoritmos de

preenchimento de outliers geralmente envolveram diferentes cálculos de médias considerando diferentes combinações de períodos anteriores para obter uma estratégia capaz de produzir observações consistentes a partir do histórico da série. Porém, o problema reside no fato de que algumas técnicas tem melhor desempenho que outras, dependendo da natureza da série ou do outlier (NASCIMENTO; et al., 2012).

O presente trabalho avaliou o uso de Redes Neurais Artificiais, mais especificamente Autoencoders, na tarefa de detecção e correção de outliers.

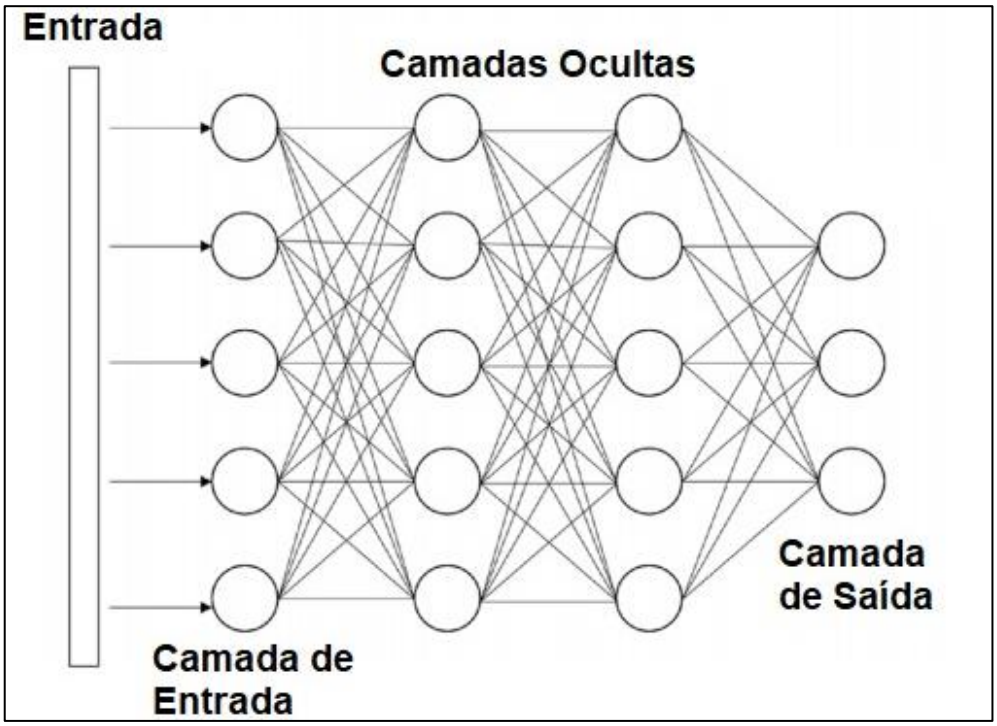
2.5 AUTOENCODERS

Redes neurais artificiais são camadas de nós interconectados, ou neurônios artificiais, que funcionam de maneira inspirada nas redes neurais biológicas. (ALLA; ADARI, 2019).

Na Figura 5 pode ser visto um exemplo de rede neural artificial, muitas vezes referenciada como perceptron multicamada (multilayer perceptron - MLP). Perceba que ela é formada pelas camadas de entrada, as camadas ocultas (neste exemplo, tem-se duas), que são as camadas internas, e a camada de saída. Cada camada é formada por um determinado número de nós (neurônios), que são, basicamente, somatórios ponderados com uma função de ativação na saída (ALLA; ADARI, 2019).

Autoencoders são redes neurais artificiais que conseguem aprender representações eficientes dos dados de entrada, chamadas de codificações, sem a necessidade de um conjunto de treinamento rotulado, isto é, não necessita de supervisão. Essas codificações geralmente têm uma dimensionalidade muito menor do que os dados originais de entrada, tornando-os ótimos redutores de dimensionalidade. Outra característica muito importante dos autoencoders é que eles podem atuar como detectores de características, habilitando-os a serem utilizados para pré-treinamento não supervisionado de redes neurais profundas. Autoencoders também são capazes de gerar novos dados muito semelhantes aos dados de treinamento, de forma aleatória (GÉRON, 2017). Essas características tornam o uso do autoencoder uma alternativa muito interessante na tarefa de detecção e substituição de outliers.

FIGURA 5: REPRESENTAÇÃO DE UMA REDE NEURAL ARTIFICIAL COM DUAS CAMADAS OCULTAS.



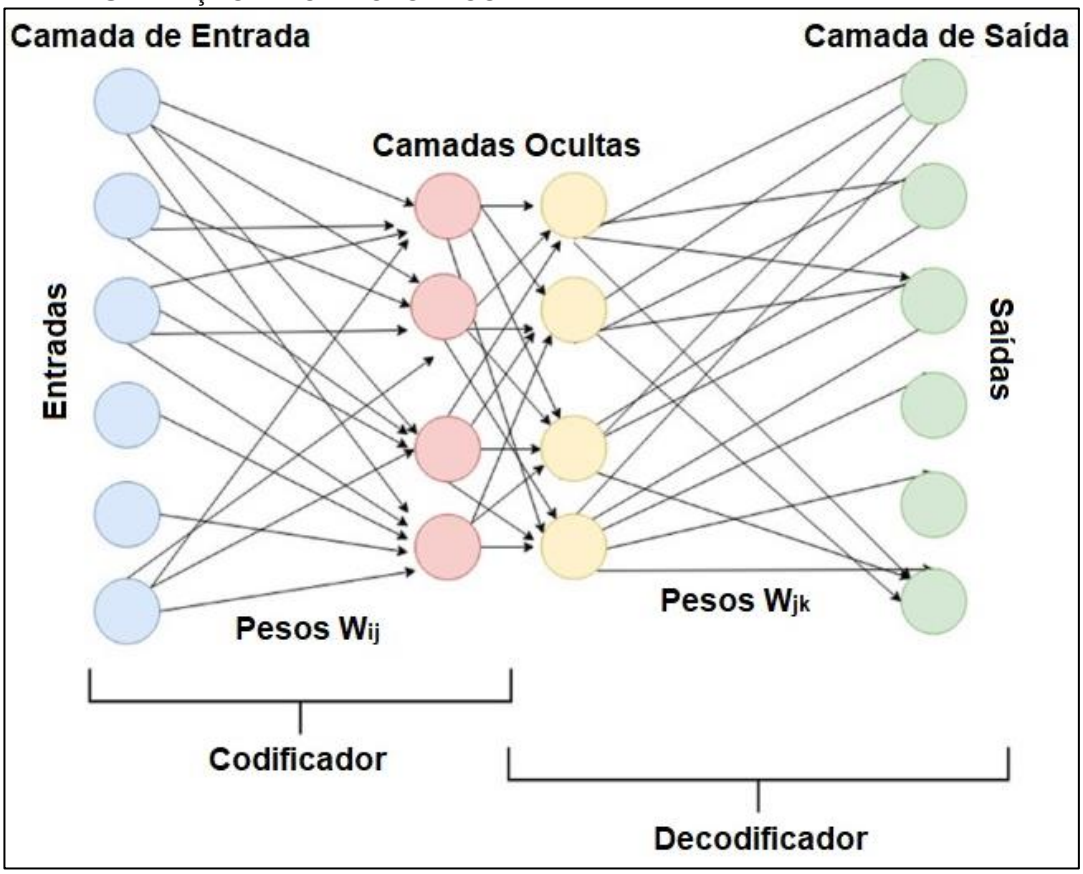
Fonte: Alla e Adari, 2019.

Como pode ser visto na Figura 6, um autoencoder é composto por um par de duas sub-redes conectadas, um codificador e um decodificador. O codificador recebe uma entrada e a converte em uma representação com menor dimensão e mais densa, também conhecida como representação latente da entrada, que a rede decodificadora pode usar para reproduzir a entrada o mais próximo possível da original (ALLA; ADARI, 2019).

Autoencoders podem ser classificados em algumas subcategorias de acordo com a forma que a restrições adicionais são aplicadas, dentre outras coisas.

Em um autoencoder esparso, por exemplo, adiciona-se uma restrição de esparsidade na atividade das representações ocultas, para que menos neurônios sejam disparados em um determinado momento. (ALLA; ADARI, 2019) Na prática, isso é feito adicionando-se um termo apropriado à função de custo, o que faz com que o autoencoder seja forçado a reduzir o número de neurônios ativos na camada de codificação (GÉRON, 2017).

FIGURA 6:REPRESENTAÇÃO DE UM AUTOENCODER



Fonte: Alla e Adari, 2019.

Outra forma de forçar o autoencoder a aprender recursos úteis é adicionando ruído aleatório às suas entradas na fase de treinamento, fazendo com que a rede recupere os dados originais sem ruído. Com isto, o autoencoder não pode simplesmente copiar a entrada para sua saída, e o chamado “autoencoder de remoção de ruídos” é forçado a aprender uma representação mais eficiente dos dados (ALLA; ADARI, 2019).

Já os autoencoders variacionais tentam aprender um modelo de variável latente para os dados de entrada, ou seja, ao invés de aprender uma função arbitrária, a rede neural aprende os parâmetros de uma distribuição de probabilidade modelando seus dados. Portanto, se forem amostrados pontos dessa distribuição, é possível gerar novas instâncias de dados que parecem ter sido amostradas do conjunto treinamento, razão pela qual os autoencoders variacionais são considerados modelos geradores (GÉRON, 2017; ALLA e ADARI, 2019).

Denomina-se rede neural profunda uma rede neural artificial possui duas ou mais camadas ocultas (GÉRON, 2017). Por consequência, pode-se estender a nomenclatura os autoencoders, surgindo então a classe dos deepautoencoders ou autoencoders profundos. É possível, então, realizar testes adicionando camadas internas, assim como pode-se mudar o número de neurônios das camadas internas, modificando a compressão, de forma a encontrar a melhor configuração para nossa aplicação. É necessário, contudo, cuidado de evitar alguns problemas, como, por exemplo, o overffinting, quando a rede está super ajustada para os dados de treinamento, tendo dificuldade de generalização quando o sistema se depara com dados inéditos. (GÉRON, 2017; ALLA e ADARI, 2019).

2.6 MÉTRICAS DE AVALIAÇÃO

Ao longo do desenvolvimento do trabalho, foram necessárias algumas métricas para avaliar o desempenho da metodologia desenvolvida. Do ponto de vista de detecção de outliers, utilizou-se o conceito de Matriz de Confusão. A Matriz de Confusão é uma matriz quadrada que resume todas as previsões feitas por um modelo. No conjunto de resultados possíveis temos:

- Positivo Verdadeiro **TP** (*True Positive*): ocorre quando o modelo prevê corretamente a classe que se pretende. Neste trabalho, um verdadeiro positivo ocorre quando determinada instância de dados é de fato um outlier e o subsistema de detecção de outliers o classifica como tal;
- Positivo Falso **FP** (*False Positive*): ocorre quando o modelo prevê incorretamente a classe pretendida. Neste estudo, um positivo falso ocorre quando determinada instância de dados não é um outlier e o subsistema de detecção de outliers o classifica como outilier;
- Negativo Verdadeiro **TN** (*True Negative*): ocorre quando o modelo prevê corretamente a classe que não se pretende. Neste trabalho, um negativo verdadeiro ocorre quando determinada instância de dados não é um outlier e o subsistema de detecção de outliers o classifica como tal;
- Negativo Falso **FN** (*False Negative*): ocorre quando o modelo prevê incorretamente a classe não pretendida. Neste estudo, um negativo

falso ocorre quando determinada instância de dados é um outlier e o subsistema de detecção de outliers não o classifica como outlier (TING, 2011).

Atrelados à Matriz de Confusão, surge o conceito de Acurácia, Precisão e Recall, parâmetros que são bastante úteis na avaliação de modelos de classificação (TING, 2011; GÉRON, 2017).

.

$$\boldsymbol{Acurácia} = \frac{Nº\ de\ Predições\ Corretas}{Nº\ Total\ de\ Predições} = \frac{\boldsymbol{TP+TN}}{\boldsymbol{TP+TN+FP+FN}} \quad (1)$$

$$\boldsymbol{Precisão} = \frac{Nº\ de\ Predições\ Corretas\ Positivas}{Nº\ Total\ de\ Predições\ Positivas} = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FP}} \quad (2)$$

$$\boldsymbol{Recall} = \frac{Nº\ de\ Predições\ Corretas\ Positivas}{Nº\ Total\ de\ Positivos\ Reais} = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FN}} \quad (3)$$

Geralmente, a Precisão e o *Recall* estão interligados de tal forma que, melhorar (aumentar) um, ocasiona uma piora (redução) no outro. Porém, para uma melhor avaliação da eficácia do modelo como um todo, é necessário levar ambos em consideração (TING, 2011; GÉRON, 2017).

Para tanto, podemos usar o conceito do ***F-score,*** que nada mais é que a média harmônica entre a Precisão e o *Recall* (GÉRON, 2017).

$$\boldsymbol{F\text{-}Score} = 2x\frac{\boldsymbol{Precisão x Recall}}{\boldsymbol{Precisão+Recall}} \quad (4)$$

Para exemplificar o uso desses conceitos, tomemos um exemplo hipotético, no qual um banco de dados possui 5 amostras (A_1, A_2, A_3, A_4 e A_5), das quais, 2 foram corrompidas (A_1 e A_4) e, portanto, representam outliers reais.

Suponha que este banco de dados foi submetido à metodologia proposta neste trabalho e que o subsistema de detecção de outliers classificou 2 amostras (A_3 e A_4) como outliers, e as restantes como normais, isto é, não corrompidas. Como apenas A_4 foi identificada corretamente (A_4 é de fato um outlier), esta previsão do

sistema representa um **TP** (positivo verdadeiro). A amostra A_3 foi identificada erroneamente, isto é, foi classificada como outlier pela metodologia proposta sem ser, de fato, um outlier real. Portanto, a previsão referente a A_3 representa um **FP** (positivo falso). Como o sistema deixou de identificar 1 dos outliers reais, isto é, classificou A_1 como normal, sendo ela um outlier, a previsão referente a A_1 corresponde a um **FN** (negativo falso). As amostras A_2 e A_5 foram classificadas pelo sistema de detecção como normais e, de fato, não foram deturpadas. Portanto, as previsões referentes a A_2 e A_5 representam dois **TN** (negativos verdadeiros).

TABELA 2: EXEMPLO DE UMA MATRIZ DE CONFUSÃO.

		Classe Prevista	
		É outlier	**Não é outlier**
Real	**É outlier**	TP: 1	FP: 1
	Não é outlier	FN: 1	TN: 2

Fonte: Elaborada pelo autor.

Resta agora encontramos o número de Negativos Verdadeiros. Sabemos que o modelo classificou 8 instâncias como normais, dentre as quais, há um falso negativo. Portanto, o modelo previu 7 Negativos Verdadeiros.

A partir dessas informações, é possível construir a Matriz de Confusão, representada na Tabela 2. Aplicando as equações de (1) a (4), obtém-se:

$$\boldsymbol{Acurácia} = = \frac{\mathbf{1+2}}{\mathbf{5}} = 60\%$$

$$\boldsymbol{Precisão} = \frac{\mathbf{1}}{\mathbf{1+1}} = 50\%$$

$$\boldsymbol{Recall} = \frac{\mathbf{1}}{\mathbf{1+1}} = 50\%$$

$$\boldsymbol{F\text{-}Score} = 2x\,\frac{\mathbf{50\%x50\%}}{\mathbf{50\%+50\%}} = 50\%$$

Do ponto de vista da substituição de outiliers, a estatística nos fornece alguns parâmetros que são amplamente usados, por exemplo, na predição de séries temporais, tais como erro relativo máximo, erro percentual absoluto médio (mean absolute percentage error - **MAPE**) erro quadrático médio, (mean square error – **MSE**),

dentre outros. (ABEDIN et al., 2017; ZHONG et al., 2020; MOSBAH; EL-HAWARY, 2015).

Neste trabalho, para avaliação dos resultados, foram usados o erro absoluto máximo **EAM**, erro relativo máximo **ERM**, **MAPE** e a raiz quadrada do erro quadrático médio (root mean square error - **RMSE**), que foram calculados para os pontos identificados como outliers.

Seja a saída verdadeira de um sistema qualquer representada por y. Seja y^l a saída do modelo através do qual pretende-se estimar a saída verdadeira. Sejam **y** e $\mathbf{y^l}$ os vetores que armazenam os valores assumidos por y e y^l, respectivamente, sendo que cada amostra, que corresponde a uma posição dos vetores, é denotada por y_i e y^l_i, respectivamente, onde i varia de 1 até o número total de amostras, N. Pode-se definir o erro absoluto, ou simplesmente erro, como y - y^l (ZHONG et al., 2020). Sendo assim, é possível escrever:

$$erro\ absoluto\ máximo, \boldsymbol{EAM} = ma\,x\,|(y_i - y_i')| \tag{5}$$

$$erro\ relativo\ máximo, \boldsymbol{ERM} = \max\ |(y_i - y_i')/y_i| \tag{6}$$

$$raiz\ quadrada\ do\ erro\ médio\ quadrático, \boldsymbol{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - y_i')^2} \tag{7}$$

$$erro\ percentual\ absoluto\ médio, \boldsymbol{MAPE} = \frac{1}{N}\sum_{i=1}^{N}|(y_i - y_i')/y_i|\,x100\% \tag{8}$$

Avanços relacionados a hardware, software e desenvolvimento de algoritmos possibilitaram avanços significativo na pesquisa científica (LANNELONGUE et al. ,2021). Porém, os impactos associados à computação em larga escala não têm sido estimados em sua totalidade. Os computadores onde funcionam os softwares e algoritmos necessitam de energia para funcionar e a geração desta energia está associada à geração de dióxido de carbônico (CO_2) e outros gases de efeito estufa (LIGHT,2017).

Diante disto, a fim de estimar o impacto ambiental dos algoritmos desenvolvidos neste trabalho em cenários que serão descritos mais adiante, duas métricas foram adotadas: a primeira é a “pegada de carbono”, cuja unidade de medida

é o dióxido de carbono equivalente (CO_2e) e mede o potencial de aquecimento global de uma mistura de gases de efeito estufa, representando a quantidade de CO_2 que teria o mesmo impacto no aquecimento global que a mistura a qual se deseja medir; a segunda unidade é a “mês-árvore”, que representa a quantidade de meses que uma árvore adulta, em média, demora pra absorver determinada quantidade de CO_2. Mais especificamente, considerou-se que uma árvore adulta, em média, leva 1 ano para sequestrar 11 kg de CO_2, o que equivale a dizer que ela equestra cerca de 0,92kg por mês. Ambas as métricas são calculadas a partir da energia estimada para execução do algoritmo em determinado hardware assim como o carbono gerado na produção desta energia em determinado local (LANNELONGUE et al., 2021).

Com isto, tem-se um conjunto de ferramentas matemáticas que tornam possível a análise da metodologia de detecção e correção de outliers desenvolvida neste trabalho.

3 METODOLOGIA

3 METODOLOGIA

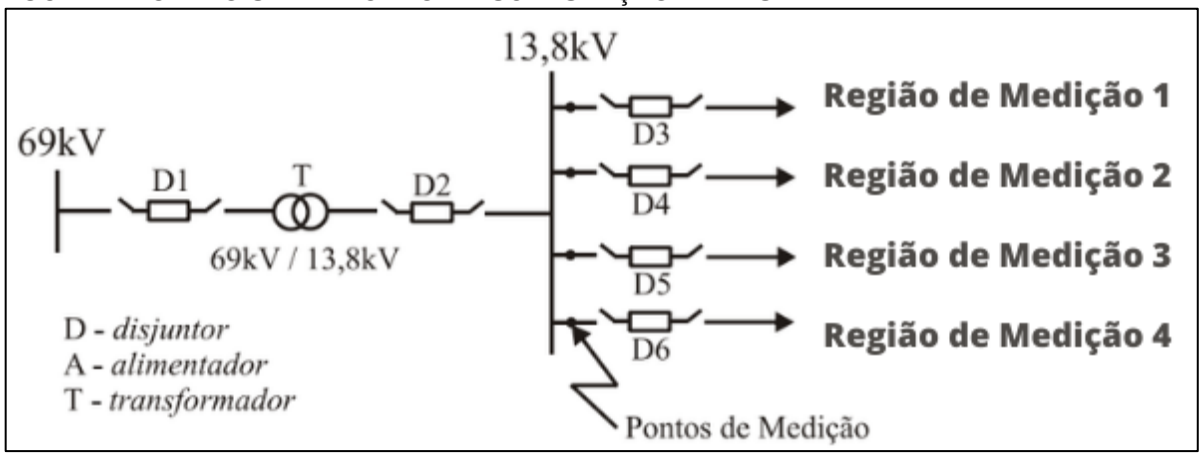
Neste capítulo, são apresentadas as técnicas de detecção e correção de outliers que foram desenvolvidas. A técnica de correção de outliers faz uso de autoencoders. A técnica de detecção é formada por autoencoders com uma camada adicional de softmax responsável por classificar um ponto qualquer da curva de carga como outlier ou não.

Adicionalmente, foi desenvolvido um algoritmo complementar para de injeção de outliers. A estratégia consiste em tomar dados assumidamente limpos, isto é, sem outliers, inserir dados discrepantes estrategicamente para treinar a rede de maneira que quando outliers ocorrerem de fato durante o funcionamento usual da subestação, o sistema seja capaz de identifica-los e substituí-los por valores o mais próximo possível dos valores reais.

3.1 ESTUDO DE CASO

As curvas de demandas foram obtidas de uma subestação localizada numa cidade do estado da Paraíba. As medições foram aferidas dos troncos dos quatro alimentadores presentes nesta subestação. Na Figura 7 é possível ver o modelo simplificado da subestação abaixadora de onde os dados foram obtidos, assim como os pontos de medição.

FIGURA 7:MODELO SIMPLIFICADO DA SUBESTAÇÃO ANALISADA



Fonte: Adaptado de Andrade (2018).

O intervalo entre medições é de 15 minutos, o que resulta em 96 amostras diárias. O banco de dados usado nos testes possui um total de 199080 amostras que compreende o período de janeiro de 2008 a setembro de 2013, perfazendo o período de pouco mais que 5 anos e 8 meses.

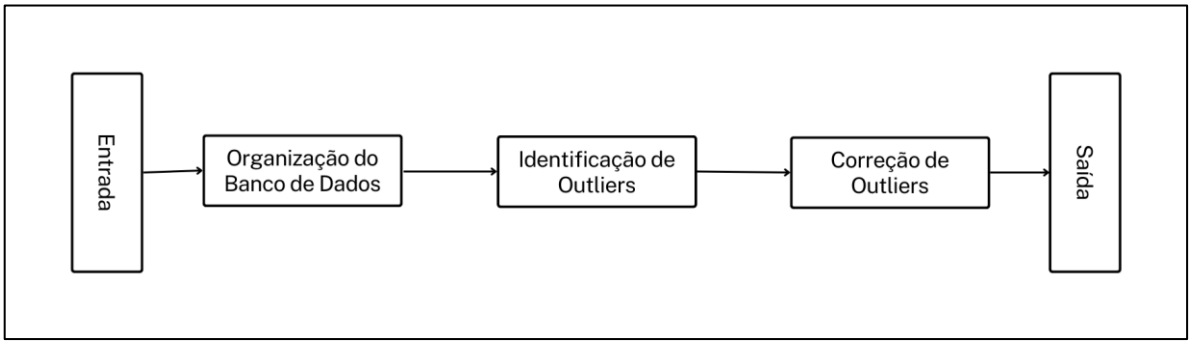
Alguns trabalhos relacionados ao tema, se concentram apenas no tratamento de outliers em outliers do tipo pico (spike), ou zero (ou ausência de dados), como por exemplo em NASCIMENTO et al. (2012), NETO et al. (2018) e ANDRADE et al. (2020). Porém, neste trabalho, não fizemos distinção entre ruído, o significa que os outliers inseridos podem assumir qualquer valor dentro de um determinado intervalo, o que acaba dificultando a tarefa de detecção, visto que estes podem estar muito próximos dos dados originais.

Mais especificamente, foram inseridos uma quantidade de outliers menor ou igual a dez porcento (10%) da quantidade total de amostras em posições do vetor de dados de entrada escolhidas aleatoriamente, com distribuição uniforme, estando os valores assumidos pelos outliers, em módulo, compreendidos entre 0% e 200% do valor máximo de potência presente no banco de dados. As amplitudes dos outliers também são escolhidas aleatoriamente, dentro deste intervalo, e estão distribuídas uniforme.

3.2 ALGORITMOS DE DETECÇÃO E CORREÇÃO DE OUTLIERS PROPOSTOS

De maneira simplificada, a metodologia proposta para a detecção e correção de outliers é composta por três módulos: a) organização do banco de dados; b) detecção de outliers; c) correção de outliers, cuja representação pode ser vista na Figura 8. A seguir, estes módulos serão descritos sucintamente.

FIGURA 8: DIAGRAMA DE BLOCOS DA METODOLOGIA PROPOSTA.



Fonte: Elaborada pelo autor.

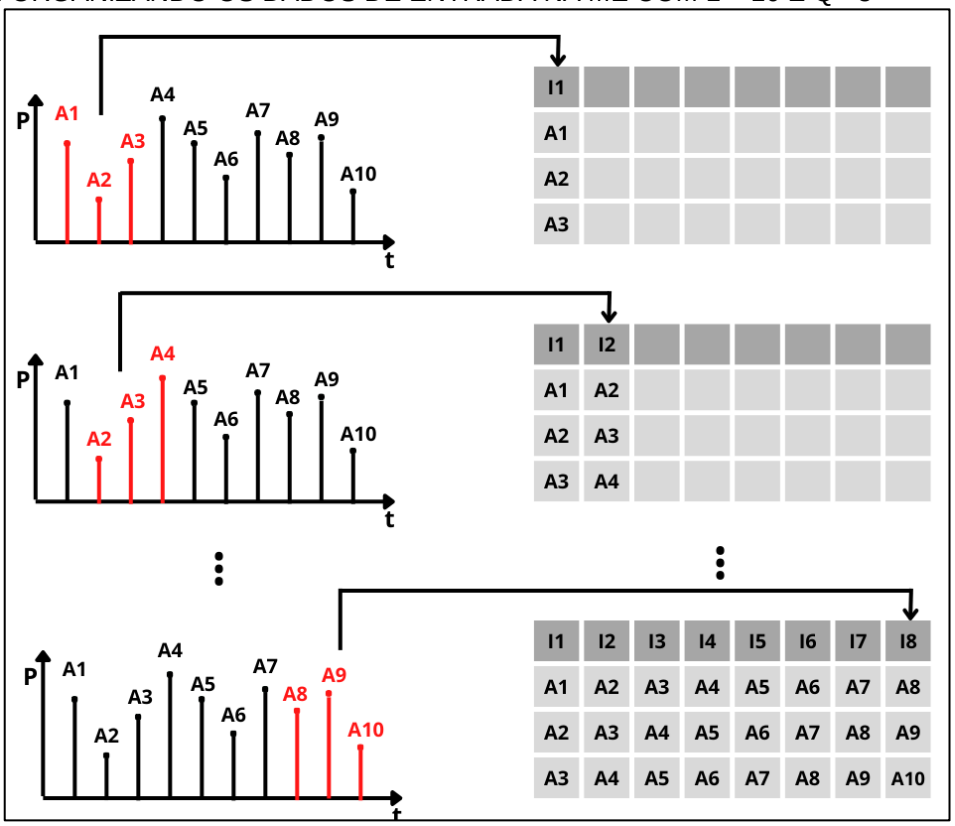
3.2.1 Organização do banco de dados

Primeiramente, os dados passam por uma etapa prévia na qual são organizados em uma matriz que será denominada de Matriz de Entrada - **ME** - de maneira que cada coluna possui um número **q** de amostras. A primeira coluna é preenchida com as amostras de 1 a **q**; a segunda coluna recebe as amostras de 2 até **q**+1; a enésima coluna contém as amostras de n até **q** + n – 1... desta forma, para um banco de dados com um total de **L** amostras, toda a matriz é preenchida com as amostras disponíveis de forma que a matriz possuirá q linhas e L – q + 1 colunas. Com isso, cada instância de dados do autoencoder possuirá **q** amostras e a rede neural será capaz de considerar informações de amostras vizinhas no momento de classificar um ponto como outlier assim como na correção deste. Perceba que **q** determina o número de entradas do autoencoder, isto é, o número de entradas do autoencoder é igual ao número de linhas da **ME**. A Figura 9 exemplifica o preenchimento desta matriz para um número total de **L** = 10 amostras e **q** = 3 amostras por instância. Cada coluna representa uma instância (de I1 até I8) de dados onde cada instância possui 3 amostras (de A1 a A3; de A2 a A4, ..., A8 a A10).

Após a etapa de preenchimento da **ME**, os dados de entrada passam pelo módulo de detecção de outliers, que por sua vez, fornece os dados ao módulo de correção de outliers com a informação da localização dos pontos identificados como

outliers, assim como faz a substituição prévia desses dados, auxiliando o módulo seguinte na tarefa de correção.

FIGURA 9: ORGANIZANDO OS DADOS DE ENTRADA NA ME COM L = 10 E Q =3



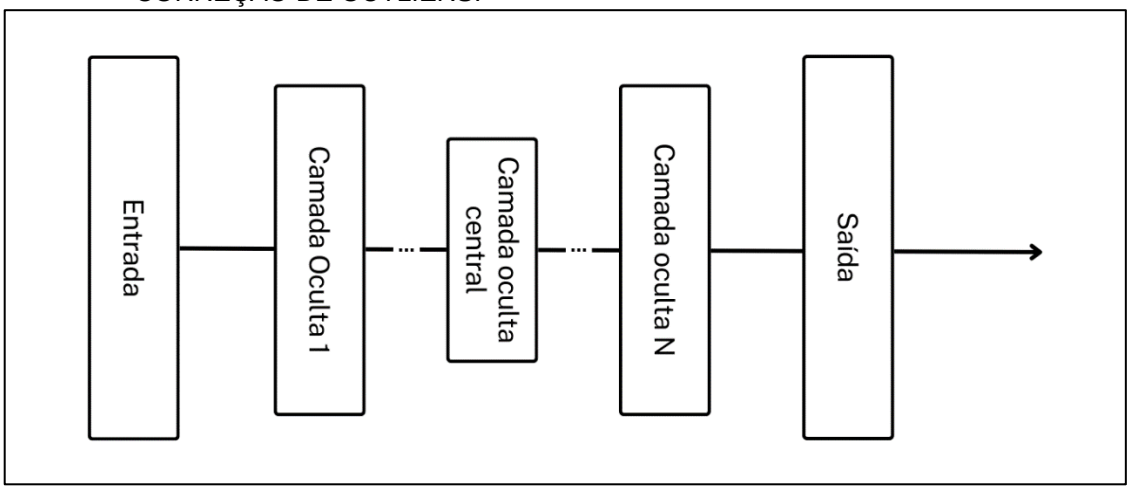
Fonte: Elaborada pelo autor.

Por fim, a título de comparação, para ambos os algoritmos desenvolvidos, foi também aplicada uma técnica tradicionalmente usada na literatura: Para o algoritmo de detecção, foi utilizada uma estratégia baseada no desvio padrão e, para o algoritmo de correção, foi utilizada a técnica de interpolação linear.

3.2.2 Algoritmo de Detecção de Outliers

Para o módulo de detecção de outliers, foi utilizado um autoencoder esparso com N camadas internas, cujo diagrama pode ser visto na Figura 10, e uma camada de softmax. A rede neural artificial completa pode ser vista na Figura11.

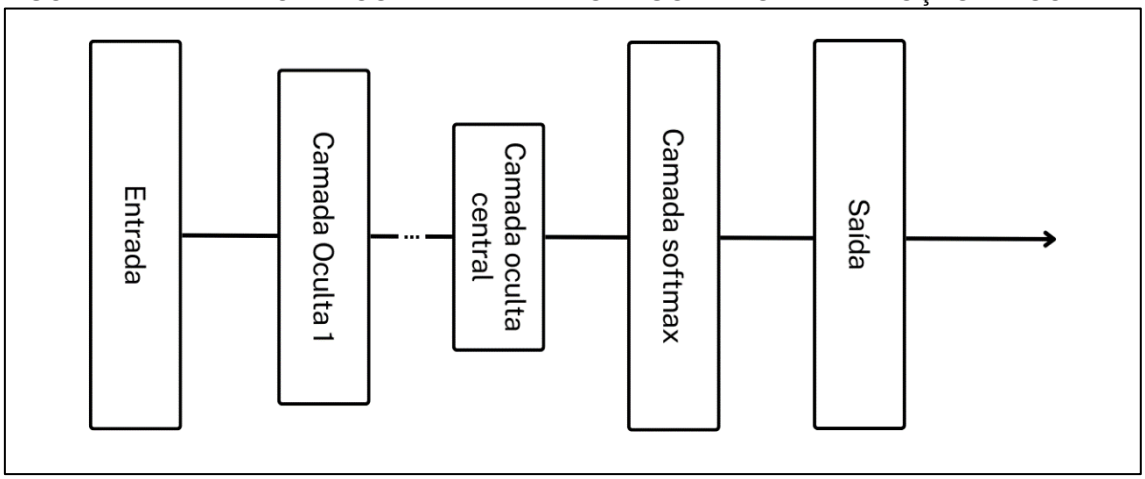
FIGURA 10: DIAGRAMA DO AUTOENCODER UTILIZADO NO ALGORITMO DE DETECÇÃO E CORREÇÃO DE OUTLIERS.



Fonte: Elaborada pelo autor.

A camada softmax é responsável por fornecer, em sua saída, a probabilidade de determinada instância de entrada pertencer à determinada classe. Neste trabalho, há a necessidade de uso uma classe apenas, a dos dados que pertencem a classe dos outliers. Portanto, a saída da rede neural fornece, para cada instância de entrada, um valor entre zero e um que indica a probabilidade dessa instância ser um outlier.

FIGURA 11: REDE NEURAL COMPLETA PARA O ALGORITMO DE DETECÇÃO DE OUTLIERS.



Fonte: Elaborada pelo autor.

Posteriormente, escolhe-se uma probabilidade de corte **pc** (neste trabalho fixada em 0,5) para classificar o dado como outlier, isto é, para valores maiores que a probabilidade de corte, a respectiva entrada é considerada um outlier.

Como abordou-se no tópico 2.4 do capítulo anterior, dentre as várias características dos autoencoders que são úteis na aplicação desenvolvida no presente trabalho, é interessante destacar o fato de que os autoencoders são ótimos detectores de características, o que os torna úteis no pré-treinamento não supervisionado (isto é, sem a presença de labels), característica que foi usada na rede neural que compõe o subsistema de detecção de outliers (GÉRON, 2017). Mais especificamente, num primeiro momento, o autoencoder foi treinado sem supervisão alguma. Depois, utilizou-se a representação latente do autoencoder presente na camada central oculta central (ver Figura 10) para treinar a rede softmax, situação onde tem-se um treinamento supervisionado, visto que é necessário fornecer um vetor de labels à camada softmax. Para finalizar o treinamento da rede neural completa, após empilhar-se o codificador do autoencoder com a camada de softmax (ver Figura 11), executou-se o treinamento supervisionado novamente, num procedimento chamado de ajuste fino (MathWorks, 2022).

Para selecionar os principais parâmetros da rede neural, procedeu-se da seguinte maneira:

1. Primeiramente, foram realizados alguns testes para selecionar o número de camadas internas do autoencoder. Mais especificamente, foram adicionadas camadas ocultas aos autoencoders do algoritmo de detecção a fim de selecionar a configuração com melhor desempenho;
2. Uma vez fixado o número de camadas internas, mantendo-se fixo o valor de entradas do autoencoder, que por sua vez é definido pelo número de colunas **q** escolhido na etapa de pré-processamento, variou-se o tamanho (número de neurônios) da(s) camada(s) interna(s), a fim de avaliar a influência do tamanho da camada interna no algoritmo de detecção;
3. Em seguida, variou-se o número de entradas **q** do autoencoder e repetiu-se o procedimento descrito no passo anterior;
4. Por fim, dentre todas as configurações, selecionou-se a que apresentou o melhor desempenho.

Com o número de entradas, o número de camadas internas e o número de neurônios de cada camada da rede neural artificial definidos, foi iniciada a avaliação

em alguns cenários do ponto de vista do número de outliers inseridos **NO**, tomando como referência o número total de amostras **L**:

A. O número de outliers corresponde a cerca de 2% de **L**;
B. O número de outliers corresponde a cerca de 4% de **L**;
C. O número de outliers corresponde a cerca de 5% de **L**;
D. O número de outliers corresponde a cerca de 6% de **L**;
E. O número de outliers corresponde a cerca de 8% de **L**;
F. O número de outliers corresponde a cerca de 10% de **L**.

Em seguida, considerando o número de outliers inseridos cujo algoritmo obteve o melhor desempenho, dentre os cenários supracitados, o subsistema de detecção foi avaliado em cenários onde variou-se a amplitude dos outlers inseridos, tomando como referência **Pmax** (máxima amplitude de potência disponível no banco de dados):

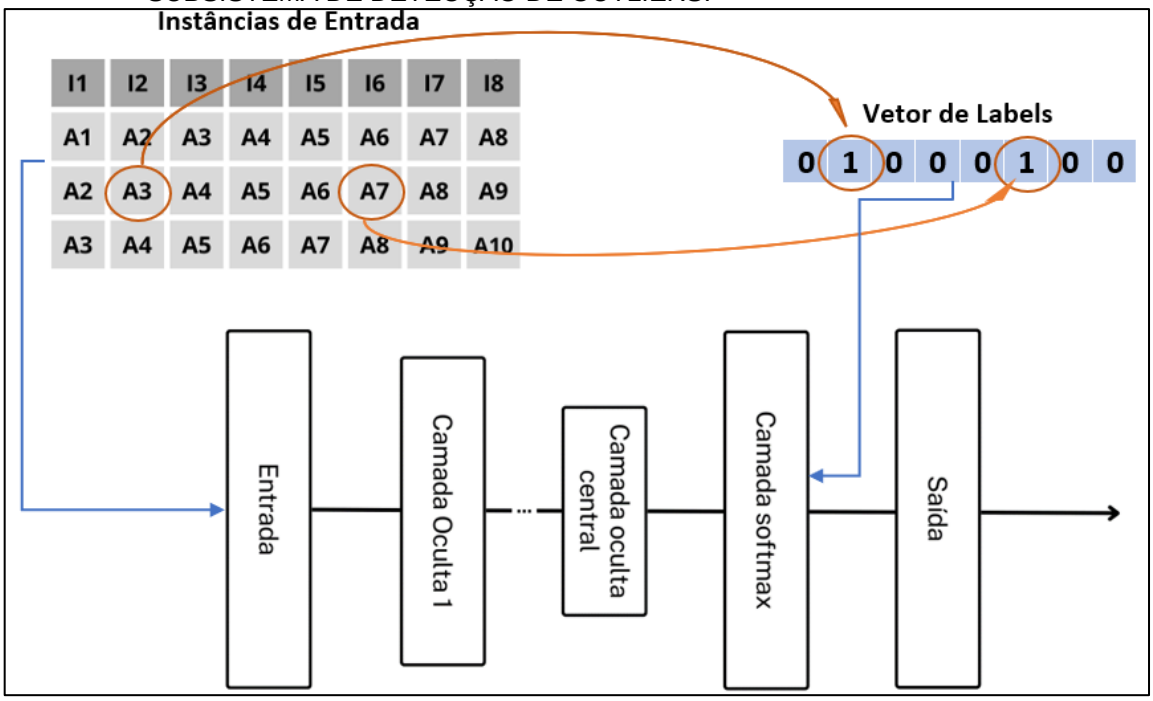
G. A amplitude dos outliers inseridos variam de 0 a 1% de **Pmax** em módulo;
H. A amplitude dos outliers inseridos variam de 0 a 25% de **Pmax** em módulo;
I. A amplitude dos outliers inseridos variam de 0 a 50% de **Pmax** em módulo;
J. A amplitude dos outliers inseridos variam de 0 a 75% de **Pmax** em módulo;
K. A amplitude dos outliers inseridos variam de 0 a 100% de **Pmax** em módulo;
L. A amplitude dos outliers inseridos variam de 0 a 200% de **Pmax** em módulo;
M. A amplitude dos outliers inseridos variam de (**u** – 3**dp)** até (**u** + 3**dp**).

Na busca de um cenário que dificultasse consideravelmente a tarefa de detecção de outliers, submetemos o algoritmo de detecção ao cenário M, onde **u** e **dp** representam, respectivamente, a média e o desvio padrão dos dados de treinamento. Para cada um desses cenários, a rede foi treinada considerando outliers distribuídos uniformemente. Para tanto, foi fornecido um vetor de labels que informa à rede neural se determinada instância de entrada é ou não um outlier. Mais especificamente, para cada instância de dado que é um outlier, o vetor de label conterá, na posição

corresponde, o valor 1, caso contrário, o vetor de label conterá o valor 0 na posição correspondente.

Para exemplificar, considere novamente o banco de dados hipotético descrito na Figura 9. Podemos olhar para a linha central de amostras (destacada em vermelho) como o vetor que contém os dados de interesse. As linhas superiores contêm valores passados e as linhas inferiores contêm valores futuros. Na Figura 12 é possível ver a representação do processo de treinamento da rede neural artificial do subsistema de detecção de outlier. Suponha que outliers foram inseridos nas amostras A3 e A7.

FIGURA 12: REPRESENTAÇÃO DO PROCESSO DE TREINAMENTO DA REDE NEURAL DO SUBSISTEMA DE DETECÇÃO DE OUTLIERS.



Fonte: Elaborada pelo autor.

O vetor de labels que precisa ser fornecido à rede neural, está representado na parte superior direita da Figura 12, destacado em azul, onde cada posição do vetor que corresponde a uma instância que contém um outilier na amostra central, é preenchida com o valor “1”, enquanto todas as posições do vetor de labels que correspondem às instâncias que representam pontos de dados sem outliers, são preenchidas com “0”.

Por fim, os resultados obtidos foram comparados com um algoritmo de detecção de outliers baseado no desvio padrão. Basicamente, foram calculados a

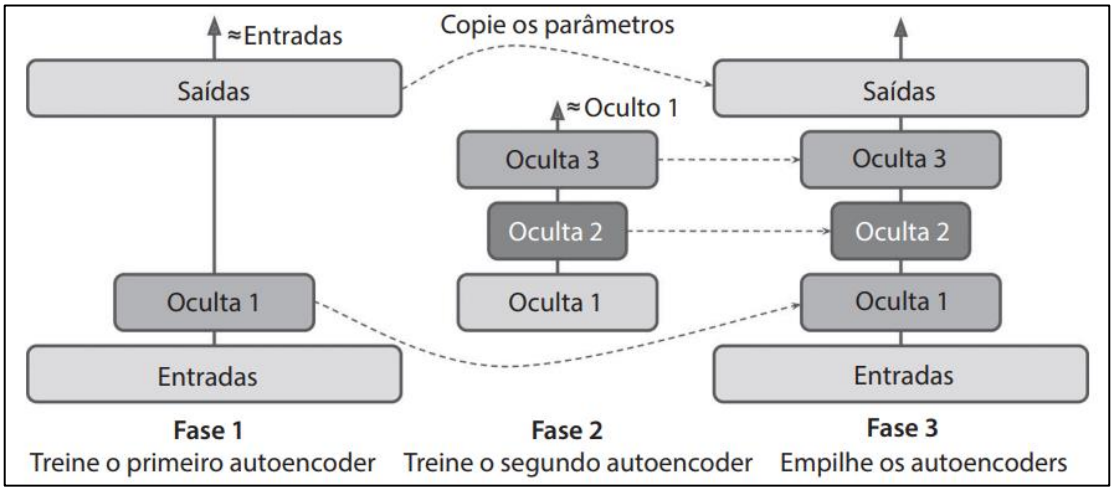
média e o desvio padrão do mesmo banco utilizado para treinamento da rede neural do subsistema de detecção. Então, o algoritmo calcula a diferença entre um novo dado a ser testado e a média já calculada. Caso a diferença seja maior que, por exemplo, o triplo do desvio padrão, o dado é considerado um outlier. Isso é possível devido ao fato de que numa distribuição normal, 99,73% das medidas estão dentro do intervalo de 3 desvios padrão, em módulo, em torno da média. Para o intervalo de 4 desvios padrão, em módulo, em torno da média, a probabilidade de cobertura é de 99,994%. Para este trabalho, o algoritmo auxiliar para fins de comparação utilizou o dobro e o quadruplo do desvio padrão.

3.2.3 Algoritmo de Correção de Outliers

Para o subsistema de correção de outliers, utilizou-se a característica mais fundamental do autoencoder, na qual ele tenta copiar a função identidade, sendo forçado a aprender uma representação mais eficiente dos dados de entrada para reconstruí-los na saída. O algoritmo de correção de outliers é, portanto, formado por um autoencoder esparso, que pode ser visto na Figura 10.

Na rede neural presente no algoritmo de detecção de outliers, foi necessário o treinamento supervisionado, no qual forneceu-se um vetor de labels informando ao sistema quais pontos deveriam ser considerados como outliers. Para o algoritmo de detecção de outliers, está presente o conceito de treinamento não supervisionado, isto é, quando não é necessário fornecer um vetor de labels, visto que o algoritmo é formado por um autoencoder puro (GÉRON, 2017).

FIGURA 13: TREINANDO UM AUTOENCODER DE CADA VEZ.



Fonte: Géron, 2017.

Para treinamento os autoencoders utilizados neste trabalho, utilizou-se a estratégia de treinar um autoencoder por vez (nas situações em que utilizamos autoencoder empilhados) descrita em (GÉRON, 2017).

Na Figura 13 é possível ver a representação desta estratégia de treinamento empihando-se dois (2) autoencoders. Procedendo-se dessa maneira, é possível treinar a rede completa muito mais rápido que se fossemos treinar todo o autoencoder profundo de uma vez, o que se torna mais evidente à medida que vamos adicionando mais camadas internas ao autoencoder. Na primeira etapa de treinamento, o primeiro autoencoder aprende a reconstruir as entradas do banco de dados; posteriormente, o segundo autoencoder recebe como entrada a representação compacta do primeiro autoencoder, ou seja, ele aprende a reconstruir a saída da camada oculta do primeiro autoencoder. Por fim, podemos montar o autoencoder profundo empilhando-se, primeiro, as camadas ocultas de cada autoencoder e, em seguida, as camadas de saída na ordem inversa (GÉRON, 2017).

Após a passagem dos dados pelo algoritmo de detecção, as posições do vetor de entrada onde foram identificados outliers já são conhecidas. O algoritmo de detecção pode, então, passar os dados para o algoritmo de correção substituindo os outliers por outro valor conveniente, auxiliando, então o algoritmo de correção de outliers a reconstruir os dados de entrada.

Neste sentido, foram testadas três estratégias, nas quais os pontos identificados como outliers são substituídos:

- por 0;
- pela média dos dados de treinamento;
- pela interpolação linear.

Para selecionar o número de camadas ocultas do autoencoder que compõe o sistema de detecção, procedemos de maneira análoga à descrita no passo 1 do tópico 3.1 considerando as 3 estratégias de passagem de dados supracitadas. Para selecionar os outros parâmetros como o número de neurônios das camadas ocultas, número de amostras por instância (isto é, o número de entradas do autoencoder), dentre outros, procedemos de forma análoga a descrita nos passo de 2 a 4 do tópico 3.1. Da mesma forma, os procedimentos adotados para o algoritmo de detecção foram mantidos, o que significa que o algoritmo de correção de outliers foi submetido aos mesmos cenários aplicados ao algoritmo de detecção de outliers, a saber, os descritos nos subtópicos de A até L descritos no tópico 3.1.

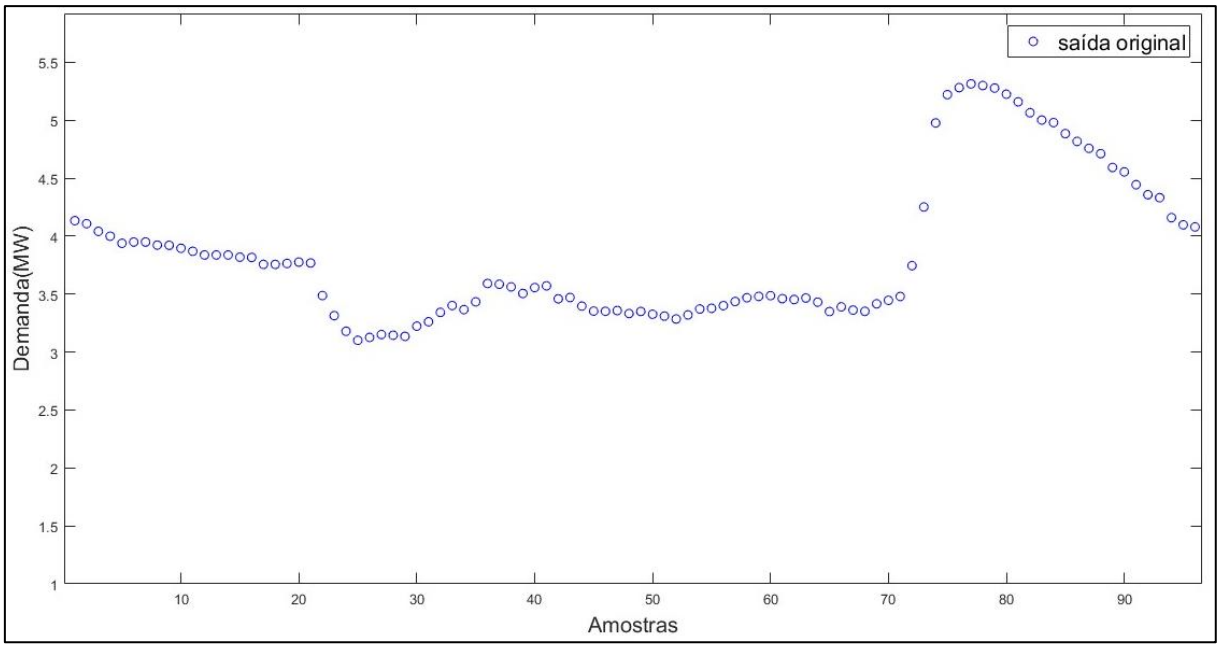
É importante dizer que tanto o algoritmo de detecção e correção de outliers, quanto os algoritmos complementares de pré-processamento e de inserção de outliers foram desenvolvidos com auxílio do software MATLAB.

4 RESULTADOS E DISCUSSÕES

4 RESULTADOS E DISCUSSÕES

Os dados usados para avaliação da metodologia constituem dados de um cenário real, sendo estes disponibilizados pela companhia de energia local. As medições, que são referentes a uma subestação de uma cidade localizada no estado da Paraíba, possuem intervalo de amostragem de 15 minutos, o que nos fornece 96 amostras diárias. O banco de dados usado nos testes possui, em sua totalidade, 199080 amostras que compreende o período de janeiro de 2008 a setembro de 2013, perfazendo o período de pouco mais que 5 anos e 8 meses. Devido ao grande número de amostras, todos os gráficos construídos aqui serão trechos da curva de demanda separada para testes.

FIGURA 14: GRÁFICO DE PARTE DA CURVA DE DEMANDA COM 96 AMOSTRAS REFERENTE DE 1º DIA DE JANEIRO DE 2008.



Fonte: Elaborada pelo autor.

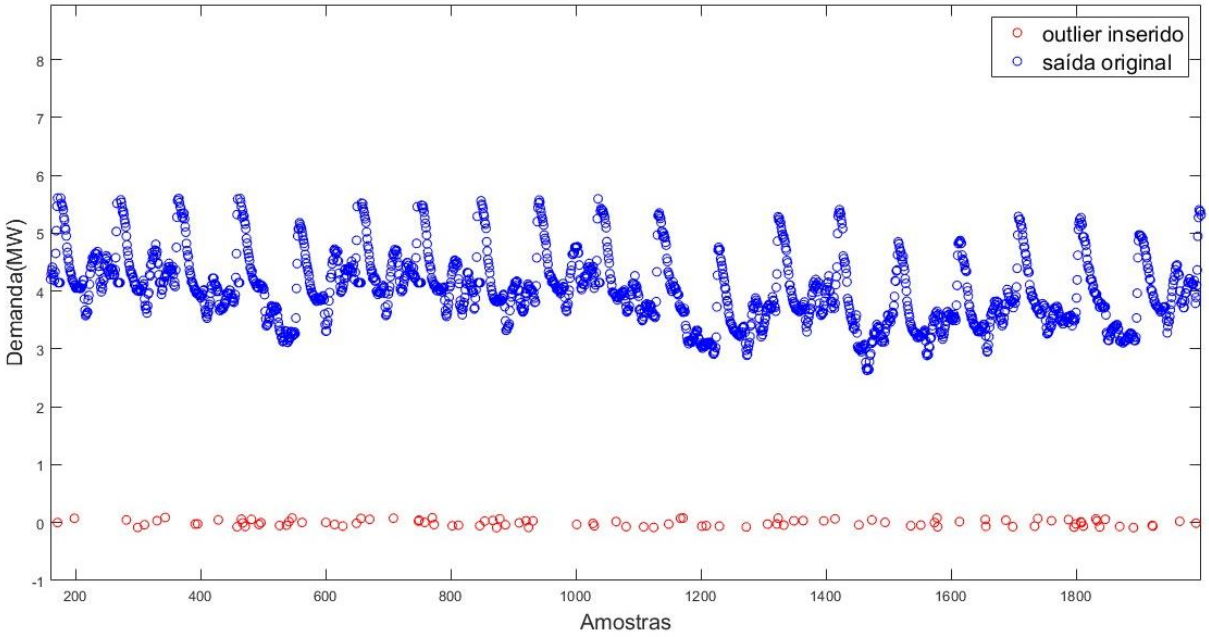
Na Figura 14 é possível ver o gráfico da curva de demanda com uma janela de 96 amostras referentes ao primeiro dia de janeiro de 2008.

O valor máximo de potência registrado no banco de dados em questão foi de P_{max} = 8,9503 MW, que corresponde a amostra de número 184764, aferição que foi realizada as 14h:45min do dia 8 de abril do ano de 2013. Por outro lado, o menor valor

registrado foi de Pmin = 1,4750 MW, que corresponde a amostra de número 4362, aferição que foi realizada as 10h:15min do dia 15 de fevereiro do ano de 2008.

Para avaliação da efetividade da metodologia proposta, foram inseridos outliers em posições aleatórias do vetor de dados de entrada, conforme os cenários de A até Q descritos no tópico anterior, em que o número de outliers varia de 2% a 10% do número total de amostras e a amplitude dos outliers variam, em módulo, de 0% a 200% do valor máximo de potência presente no banco de dados.

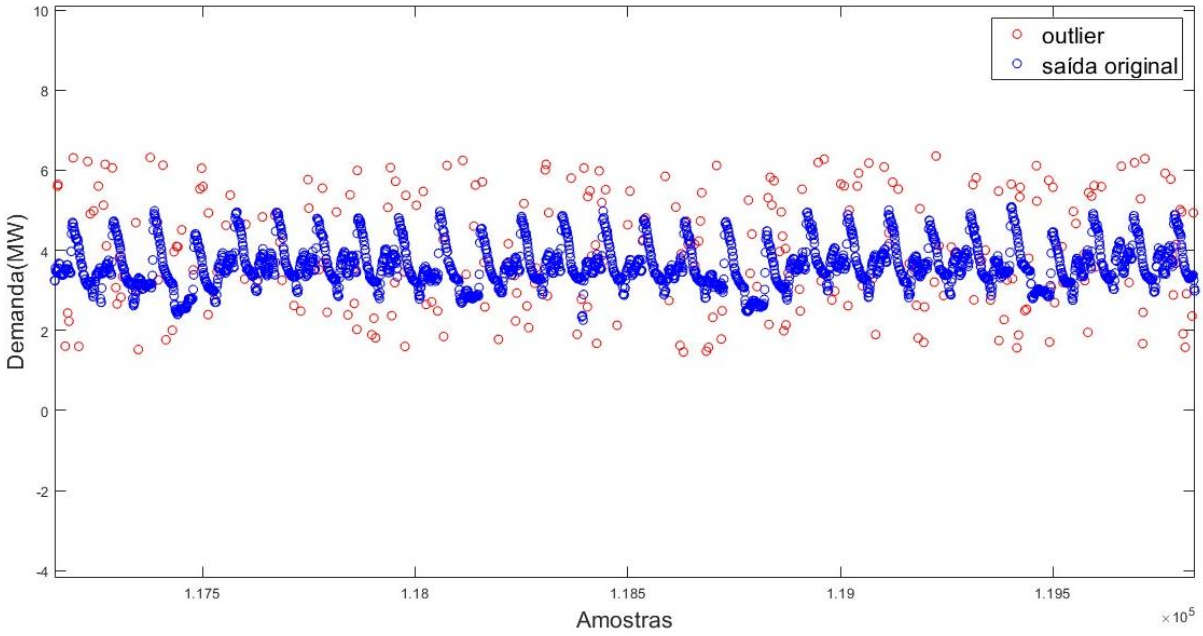
FIGURA 15: MELHOR CENÁRIO DE INSERÇÃO DE OUTLIERS NO BANCO DE DADOS.



Fonte: Elaborada pelo autor.

A inserção de outliers está exemplificada nas Figuras 15 e 16, na qual os pontos discrepantes inseridos estão destacados em vermelho. A Figura 15 representa o melhor cenário, onde a amplitude dos outliers inseridos estão contidas no intervalo de 0 a 1% de Pmax, em módulo. Já a Figura 16 representa o pior cenário onde os outliers inseridos possuem amplitudes dentro do intervalo que varia 3 vezes o valor do desvio padrão em torno da média dos dados de treinamento, e o número de outliers equivale a 10% de L, o máximo número de outliers inseridos em nossos testes.

FIGURA 16: PIOR CENÁRIO DE INSERÇÃO DE OUTLIERS NO BANCO DE DADOS.



Fonte: Elaborada pelo autor.

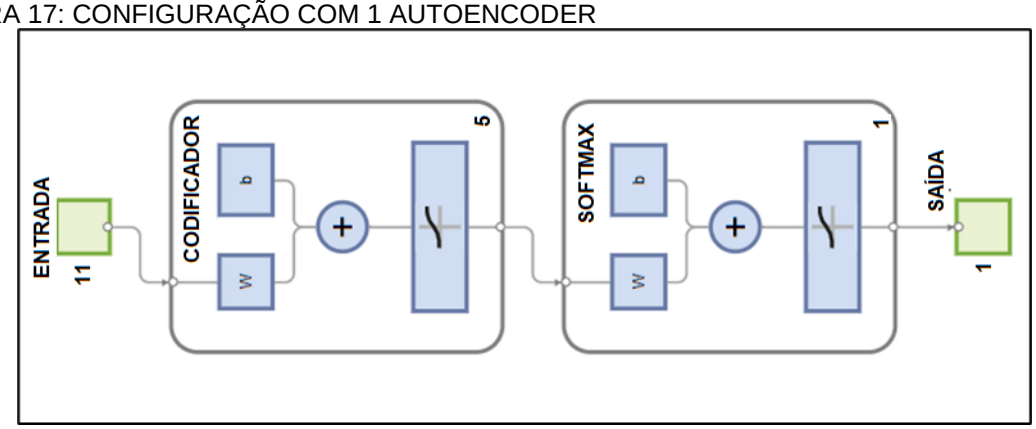
Do total de amostras do banco de dados, 80% foram utilizadas para treinamento das redes neurais presentes no sistema proposto, enquanto que os 20% restantes foram utilizados como dados inéditos para avaliação do sistema. Esse percentual também se reflete no número de outliers inseridos, isto é, se um número de outliers **NO** foi inserido em todo banco de dados, 80% foi disponibilizado pra treinamento e 20% para teste, quando necessário.

4.1 SELEÇÃO DE PARÂMETROS PARA O ALGORITMO DE DETECÇÃO

Como pode-se observar na Figura 11, a rede neural completa que compõe a metodologia proposta para detecção de outliers é formada por um autoencoder, que pode conter uma ou mais camadas ocultas, e uma camada softmax. Faz-se necessário, então, determinar o número de autoencoders empilhados, conforme esquema de treinamento descrito na Figura 13, que melhor atende ao nosso propósito.

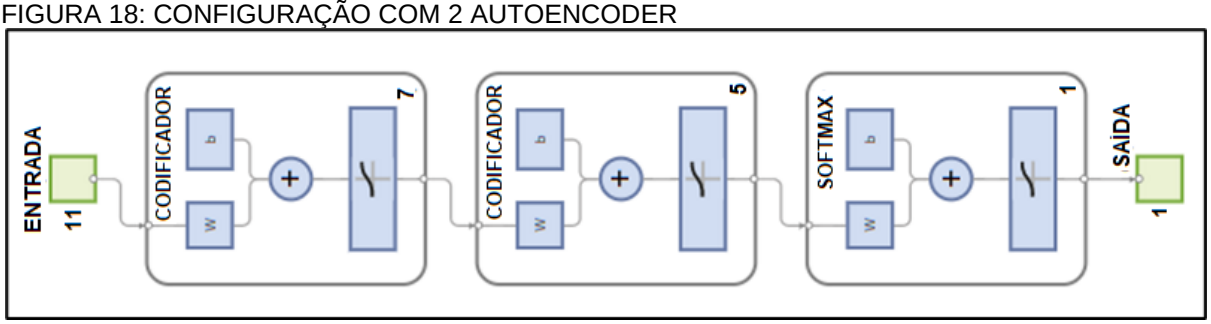
Nas Figuras de 17 a 19, é possível ver as configurações nas quais foram utilizados 1, 2 e 3 autoencoders na rede neural que compõe o subsistema de detecção de outliers.

FIGURA 17: CONFIGURAÇÃO COM 1 AUTOENCODER



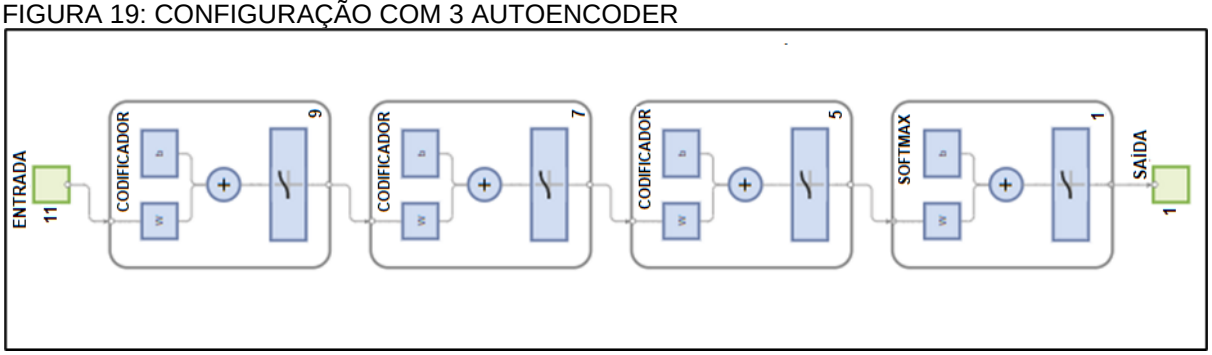
Fonte: Elaborada pelo autor.

FIGURA 18: CONFIGURAÇÃO COM 2 AUTOENCODER



Fonte: Elaborada pelo autor.

FIGURA 19: CONFIGURAÇÃO COM 3 AUTOENCODER



Fonte: Elaborada pelo autor.

Os resultados estão resumidos na Tabela 3. Para os testes, foram consideradas uma porcentagem de corte de 50%, o número de outliers foi de, aproximadamente, 10% do total de amostras.

Para medir o desempenho do sistema, foram utilizados os conceitos de **Acurácia, Precisão, Recall e F-score** relacionados à Matriz de Confusão, discutidos

no tópico 2.6. Tanto na Tabela 3, quanto nas outras apresentadas daqui em diante, as posições da Tabela que estão preenchidas com '*' são casos onde não foi possível obter resultado devido á existência de divisão por zero no cálculo do respectivo parâmetro.

TABELA 3: SELEÇÃO DO NÚMERO DE AUTOENCODERS PRESENTES NA REDE NEURAL QUE COMPÕE O SUBSISTEMA DE DETECÇÃO DE OUTLIERS.

Nº de autoencoders empilhados	**Métricas de Avaliação**			
	Acurácia	**Precisão**	**Recall**	**F-score**
1	99,43%	94,00%	94,75%	94,37%
2	99,48%	94,23%	95,45%	94,83%
3	94,98%	*	0,00%	*

Fonte: Elaborada pelo autor.

É interessante notar que para a configuração com 3 autoencoders, ocorreu overfitting, quando a rede neural consegue lidar muito bem com os dados de treinamento, porém, não consegue generalizar diante de dados inéditos. Neste caso, o sobreajuste ocorreu, muito provavelmente, devido ao fato de a rede ser complexa demais (ter muitas camadas ocultas) para a tarefa proposta e, portanto, deu-se o teste por encerrado, selecionando-se a configuração com 2 autoencoders, com **Acurácia** = 99,48% e **F-score** = 94,83, que obteve melhor desempenho.

O próximo passo, então, foi variar o número de neurônios das camadas ocultas, isto é, variou-se o número de neurônios da 1ª camada oculta **NNC1**, e da 2ª camada oculta **NNC2**, para um dado número de entradas **q** do autoencoder. Em seguida, modificou-se **q** e variou-se, novamente, o número de neurônio das camadas ocultas. O intuito deste procedimento é selecionar a melhor configuração, tendo já fixado o número de camadas internas do autoencoder visto na Figura 18, avaliando, neste momento, o número de entradas assim como o número de neurônios das camadas internas. Os resultados para todos os experimentos realizados, estão disponíveis no Apêndice A. O melhor resultado está disposto na Tabela 4.

TABELA 4: SELEÇÃO DO NÚMERO DE ENTRADAS E DO NÚMERO DE NEURÔNIO DAS CAMADAS OCULTAS PARA O ALGORÍTMO DE DETECÇÃO: MELHOR RESULTADO.

Configuração do Autoencoder			Métricas de Avaliação			
q	NNC1	NNC2	Acurácia	Precisão	Recall	F-score
11	10	9	99,74%	98,47%	96,40%	97,42%

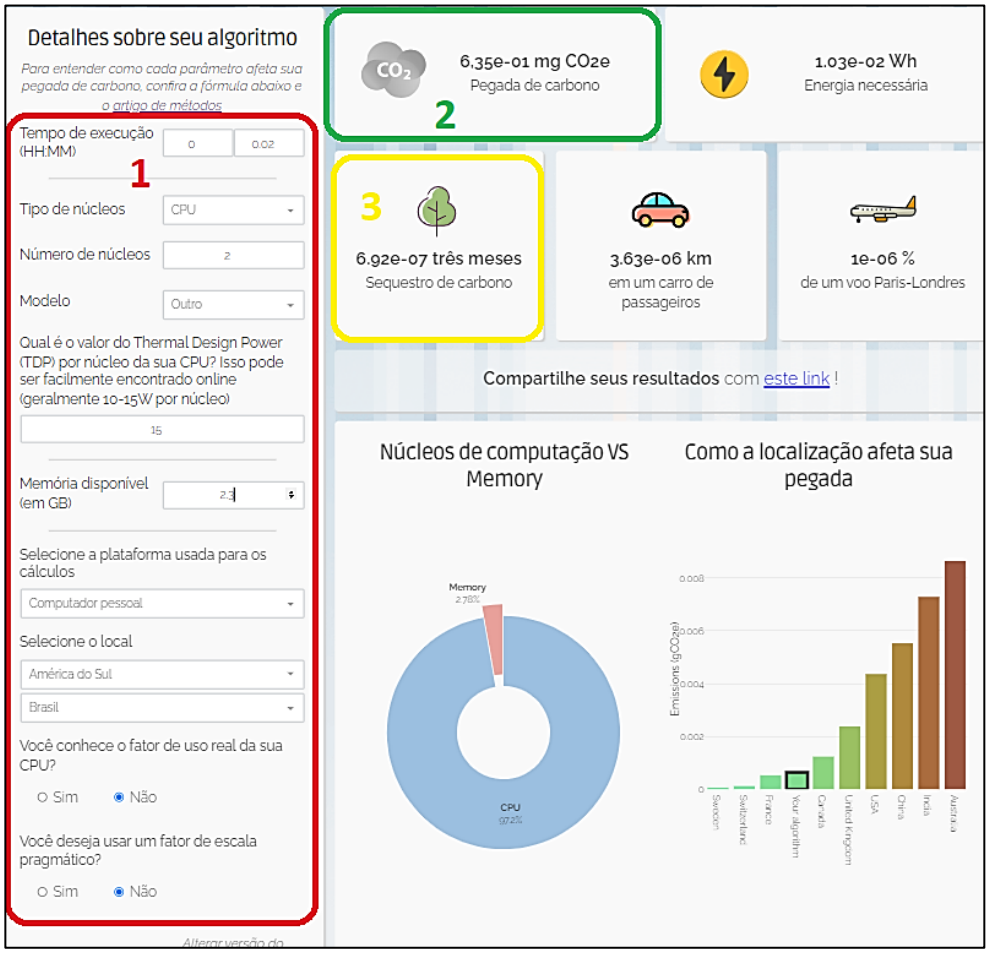
Fonte: Elaborada pelo autor.

Analisando os resultados, observa-se que 2 configurações tiveram desempenho muito superiores à todas as outras testadas: a configuração com 11 entradas, 10 neurônios na primeira camada oculta e 9 neurônios na segunda camada oculta (denominada de configuração **A**), e a configuração com 11 entradas, 10 neurônios na primeira camada oculta e 7 neurônios na segunda camada oculta (denominada de configuração **B**). Embora ambas tenham obtido desempenhos muito próximos, a configuração **A**, presente na Tabela 4, obteve melhor desempenho, tendo em vista uma maior Acurácia (96,40% ante os 96,05% da configuração **B**) e maior F-score (97,42% ante os 96,75% da configuração **B**).

Em seguida, conforme o roteiro pré-estabelecido, foi testada a influência do número de outliers presentes nos dados sobre a capacidade de detecção do algoritmo. Para tanto, foram inseridas quantidades de outliers equivalentes a 2%, 4%, 5%, 6%, 8% e 10% do número total de amostras **L**. O resultado dos testes está expresso na Tabela 5.

Nos experimentos conduzidos em cenários onde avalia-se a influência da variação tanto do número de outliers, quanto da amplitude dos outliers inseridos, foram calculados a pegada de carbono (CO_2e), em mg, e a quantidade de meses-arvores necessários para que o carbono gerado fosse sequestrado, conforme explicado no tópico 2.6. Para tanto, foi utilizada uma calculadora de carbono, desenvolvida por (LANNELONGUE et al., 2021).

FIGURA 20: CALCULADORA DE CARBONO.



Fonte: Obtida de ANNELONHUE et al. (2021).

Na Figura 20 é possível ver a interface da aplicação desenvolvida que está disponível online de forma gratuita. No retângulo (1), destacado em vermelho, é possível ver os parâmetros de entrada necessários para cálculo dos parâmetros. São eles:

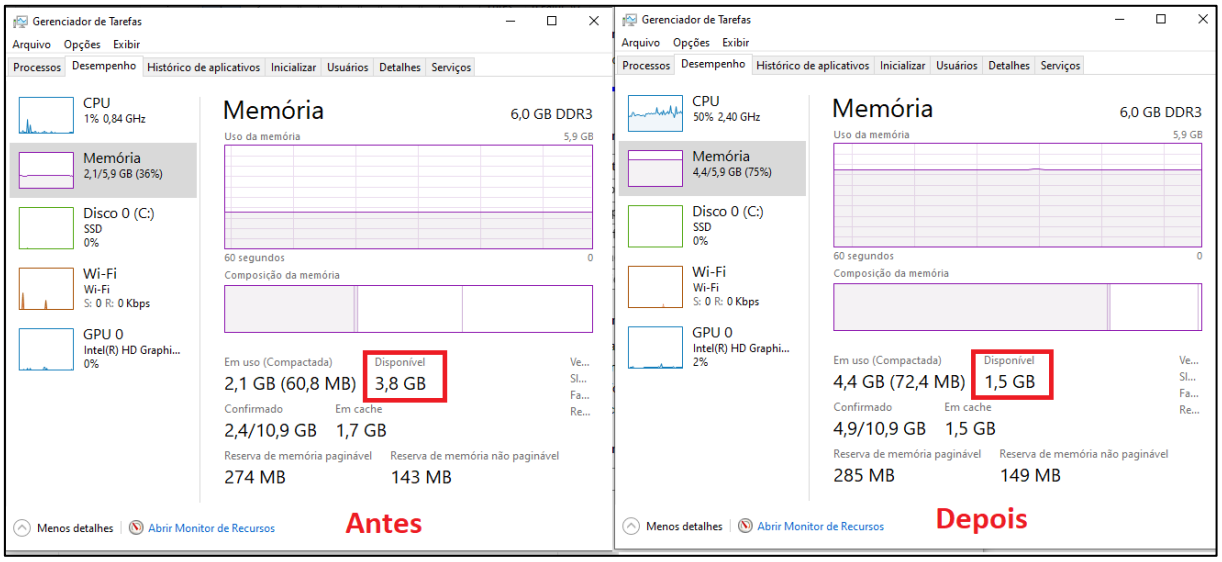
- Tempo de execução;
- Tipo de núcleos: “CPU” foi selecionado como resposta;
- Número de núcleos: “2” foi a resposta (INTEL, 2023);
- Modelo: Há alguns modelos de CPU disponíveis, dentre os quais não consta o modelo processador presente no computador utilizado nos experimentos. A opção “Outros” foi selecionada como resposta;
- Memória disponível, em GB: “2,3” foi a resposta. Para estimar a quantidade de memória RAM solicitada durante a execução dos algoritmos, primeiramente, abrimos o gerenciador de tarefas do

Windows 10, sistema operacional instalado na máquina utilizada para os experimentos, e verificamos a quantidade de memória RAM disponível, que era de 3,8GB. Depois, abrimos o software MATLAB e executamos os algoritmos e verificamos que a memória se mantinha em torno de 1,5GB. Então, a quantidade de memória RAM que efetivamente foi solicitada durante a execução dos algoritmos foi de cerca de 2,3GB. A Figura 21 ilustra este procedimento.

- Qual o TDP (Thermal Design Power) da CPU, em W: “15” foi a resposta (INTEL, 2023);
- Plataforma usada para os cálculos (“Computador pessoal” foi selecionado como resposta);
- Você conhece o uso real da sua CPU? (“Não” foi selecionado como resposta)
- Você deseja usar um fator de escala pragmático? (“Não” foi selecionado como reposta).

Ainda sobre a Figura 20, é possível ver no retângulo (2), destacado em verde e no retângulo (3), destacado em amarelo, a pegada de carbono e a quantidade de meses-árvores calculada pela aplicação, considerando os parâmetros fornecidos.

FIGURA 21: ESTIMANDO A QUANTIDADE DE RAM SOLICITADA PARA EXECUÇÃO DOS ALGORITMOS.



Fonte: Obtida do Windows 10 MICROSOFT, (2023).

TABELA 5: AVALIANDO A INFLUÊNCIA DO NÚMERO DE OUTLIERS INSERIDOS SOBRE O FUNCIONAMENTO DO ALGORITMO DE DETECÇÃO PROPOSTO.

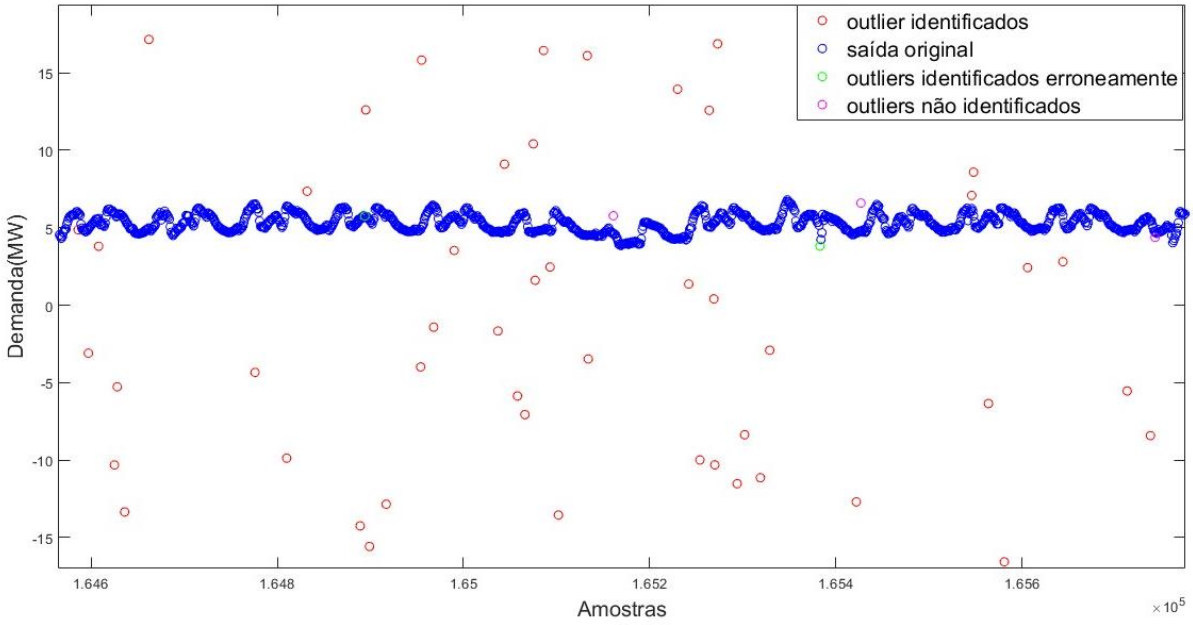
NO	Métricas de Avaliação					
	Acurácia	Precisão	Recall	F-score	CO_2e(mg)	Mês-árvore
2% de L	99,23%	99,20%	62,38%	76,59%	$6,98x10^{-2}$	$7,62x10^{-8}$
4% de L	98,52%	98,75%	63,94%	77,62%	$6,66x10^{-2}$	$7,27x10^{-8}$
5% de L	99,74%	98,12%	96,75%	97,43%	$6,66x10^{-2}$	$7,27x10^{-8}$
6% de L	99,65%	98,09%	96,04%	97,05%	$7,62x10^{-2}$	$8,31x10^{-8}$
8% de L	97,82%	84,97%	88,50%	86,70%	$8,88x10^{-2}$	$9,69x10^{-8}$
10% de L	96,10%	98,96%	61,85%	76,12%	$7,93x10^{-2}$	$8,65x10^{-8}$

Fonte: Elaborada pelo autor.

Analisando a Tabela 5, nota-se que incialmente, para poucos outliers inseridos, o algoritmo tem desempenho baixo, com F-score de 76,59% dos outliers inseridos na fase de testes. Em seguida, o algoritmo atinge a sua melhor performance, com F-score de 97,43% dos outliers inseridos na fase de teste, tendo sido treinado com a inserção de uma quantidade de outliers equivalente a 5% do número total de amostras **L**.

Observa-se também, na Tabela 5, que a pegada de carbono, assim como a quantidade de meses-árvore obtiveram valores mínimos de 0,0666mg e $7,27x10^{-8}$ e máximos de 0,0888mg e $9,69x10^{-8}$, respectivamente.

FIGURA 22: CENÁRIO COM NÚMERO DE OUTLIERS EQUIVALENTE A 5% DE L.

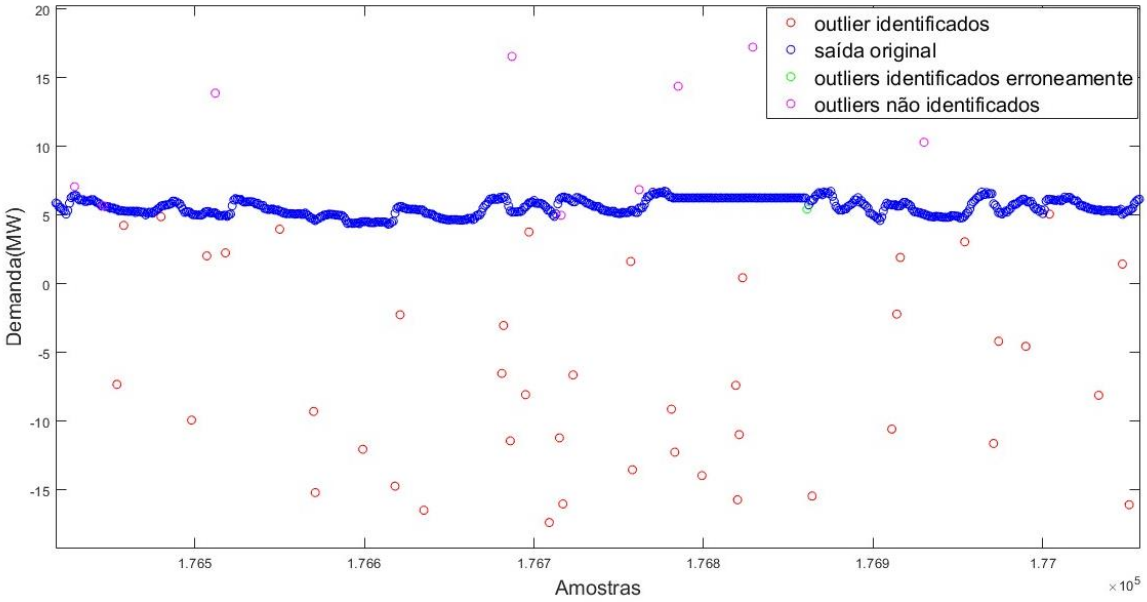


Fonte: Elaborada pelo autor.

O algoritmo apresenta uma tendência de decréscimo no F-score outliers à medida que o número de outliers inseridos na fase de treinamento cresce. Este resultado sugere que pode haver um número ótimo, ou uma faixa de valores ótima, de outliers a serem inseridos de forma que o algoritmo identifique o maior número de outliers possível, mantendo uma taxa de erros baixa.

As Figuras 22 e 23 representam trechos das curvas de demanda após aplicação do algoritmo de detecção para dois cenários presentes na Tabela 4: com NO = 5 % de L e NO = 2% de L, respectivamente. Observa-se que no segundo caso, cenário no qual o algoritmo obteve menor rendimento, grande parte dos outliers inseridos acima da curva de demanda não foram identificados pelo algoritmo.

FIGURA 23: CENÁRIO COM NÚMERO DE OUTLIERS EQUIVALLENTE A 2% DE L.



Fonte: Elaborada pelo autor.

Por conseguinte, analisou-se a influência dos intervalos numéricos onde os outliers estão inseridos sob a precisão do subsistema de detecção de outliers. Os resultados podem ser vistos na Tabela 6.

TABELA 6: AVALIANDO A INFLUÊNCIA DA AMPLITUDE DOS OUTLIERS INSERIDOS SOBRE O FUNCIONAMENTO DO ALGORITMO DE DETCÇÃO PROPOSTO.

Faixa de variação da amplitude dos outliers (em módulo)	**Acurácia**	**Precisão**	**Recall**	**F-score**	**CO_2e(mg)**	**Mês-árvore**
de 0 a 1% de Pmax	100,00%	100,00%	100,00%	100,00%	$1{,}01\times10^{-1}$	$1{,}1\times10^{-7}$
de 0 a 25% de Pmax	100,00%	100,00%	99,95%	99,97%	$6{,}66\times10^{-2}$	$7{,}27\times10^{-8}$
de 0 a 50% de Pmax	99,82%	98,69%	97,80%	98,24%	$7{,}30\times10^{-2}$	$7{,}96\times10^{-8}$
de 0 a 75% de Pmax	99,16%	98,15%	84,80%	90,99%	$7{,}30\times10^{-2}$	$7{,}96\times10^{-8}$
de 0 a 100% de Pmax	99,56%	97,75%	93,40%	95,53%	$6{,}98\times10^{-2}$	$7{,}62\times10^{-8}$
de 0 a 200% de Pmax	99,74%	98,66%	96,05%	97,34%	$6{,}66\times10^{-2}$	$7{,}27\times10^{-8}$
u ± 3dp	95,92%	86,26%	70,60%	77,65%	$6{,}66\times10^{-2}$	$7{,}27\times10^{-8}$

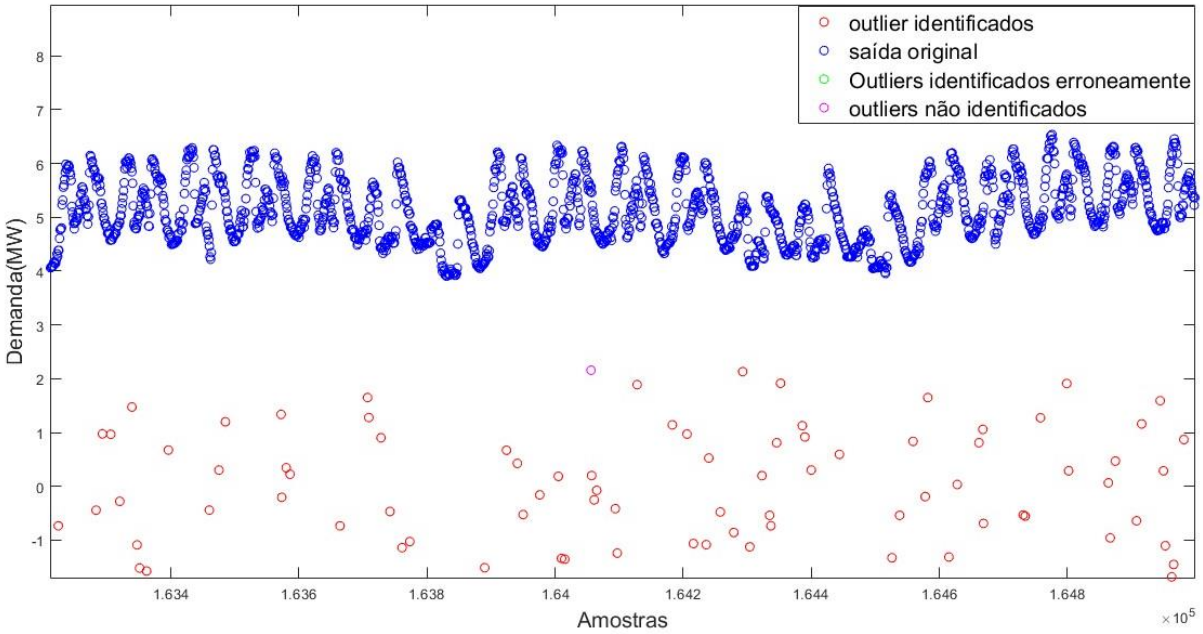
Fonte: Elaborada pelo autor.

Através da Tabela 6, é possível observar que os melhores resultados são obtidos para a faixa de variação de 0 a 1%, e de 0 a 25% de **Pmax**, alcançando-se

um F-score de 100% e de 99,97%, respectivamente. A performance mais baixa foi obtida para a faixa de variação u ± 3dp, para a qual, o algoritmo apresentou um F-score de 77,65%.

Do ponto de vista do impacto ambiental, considerando as Tabelas 5 e 6, observamos que em média, a pegada de carbono do algoritmo foi de 0,0741mg de CO_2e e que seriam necessários $8{,}09 \times 10^{-8}$ meses-árvores para sequestro deste carbono.

FIGURA 24: CENÁRIO PARA O QUAL AS AMPLITUDES DOS OUTLIERS ESTÃO CONTIDAS NO INTERVALO QUE VARIA DE 0 A 25% DE PMAX.



Fonte: Elaborada pelo autor.

Para visualização do cenário descrito na Tabela 6 para o qual todos os outliers foram identificados corretamente, basta recorrer à Figura 15. Já o cenário da Tabela 6, para o qual obteve-se um F-score de 99,97%, está exemplificado na Figura 24.

4.2 COMPARAÇÃO 1: AUTOENCODERS X TRÊS ALGORITMOS DE DETECÇÃO TRADICIONAIS.

Nesta subseção, foi realizada a comparação entre o algoritmo de identificação de outliers baseado em autoencoders, desenvolvido neste trabalho, e 3 algoritmos usados tradicionalmente para a detecção de outliers. O primeiro algoritmo

basicamente, calcula a média e o desvio padrão do banco de dados usado para treinamento. Em seguida, o algoritmo calcula a diferença entre cada ponto dos dados separados para teste e a média já calculada e os compara com o triplo, ou o quádruplo, do desvio padrão. Caso esta diferença seja maior que 3, (ou 4) vezes o desvio padrão, o ponto correspondente é considerado um outlier (MEHROTRA; MOHAN; HUANG, 2017).

Os outros dois algoritmos utilizados para comparação estão disponíveis na biblioteca de detecção de outliers do MATLAB: o primeiro método, denominado "*median*" (mediana), é semelhante ao algoritmo baseado em desvio padrão, porém, este compara a distância entre cada ponto e a mediana com o Desvio Absoluto Médio (MAD - Mean Absolute Deviation) escalonado, para decidir se determinado ponto deve ser considerado outlier; O segundo, denominado *"isolation forest"* (floresta de isolamento) é um famoso algoritmo para detecção de anomalias onde os dados são particionados recursivamente com cortes paralelos de eixo em pontos escolhidos aleatoriamente e em atributos selecionados também de forma aleatória, de modo a isolar as instâncias em nós com menos e menos instâncias a cada iteração, até que os pontos sejam isolados em nós uma instância. Como outliers, em geral, estão localizados em regiões esparsas, os "galhos de árvores" contendo anomalias são menos profundos (MATHWORKS, 2022; DEVORE, 2014).

TABELA 7: ANALISANDO A INFLUÊNCIA DO NÚMERO DE OUTLIERS INSERIDOS SOBRE O FUNCIONAMENTO DO ALGORITMO BASEADO NO DESVIO PADRÃO.

NO	Métricas - Ponto de Corte de 3 desvios padrão.				Métricas - Ponto de Corte de 4 desvios padrão.					
	Acurácia	Precisão	Recall	F-score	Acurácia	Precisão	Recall	F-score	CO_2e(mg)	mês-árvore
2%	98,05%	52,42%	29,75%	37,96%	98,54%	99,10%	27,63%	43,21%	$5{,}08\times10^{-2}$	$5{,}54\times10^{-8}$
4%	95,98%	50,00%	30,56%	37,94%	97,07%	99,54%	27,19%	42,71%	$9{,}84\times10^{-1}$	$1{,}07\times10^{-6}$
5%	95,98%	74,24%	30,55%	43,29%	96,38%	99,64%	28,00%	43,72%	$5{,}39\times10^{-1}$	$5{,}88\times10^{-6}$
6%	95,23%	76,06%	30,58%	43,63%	95,74%	99,72%	29,42%	45,43%	$7{,}82\times10^{-1}$	$8{,}31\times10^{-7}$
8%	93,85%	82,78%	29,59%	43,60%	94,17%	99,77%	27,50%	43,12%	$8{,}25\times10^{-2}$	$9{,}00\times10^{-8}$
10%	92,52%	85,51%	30,83%	45,31%	92,71%	99,82%	27,55%	43,18%	$6{,}35\times10^{-2}$	$6{,}92\times10^{-8}$

Fonte: Elaborada pelo autor.

TABELA 8: ANALISANDO A INFLUÊNCIA DO NÚMERO DE OUTLIERS INSERIDOS SOBRE O FUNCIONAMENTO DO ALGORITMO BASEADO NO MAD.

Nº de outliers em relação a L	Métricas de avaliação - MAD					
	Acurácia	Precisão	Recall	f-score	CO_2e(mg)	mês-árvore
2%	99,46%	89,81%	82,63%	86,07%	$1,62x10^{-1}$	$1,77x10^{-7}$
4%	99,11%	95,87%	81,25%	87,96%	$3,81x10^{-2}$	$4,15x10^{-8}$
5%	98,99%	97,16%	82,20%	89,06%	$4,13x10^{-2}$	$4,50x10^{-8}$
6%	98,73%	98,17%	80,42%	88,41%	$3,49x10^{-2}$	$3,81x10^{-8}$
8%	98,58%	99,40%	82,78%	90,33%	$4,13x10^{-2}$	$4,50x10^{-8}$
10%	98,00%	99,88%	80,18%	88,95%	$3,49x10^{-2}$	$3,81x10^{-8}$

TABELA 9: ANALISANDO A INFLUÊNCIA DO NÚMERO DE OUTLIERS INSERIDOS SOBRE O FUNCIONAMENTO DO ALGORITMO I-FOREST.

Nº de outliers em relação a L	Métricas de avaliação - iforest					
	Acurácia	Precisão	Recall	f-score	CO_2e(mg)	mês-árvore
2%	99,53%	89,04%	87,38%	88,20%	1,63	$1,78x10^{-6}$
4%	99,14%	89,60%	88,88%	89,24%	1,20	$1,31x10^{-6}$
5%	98,90%	89,24%	88,75%	88,99%	1,14	$1,25x10^{-6}$
6%	98,72%	89,71%	89,04%	89,38%	1,17	$1,28x10^{-6}$
8%	98,46%	90,64%	90,19%	90,41%	1,18	$1,28x10^{-6}$
10%	98,14%	91,18%	90,20%	90,69%	1,12	$1,22x10^{-6}$

Nas Tabelas 7, 8 e 9, é possível ver os resultados dos testes realizados onde se avalia a influência da variação do número de outliers sobre o funcionamento dos algoritmos baseado em desvio padrão, MAD e o de floresta de isolamento, respectivamente. Note que para o algoritmo baseado em desvio padrão, os testes foram realizados considerando o limiar de 3 e 4 vezes o desvio padrão e que, para este último, o algoritmo de detecção de outliers obteve melhor desempenho considerando todos os parâmetros avaliativos, exceto para o Recall.

Observando as Tabelas 7,8 e 9, verifica-se que o desempenho dos algoritmos baseado em MAD e o iforest apresentam desempenho consideravelmente superior, em todos os cenários, quando comparados com o algoritmo baseado no desvio padrão.

De uma forma geral, entre estes dois últimos algoritmos, apesar de apresentarem resultados próximos, o i-forest apresenta desempenho superior, tendo em vista que apresenta f-score superior em todos os cenários, exceto no cenário onde NO = 5% de L. O i-forest apresenta Acurácia superior em 3 cenários (NO = 2%, 4% e 10% de L). É interessante notar que o algoritmo baseado no MAD apresentou melhor

precisão em todos os cenários, enquanto o i-forest apresentou melhor recall em todos os cenários.

Comparando a Tabela 5 com as Tabelas 8 e 9, observa-se que o algoritmo baseado em autoencoders obtém melhor desempenho nos cenários para os quais o número de outliers inseridos é de 5% e 6 % do tota de amostras. A Tabela 10 resume o melhor desempenho obtido por cada algoritmo de detecção considerando os cenários para os quais avalia-se a influência do número de outliers.

TABELA 10: COMPARAÇÃO ENTRE OS MELHORES CENÁRIOS: AUTOENCODERS X 3 ALGORITMOS DE DETECÇÃO TRADICIONAIS.

Algoritmo	**Nº de outliers em relação a L**	**Acurácia**	**Precisão**	**Recall**	**F-score**	**CO_2e(mg)**	**mês-árvore**
Baseado em Autoencoders	5%	99,74%	98,12%	96,75%	97,43%	$6{,}66\times10^{-2}$	$7{,}27\times10^{-8}$
Baseado no desvio padrão (4x)	6%	95,74%	99,72%	29,42%	45,43%	$7{,}82\times10^{-1}$	$8{,}31\times10^{-7}$
MAD	8%	98,58%	99,40%	82,78%	90,33%	$4{,}13\times10^{-2}$	$4{,}50\times10^{-8}$
I-forest	10%	98,14%	91,18%	90,20%	90,69%	1,12	$1{,}22\times10^{-6}$

Fonte: Elaborada pelo autor.

Em seguida, os algoritmos de detecção baseados em desvio padrão, MAD e o i-forest foram avaliados do ponto de vista da variação da amplitude dos outliers e os resultados podem ser vistos nas Tabelas 11, 12 e 13, respectivamente.

TABELA 11: ANALISANDO A INFLUÊNCIA DA AMPLITUDE DOS OUTILIERS INSERIDOS SOBRE O FUNCIONAMENTO DO ALGORITMO BASEADO NO DESVIO PADRÃO.

Faixa de variação da amplitude dos outliers (em módulo)	**Métricas - Ponto de Corte de 3 desvios padrão.**				**Métricas - Ponto de Corte de 4 desvios padrão.**					
	Acurácia	**Precisão**	**Recall**	**F-score**	**Acurácia**	**Precisão**	**Recall**	**F-score**	**CO_2e(mg)**	**mês-árvore**
de 0 a 1% de Pmax	93,43%	0,00%	0,00%	*	93,97%	0,00%	0,00%	*	$7{,}62\times10^{-2}$	$8{,}31\times10^{-8}$
de 0 a 25% de Pmax	93,44%	0,00%	0,00%	*	93,97%	0,00%	0,00%	*	$8{,}25\times10^{-2}$	$9{,}00\times10^{-8}$
de 0 a 50% de Pmax	93,43%	0,00%	0,00%	*	93,97%	0,00%	0,00%	*	$9{,}84\times10^{-2}$	$1{,}07\times10^{-8}$
de 0 a 75% de Pmax	93,46%	0,00%	0,00%	*	93,97%	0,00%	0,00%	*	$6{,}66\times10^{-2}$	$7{,}27\times10^{-8}$
de 0 a 100% de Pmax	94,17%	58,13%	11,92%	19,78%	94,38%	98,81%	6,92%	12,93%	$7{,}30\times10^{-2}$	$7{,}96\times10^{-8}$
de 0 a 200% de Pmax	95,30%	77,80%	30,96%	44,29%	95,72%	99,86%	29,13%	45,10%	$7{,}62\times10^{-2}$	$8{,}31\times10^{-8}$
u ± 3dp	89,44%	0,00%	0,00%	*	89,95%	0,00%	0,00%	*	$9{,}20\times10^{-2}$	$1{,}00\times10^{-8}$

Fonte: Elaborada pelo autor.

TABELA 12: ANALISANDO A INFLUÊNCIA DA AMPLITUDE DOS OUTILIERS INSERIDOS SOBRE O FUNCIONAMENTO DO ALGORITMO BASEADO NO MAD.

Faixa de variação da amplitude dos outliers (em módulo)	Métricas de avaliação - MAD					
	Acurácia	Precisão	Recall	f-score	CO_2e(mg)	mês-árvore
de 0 a 1% de Pmax	99,85%	97,04%	100,00%	98,50%	$3,49x10^{-2}$	$3,81x10^{-8}$
de 0 a 25% de Pmax	98,25%	95,59%	68,35%	79,71%	$3,81x10^{-2}$	$4,15x10^{-8}$
de 0 a 50% de Pmax	97,76%	93,96%	59,15%	72,60%	$4,13x10^{-2}$	$4,50x10^{-8}$
de 0 a 75% de Pmax	97,62%	94,29%	56,10%	70,34%	$4,76x10^{-2}$	$5.19x10^{-8}$
de 0 a 100% de Pmax	98,06%	95,55%	64,45%	76,98%	$5,39x10^{-2}$	$5,88x10^{-8}$
de 0 a 200% de Pmax	98,96%	97,04%	81,85%	88,80%	$4,44x10^{-2}$	$4,85x10^{-8}$
u ± 3dp	94,84%	0,00%	0,00%	*	$4,44x10^{-2}$	$4,85x10^{-8}$

Fonte: Elaborada pelo autor.

TABELA 13: ANALISANDO A INFLUÊNCIA DA AMPLITUDE DOS OUTILIERS INSERIDOS SOBRE O FUNCIONAMENTO DO ALGORITMO I-FOREST.

Faixa de variação da amplitude dos outliers (em módulo)	Métricas - iforest					
	Acurácia	Precisão	Recall	f-score	CO_2e(mg)	mês-árvore
de 0 a 1% de Pmax	96,80%	68,46%	67,40%	67,93%	1,10	$1,20x10^{-6}$
de 0 a 25% de Pmax	98,07%	81,39%	79,80%	80,59%	1,09	$1,19x10^{-6}$
de 0 a 50% de Pmax	98,16%	81,86%	81,45%	81,65%	1,31	$1,43x10^{-6}$
de 0 a 75% de Pmax	97,41%	74,28%	73,95%	74,12%	1,52	$1,65x10^{-6}$
de 0 a 100% de Pmax	98,03%	80,65%	80,00%	80,32%	1,35	$1,47x10^{-6}$
de 0 a 200% de Pmax	98,94%	89,80%	88,95%	89,37%	1,19	$1,29x10^{-6}$
u ± 3dp	94,80%	49,10%	95,55%	64,87%	1,15	$1,25x10^{-6}$

Fonte: Elaborada pelo autor.

Novamente, nota-se que o desempenho do i-forest e do algoritmo baseado no MAD têm desempenho significativamente superior ao algoritmo baseado em desvio padrão.

O algoritmo baseado em MAD tem, em todos os cenários, melhor Acurácia (exceto no cenário no qual as amplitudes dos outliers variam de 0 a 50% de Pmax) e maior precisão (exceto para o cenário onde a faixa de variação da amplitude dos outliers é de u ± 3dp). Em contrapartida, o i-forest obteve, em todos os cenários, maiores recall e f-score (exceto no cenário no qual as amplitudes dos outliers variam de 0 a 1% de Pmax).

Outro fato interessante é que, para a faixa de variação que vai de 0 a 1% de Pmax, onde os outliers estão muito próximos de outliers tipo zero, o algoritmo baseado no MAD obtém seu melhor desempenho, o que já esperado, enquanto o i-forest obtém seu segundo menor f-score (67,93%). Em contrapartida, no cenário onde faixa de variação é de u ± 3dp, o algoritmo baseado no MAD tem precisão e recall nulos, o que significa que o algoritmo não produziu nenhum verdadeiro positivo.

Comparando as Tabelas 12 e 13 com a Tabela 6, é possível ver que o algoritmo desenvolvido neste trabalho obtém desempenho superior em todos os cenários. A Tabela 14 reúne os cenários com melhor desempenho para todos os algoritmos de detecção do ponto de vista da variação da amplitude dos outliers.

TABELA 14: COMPARAÇÃO ENTRE OS MELHORES CENÁRIOS: AUTOENCODERS X 3 ALGORITMOS DE DETECÇÃO TRADICIONAIS.

Algoritmo	**Faixa de variação da amplitude dos outliers (em módulo)**	**Acurácia**	**Precisão**	**Recall**	**F-score**	**CO_2e(mg)**	**mês-árvore**
Baseado em Autoencoders	De 0 a 1% de Pmax	100,00%	100,00%	100,00%	100,00%	0,101	$1,10x10^{-7}$
Baseado no desvio padrão (4x)	de 0 a 200% de Pmax	95,72%	99,86%	29,13%	45,10%	0,0762	$8,31x10^{-8}$
MAD	de 0 a 1% de Pmax	99,85%	97,04%	100,00%	98,50%	0,0349	$3,81x10^{-8}$
I-forest	de 0 a 200% de Pmax	98,94%	89,80%	88,95%	89,37%	1,19	$1,29x10^{-6}$

Fonte: Elaborada pelo autor.

Analisando a Tabela 14, nota-se que o algoritmo baseado em autoencoders desenvolvido neste trabalho conseguiu chegar a números de um cenário ideal, onde é possível identificar todos os outliers sem cometer erros, para um número de outliers correspondente a 5% de **L**, estando as amplitudes dos outliers inseridas no intervalo e 0 a 1% de **Pmax** (ou de 0 a 25% de **Pmax**, quando o algoritmo alcançou obteve desempenho semelhante).

Considerando as Tabelas 7 e 11, 8 e 12, e 9 e 13, foram calculadas as médias da pegada de carbono e da quantidade de meses-arvores para os 3 algoritmos de detecção de outliers considerados nesta seção, obtendo-se 0,2359mg de CO_2e e $6,48x10^{-7}$ meses-arvores para o algo baseado no desvio padrão, 0,0505mg de CO_2e e $5,54x10^{-8}$ meses-arvores para o algo baseado no MAD, e 1,2423mg de CO_2e e $1,34x10^{-8}$ meses-arvores para o i-forest, respectivamente.

4.3 SELEÇÃO DE PARÂMETROS PARA O ALGORITMO DE CORREÇÃO

De maneira análoga ao procedimento realizado para seleção de parâmetros da rede neural que compõe o subsistema de detecção de outliers, foram realizados os testes para selecionar o número de camadas, considerando 3 estratégias de passagem de dados entre os algoritmos de detecção e correção, assim como o número de entradas e o número de neurônio das camadas ocultas do nosso autoencoder do subsistema de correção de outliers.

Para avaliação dos resultados, foram utilizados a raiz quadrada do erro quadrático médio **RMSE**, o erro absoluto máximo **EAM**, o erro relativo máximo **ERM** e a média percentual absoluta do erro **MAPE**, parâmetros calculados sobre os pontos que foram substituídos pelo algoritmo de correção.

Como há 4 parâmetros distintos para analisar, fica difícil decidir, dentre as várias configurações testadas, qual obteve o melhor desempenho com um simples olhar sobre os dados.

Portanto, fez-se necessário o desenvolvimento de um sistema de pontuação capaz de eleger, dentre as várias configurações disponíveis, a que obteve o melhor desempenho.

Para isso, os seguintes princípios foram adotados como norte:

- Considerou-se que a métrica mais importante é a **RMSE**, sendo este o parâmetro mais sensível a erros maiores, tendo em vista o processo de quadratura que o produziu e que, adicionalmente, é um parâmetro que reflete uma característica de um conjunto de dados de interesse (nesse caso, dos outliers), visto que considera o erro para todos os pontos estimados, isto é, pontos que foram corrigidos.
- O segundo parâmetro com mais relevância é o **MAPE**, visto que também é uma medida que considera o mesmo conjunto de dados de interesse utilizado no cálculo do RMSE, embora seja menos sensível a erros maiores.
- Por último, os erros, absoluto máximo e relativo máximo devem ter menos relevância em nosso sistema de pontuação, visto que refletem uma característica pontual da curva.

Em suma, a RMSE deve ter maior relevância, os erros máximo absoluto e relativo devem ter um menor peso, e o MAPE deve ser responsável por uma pontuação intermediária.

Sendo assim, em ordem crescente, definiu-se que:

- caso determinada configuração obtenha menor valor de erro máximo absoluto ou de erro máximo relativo, receberá 2 pontos;
- caso uma configuração obtenha menor valor de **MAPE**, esta receberá 3 pontos;
- a recompensa pelo menor valor de **RMSE** será de 4 pontos. Portanto, o sistema de pontuação atribui notas que variam de 0 a 11 pontos, auxiliando a selecionar a melhor configuração.

TABELA 15: ESCOLHENDO O NÚMERO DE CAMADAS OCULTAS DO AUTOENCODER DO SUBSISTEMA DE CORREÇÃO DE OUTLIERS.

Método de Transferência de outliers	Nº de Autoencoders empilhados	Métricas de Avaliação								
		RMSE	EAM	ERM	MAPE	P1	P2	P3	P4	total
Substituindo por 0	1	0,8908	4,0517	53,73%	13,76%	0	0	0	0	0
	2	0,7069	3,8863	50,41%	8,92%	0	0	0	0	0
	3	0,5648	3,1729	36,48%	8,22%	0	0	0	0	0
Substituindo por pela média u	1	0,6872	3,4390	38,42%	11,20%	0	0	0	0	0
	2	0,4708	3,0130	43,66%	6,39%	0	0	0	0	0
	3	0,5648	3,1729	36,48%	8,22%	0	0	0	0	0
Substituindo pela interpolação linear	1	0,1827	2,8996	34,67%	2,10%	1	1	0	1	9
	2	0,2141	2,9458	34,42%	3,20%	0	0	0	0	0
	3	0,2214	2,9800	33,59%	3,32%	0	0	1	0	2

Fonte: Elaborada pelo autor.

Os resultados dos testes para algoritmos de correção com 1, 2 e 3 autoencoders empilhados, considerando as 3 estratégias de passagem de dados descritas anteriormente, estão descritos na Tabela 15. As colunas auxiliares de P1 a P4 servem para indicar qual configuração obteve o menor valor do parâmetro da seguinte forma:

- P1 indica qual configuração obteve menor REQM
- P2 indica qual configuração obteve menor EMA
- P3 indica qual configuração obteve EMP
- P4 indica qual configuração obteve menor MAPE

Analisando a Tabela 15, nota-se que a configuração que se sagrou vencedora de acordo com sistema de pontuação proposto foi a configuração com apenas 1 autoencoder e, consequentemente, 1 camada oculta. Esta configuração obteve menor valor de REQM, EMA, e MAPE, obtendo 9 pontos.

Definido, então, o número de camadas ocultas, faz-se necessário realizar experimentos a fim determinar o número de entradas do autoencoder, assim como o número de neurônios da camada oculta.

TABELA 16: SELECIONANDO O NÚMERO DE ENTRADAS E O NÚMERO DE NEURÔNIOS DA CAMADA OCULTA PARA O ALGORITMO DE CORREÇÃO: MELHOR RESULTADO.

Configuração do Autoencoder		**Métricas de Avaliação**				**Sistema de Pontuação**				
q	**NNCO**	**RMSE**	**EAM**	**ERM**	**MAPE**	**P1**	**P2**	**P3**	**P4**	**total**
5	4	0,1452	1,3929	40,14%	1,71%	1	0	0	1	7

Fonte: Elaborada pelo autor.

Na Tabela 16, é possível ver a configuração que obteve o melhor desempenho, contendo 5 entradas e 4 neurônios na camada oculta, obtendo nota 7, com menores valores de REQM e MAPE. A Tabela completa, que contém os resultados de todos os experimentos realizados a fim de selecionar a melhor configuração para o algoritmo de correção proposto, pode ser consultada no Apêndice B.

Em seguida, considerando-se a configuração com melhor desempenho considerando o experimento anterior, avaliou-se a influência do número de outliers inseridos sobre o funcionamento do subsistema de correção de outliers. Os experimentos estão resumidos na Tabela 17.

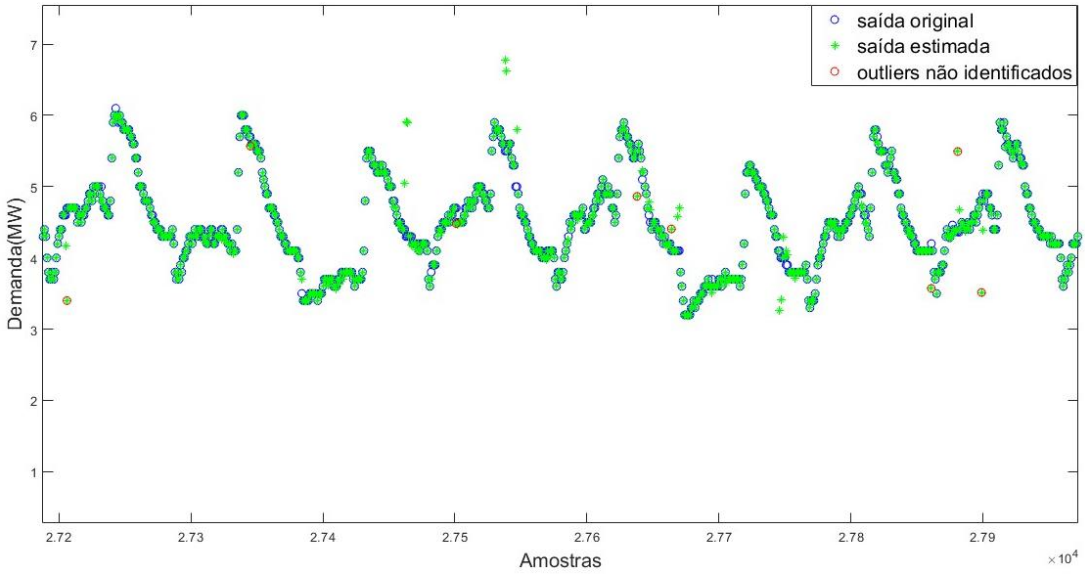
TABELA 17: AVALIANDO A INFLUÊNCIA DO NÚMERO DE OUTLIERS INSERIDOS SOBRE O FUNCIONAMENTO DO ALGORITMO DE CORREÇÃO PROPOSTO.

Nº de outliers em relação a L	**Métricas de Avaliação**				**Pontuação**						
	RMSE	**EAM**	**ERM**	**MAPE**	**p1**	**p2**	**p3**	**p4**	**total**	**CO_2e(mg)**	**mês-árvore**
2%	0,2216	2,9235	40,14%	1,96%	0	0	0	0	0	$8{,}57\times10^{-2}$	$9{,}35\times10^{-8}$
4%	0,1679	2,9235	40,14%	1,66%	0	0	0	0	0	$1{,}08\times10^{-1}$	$1{,}18\times10^{-7}$
5%	0,1592	2,9235	40,14%	1,68%	0	0	0	0	0	$1{,}05\times10^{-2}$	$1{,}14\times10^{-7}$
6%	0,1490	2,9155	40,14%	1,61%	0	1	0	1	5	$9{,}84\times10^{-2}$	$1{,}07\times10^{-7}$
8%	0,1678	2,9235	35,64%	1,83%	0	0	1	0	2	$9{,}20\times10^{-2}$	$1{,}00\times10^{-7}$
10%	0,1416	2,9155	35,64%	1,62%	1	1	1	0	8	$1{,}14\times10^{-1}$	$1{,}25\times10^{-7}$

Fonte: Elaborada pelo autor.

Analisando os resultados presentes na Tabela 13, observa-se que o algoritmo obteve melhor desempenho para o caso em que a quantidade de outliers inseridos é de 10% do número total de amostras.

FIGURA 25: CENÁRIO COM NÚMERO DE OUTLIERS EQUIVALENTE A 10% DO TOTAL DE AMOSTRAS.



Fonte: Elaborada pelo autor.

O cenário para o qual o algoritmo de correção obteve melhor desempenho dentre os testes descritos na Tabela 17, está representado na Figura 25. É importante notar que os pontos destacados em vermelho são outliers que não foram identificados no estágio anterior e, portanto, estão presentes na saída do algoritmo de correção, isto é, não foram corrigidos. Este fato demonstra a importância da etapa de detecção.

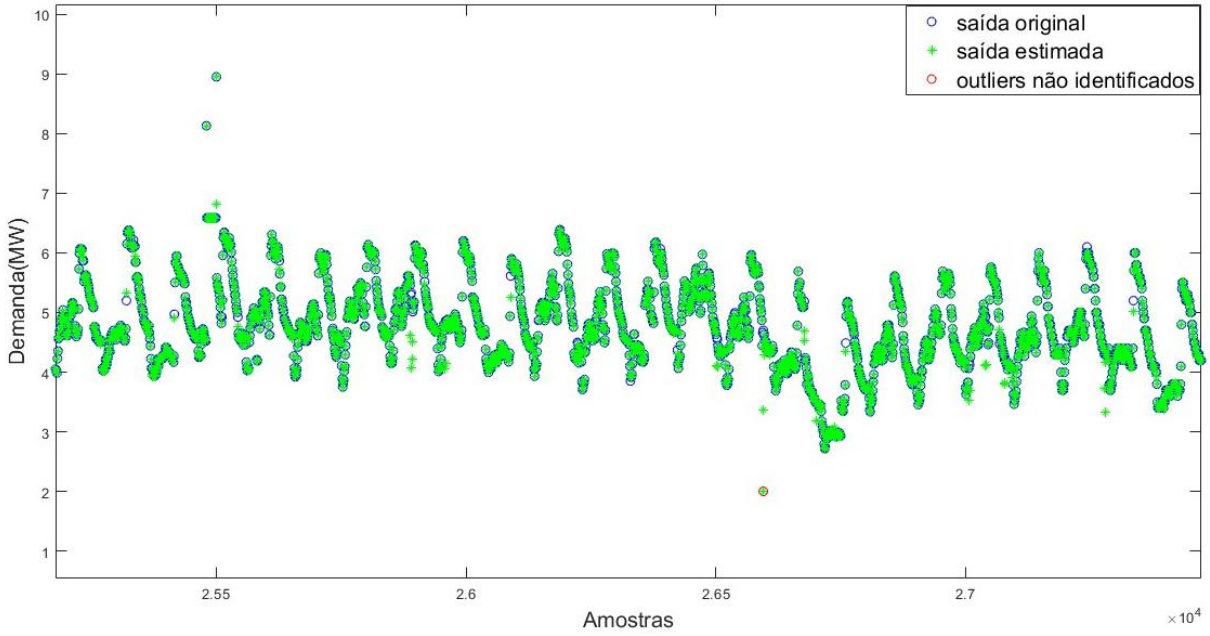
TABELA 18: AVALIANDO A INFLUÊNCIA DA AMPLITUDE DOS OUTLIERS INSERIDOS SOBRE O FUNCIONAMENTO DO ALGORITMO DE CORREÇÃO PROPOSTO.

Faixa de variação da amplitude dos outliers (em módulo)	**Métricas de Avaliação - autoencoder**				**Pontuação**						
	RMSE	**EAM**	**ERM**	**MAPE**	**p1**	**p2**	**p3**	**p4**	**total**	**CO_2e(mg)**	**mês-árvore**
de 0 a 1% de Pmax	0,1062	0,9842	22,46%	1,49%	0	0	0	0	0	$1,11x10^{-1}$	$1,21x10^{-7}$
de 0 a 25% de Pmax	0,1001	0,7987	16,76%	1,42%	1	1	1	1	11	$9,52x10^{-2}$	$1,04x10^{-7}$
de 0 a 50% de Pmax	0,1273	1,3929	40,14%	1,58%	0	0	0	0	0	$9,84x10^{-2}$	$1,07x10^{-7}$
de 0 a 75% de Pmax	0,1480	1,3929	40,14%	1,75%	0	0	0	0	0	$1,08x10^{-1}$	$1,18x10^{-7}$
de 0 a 100% de Pmax	0,1648	2,9235	40,14%	1,70%	0	0	0	0	0	$1,02x10^{-1}$	$1,11x10^{-7}$
de 0 a 200% de Pmax	0,1407	1,3929	40,14%	1,71%	0	0	0	0	0	$8,88x10^{-2}$	$9,69x10^{-8}$
Pior Cenário	0,1845	2,9235	40,14%	2,07%	0	0	0	0	0	$1,17x10^{-1}$	$1,28x10^{-7}$

Fonte: Elaborada pelo autor.

Em seguida, avaliou-se a influência da variação da amplitude dos outliers inseridos sobre a capacidade de reconstrução do algoritmo proposto. Os resultados estão resumidos na Tabela 18.

FIGURA 26: CENÁRIO PARA O QUAL AS AMPLITUDES DOS OUTLIERS ESTÃO CONTIDAS NO INTERVALO QUE VARIA DE 0 A 25% DE PMAX.



Fonte: Elaborada pelo autor.

Como é possível observar na Tabela 18, o algoritmo obteve melhor desempenho para o caso em que os outliers inseridos possuíam amplitude que variavam de 0 a 25% de **Pmax**. A representação deste cenário pode ser vista na Figura 26. Os resultados descritos na Tabela 18 também sugerem que há um aumento do erro médio quadrático com o aumento da amplitude dos outliers inseridos.

Do ponto de vista do impacto ambiental, considerando as Tabelas 17 e 18, calculou-se a média da a pegada de carbono do algoritmo de correção proposto, que foi de 0,0945mg de CO_2e assim como seriam necessários $1{,}11 \times 10^{-7}$ meses-árvores para sequestro deste carbono, em média.

4.4 COMPARAÇÃO 2: AUTOENCODERS X 3 ALGORITMOS DE CORREÇÃO TRADICIONAIS.

Por fim, a título de comparação, os mesmos dados utilizados na avaliação do algoritmo que compõe o subsistema de correção de outliers desenvolvido neste trabalho, foram submetidos a três algoritmos de correção tradicionais. O primeiro é o algoritmo de correção baseado em interpolação linear e, basicamente, substitui um ponto pela média entre seus dois pontos vizinhos, caso este seja identificado pelo subsistema anterior como um outlier. O algoritmo testa os pontos vizinhos para saber se são outliers. Caso não sejam rotulados como outliers, são utilizados na interpolação, caso contrário, os próximos vizinhos são testados. Este teste é feito para até três vizinhos consecutivos, caso haja outliers em 3 vizinhos consecutivos, o algoritmo utilizará o quarto par de vizinhos mais próximo, sem executar o teste.

Os outros dois foram retirados da biblioteca de preenchimento de dados do MATLAB: o primeiro deles é denominado “Nearest”, que substitui o ponto identificado como outlier pelo valor não atípico mais próximo; o segundo método, denominado de “Spline”, utiliza a interpolação cúbica por partes para substituir o dado identificado como outlier. (MATHWORKS, 2022; MOHANTY et al., 2016).

De forma análoga, estes algoritmos também foram submetidos aos cenários de A a F e de G a M, descritos na seção 3.1, nos quais o número de outliers inseridos e a amplitude dos outliers inseridos foram variados e os resultados podem ser vistos nas Tabelas de 19 a 24.

TABELA 19: AVALIANDO A INFLUÊNCIA DA VARIAÇÃO DO NÚMERO DE OUTLIERS INSERIDOS SOBRE O FUNCIONAMENTO DO ALGORITMO BASEADO EM INTERPOLAÇÃO LINEAR.

Nº de outliers em relação a L	Métricas de Avaliação				Pontuação -interpolação linear						
	RMSE	EAM	ERM	MAPE	p1	p2	p3	p4	total	CO_2e(mg)	mês-árvore
2%	0,2087	2,8657	40,77%	1,62%	0	1	0	0	2	$2,86x10^{-2}$	$3,12x10^{-8}$
4%	0,1533	2,8657	40,77%	1,30%	0	1	0	0	2	$3,17x10^{-2}$	$3,46x10^{-8}$
5%	0,1453	2,8657	40,77%	1,28%	0	1	0	0	2	$3,17x10^{-2}$	$3,46x10^{-8}$
6%	0,1332	2,8657	40,77%	1,25%	1	1	0	1	9	$2,86x10^{-2}$	$3,12x10^{-8}$
8%	0,1877	2,8657	32,02%	1,74%	0	1	1	0	4	$2,86x10^{-2}$	$3,12x10^{-8}$
10%	0,1429	2,8657	32,02%	1,39%	0	1	1	0	4	$3,49x10^{-2}$	$3,81x10^{-8}$

Fonte: Elaborada pelo autor.

TABELA 20: AVALIANDO A INFLUÊNCIA DA VARIAÇÃO DO NÚMERO DE OUTLIERS INSERIDOS SOBRE O FUNCIONAMENTO DO ALGORITMO NEAREST.

Nº de outliers em relação a L	Métricas de Avaliação				Pontuação-nearest						
	REQM	EMA	ERM	MAPE	p1	p2	p3	p4	total	CO_2e(mg)	mês-árvore
2%	0,2095	3,3683	40,77%	1,80%	0	0	1	0	2	$4,44x10^{-2}$	$4,85x10^{-8}$
4%	0,1919	3,3683	40,77%	1,87%	0	0	1	0	2	$7,62x10^{-2}$	$8,31x10^{-8}$
5%	0,1559	1,4000	40,77%	1,72%	1	0	1	1	9	$5,08x10^{-2}$	$5,54x10^{-8}$
6%	0,1987	3,6089	44,39%	1,79%	0	0	0	0	0	$4,44x10^{-2}$	$4,85x10^{-8}$
8%	0,1581	1,3883	40,77%	1,82%	0	1	1	0	4	$5,08x10^{-2}$	$5,54x10^{-8}$
10%	0,1573	1,4420	40,77%	1,76%	0	0	1	0	2	$5,76x10^{-2}$	$5,19x10^{-8}$

Fonte: Elaborada pelo autor.

TABELA 21: AVALIANDO A INFLUÊNCIA DA VARIAÇÃO DO NÚMERO DE OUTLIERS INSERIDOS SOBRE O FUNCIONAMENTO DO ALGORITMO SPLINE.

Nº de outliers em relação a L	Métricas de Avaliação				Pontuação - Spline						
	REQM	EAM	ERM	MAPE	p1	p2	p3	p4	total	CO_2e(mg)	mês-árvore
2%	0,1845	2,8413	40,77%	1,43%	0	0	1	0	2	$4,76x10^{-2}$	$5,19x10^{-8}$
4%	0,1505	2,8244	40,78%	1,39%	0	0	0	0	0	$6,03x10^{-2}$	$6,58x10^{-8}$
5%	0,1204	1,3883	40,77%	1,35%	0	1	1	0	4	$5,71x10^{-2}$	$6,23x10^{-8}$
6%	0,1486	3,0245	40,77%	1,33%	0	0	1	0	2	$5,08x10^{-2}$	$5,54x10^{-8}$
8%	0,1135	1,3883	40,77%	1,35%	1	1	1	0	8	$6,03x10^{-2}$	$6,58x10^{-8}$
10%	0,1190	2,3753	42,55%	1,31%	0	0	0	1	3	$5,39x10^{-2}$	$5,88x10^{-8}$

Fonte: Elaborada pelo autor.

Analisando as Tabela 19, 20 e 21, vemos que, em geral os algoritmos baseados em interpolação linear e o spline apresentam menores REQM e MAPE que o nearest, exceto para o cenário com NO = 8% de L, no qual o nearest apresenta menor REQM que o algoritmo baseado em interpolação linear, apenas. O erro máximo relativo é semelhante para os três algoritmos, com o nearest obtendo menor ERM para o cenário com NO = 6% de L e o algoritmo baseado em interpolação linear obtendo menor ERM em dois cenários (para NO = 8% e 10% de L). O nearest consegue obter menor EAM em um cenário (para NO = 10% de L). Entretanto, em contra partida, apresenta maior ERM em 3 cenários (NO = 2%, 4% e 6% de L).

Comparando o algoritmo de interpolação linear com o spline (Tabelas 19 e 21), vemos que o segundo apresenta menor REQM e MAPE praticamente em todos os cenários, exceto para os cenários com NO = 6% de L. O algoritmo baseado em interpolação linear apresenta menor MAPE em 50% por cento dos cenários (para NO = 4%, 5% e 6% de L). o spline obtem menor EAM praticamente em todos os cenários,

exceto para o cenário em que NO = 6%. Já o algoritmo baseado em interpolação linear obteve menor erro relativo médio em 2 cenários (NO = 8% e 10% de L), com ambos os algoritmos obtendo praticamente o mesmo valor de ERM em todos os outros cenários.

Confrontando-se a Tabela 17 com a Tabela 20, percebe-se que o algoritmo de detecção proposto apresenta menor RMSE em 3 cenários (NO =4%,6% e 10% de L), menor MAPE em 4 cenários (para NO = 4%, 5%, 6% e 10% de L) e menor ERM em todos os cenários. Em contrapartida, o nearest apresenta menor EAM em 50% dos cenários (para NO = 5%, 8% e 10% de L).

Confrontando-se a Tabela 17 com as Tabelas 19 e 21, percebe-se que o algoritmo de correção proposto obtém menor RMSE que o algoritmo baseado em interpolação linear em 2 cenários (NO = 8% e 10% de L). O algoritmo proposto também obtém menor ERM que o spline em todos os cenários assim como obtém menor ERM que o algoritmo baseado em interpolação linear em 4 cenários (para NO = 2%, 4%, 5% e 6% de L). O algoritmo proposto apresenta menor EAM que o spline para o cenário no qual NO = 6% de L. Em contrapartida, o spline obtém menores valores de RMSE e MAPE que o algoritmo proposto em todos os cenários. O algoritmo baseado em interpolação linear apresenta menor RMSE em 4 cenários (2%, 4%, 5% e 6% de L) e menores EAM e MAPE em todos os cenários.

TABELA 22: COMPARAÇÃO ENTRE OS MELHORES CENÁRIOS: AUTOENCODERS X 3 ALGORITMOS DE CORREÇÃO TRADICIONAIS.

Algoritmo	**Faixa de variação da amplitude dos outliers (em módulo)**	**Métricas de Avaliação**				**Pontuação**						
		RMSE	**EAM**	**ERM**	**MAPE**	**P1**	**P2**	**P3**	**P4**	**total**	**CO_2e(mg)**	**mês-árvore**
Baseado em Autoencoders	10%	0,1416	2,9155	35,64%	1,62%	1	1	1	0	8	$1,14x10^{-1}$	$1,25x10^{-7}$
Baseado em Interpolação Linear	6%	0,1332	2,8657	40,77%	1,25%	1	1	0	1	9	$2,86x10^{-2}$	$3,12x10^{-8}$
Nearest	5%	0,1559	1,4000	40,77%	1,72%	1	0	1	1	9	$5,08x10^{-2}$	$5,54x10^{-8}$
Spiline	8%	0,1135	1,3883	40,77%	1,35%	1	1	1	0	8	$6,03x10^{-2}$	$6,58x10^{-8}$

Fonte: Elaborada pelo autor.

A Tabela 22 resume os melhores resultados obtidos pelos algoritmos de correção proposto, o baseado em interpolação linear, nearest e spline, sob a ótica da variação do número de outliers inseridos. Através dela, é possível ver que dentre estes

cenários, o spline obteve menores RMSE e EAM, o algoritmo proposto obteve menor ERM e o algoritmo baseado em interpolação linerar obteve menor MAPE.

Em seguida, os 3 algoritmos de correção tradicionais foram expostos aos cenários onde avaliou-se a inflência da amplitude dos outliers, e os resultados estão disponíveis nas Tabelas 23, 24 e 25.

TABELA 23: AVALIANDO A INFLUÊNCIA DA AMPLITUDE DOS OUTLIERS INSERIDOS SOBRE O FUNCIONAMENTO DO ALGORITMO DE CORREÇÃO BASEADO EM INTERPOLAÇÃO LINEAR.

Faixa de variação da amplitude dos outliers (em módulo)	Métricas de Avaliação				Pontuação						
	RMSE	EAM	ERM	MAPE	p1	p2	p3	p4	total	CO_2e(mg)	mês-árvore
de 0 a 1% de Pmax	0,0872	0,7659	18,35%	1,09%	0	1	0	0	2	$2,86x10^{-2}$	$3,12x10^{-8}$
de 0 a 25% de Pmax	0,0869	0,8545	17,93%	1,09%	1	0	1	1	9	$3,17x10^{-2}$	$3,46x10^{-8}$
de 0 a 50% de Pmax	0,1195	1,4033	41,40%	1,24%	0	0	0	0	0	$2,86x10^{-2}$	$3,12x10^{-8}$
de 0 a 75% de Pmax	0,1403	1,6311	40,77%	1,41%	0	0	0	0	0	$2,86x10^{-2}$	$3,12x10^{-8}$
de 0 a 100% de Pmax	0,1507	2,8657	40,77%	1,34%	0	0	0	0	0	$3,17x10^{-2}$	$3,46x10^{-8}$
de 0 a 200% de Pmax	0,1257	1,3883	40,77%	1,30%	0	0	0	0	0	$3,17x10^{-2}$	$3,46x10^{-8}$
Pior Cenário	0,1510	2,8657	40,77%	1,48%	0	0	0	0	0	$3,17x10^{-2}$	$3,46x10^{-8}$

Fonte: Elaborada pelo autor.

Comparando as Tabelas de 23, 24 e 25, vemos que o algoritmo baseado em interpolação linear e o spline apresentam menores valores de RMSE e MAPE, em relação ao nerarest, em todos os cenários.

Confrontando as Tabelas 23 e 25, é possível observar que o algoritmo baseado em interpolação linear apresenta menor valor de MAPE, em relação ao spline, em 5 cenários (para as faixas de variação de 0 a 1%, de 0 a 2%, de 0 a 50%, de 0 a 100% e de 0 a 200% de Pmax). Por outro lado, o spline possui menor valor de RMSE, em relação ao algoritmo baseado em interpolação linear, em 4 cenários (para as faixas de variação de 0 a 50%, de 0 a 75%, de 0 a 100% de Pmax e u ±3dp).

Comparando a Tabela 18 com as Tabelas 23, 24 e 25, é possível observar que, em geral, o algoritmo baseado em interpolação linear e o spline apresentam menores valores de MAPE e de RMSE que o algoritmo de correção proposto. Por outro lado, o algoritmo de correção proposto apresenta, em geral, menor valor de ERM considerando os outros 3 algoritmos testados.

TABELA 24: AVALIANDO A INFLUÊNCIA DA AMPLITUDE DOS OUTLIERS INSERIDOS SOBRE O FUNCIONAMENTO DO ALGORITMO DE CORREÇÃO NEAREST.

Faixa de variação da amplitude dos outliers (em módulo)	Métricas de Avaliação				Pontuação - nearest						
	REQM	EAM	ERM	MAPE	p1	p2	p3	p4	total	CO_2e(mg)	mês-árvore
de 0 a 1% de Pmax	0,1387	1,0000	17,24%	1,63%	0	1	1	0	4	$4,44x10^{-2}$	$4,85x10^{-8}$
de 0 a 25% de Pmax	0,1371	1,4367	24,09%	1,61%	1	0	0	1	7	$4,13x10^{-2}$	$4,50x10^{-8}$
de 0 a 50% de Pmax	0,1624	1,4033	41,40%	1,78%	0	0	0	0	0	$5,08x10^{-2}$	$5,54x10^{-8}$
de 0 a 75% de Pmax	0,1742	1,5428	41,40%	1,84%	0	0	0	0	0	$4,44x10^{-2}$	$4,85x10^{-8}$
de 0 a 100% de Pmax	0,1706	1,4000	40,77%	1,84%	0	0	0	0	0	$4,76x10^{-2}$	$5,19x10^{-8}$
de 0 a 200% de Pmax	0,2040	3,3683	40,77%	1,94%	0	0	0	0	0	$4,44x10^{-2}$	$4,85x10^{-8}$
Pior Cenário	0,1761	3,3683	40,77%	1,79%	0	0	0	0	0	$4,13x10^{-2}$	$4,50x10^{-8}$

Fonte: Elaborada pelo autor.

TABELA 25: AVALIANDO A INFLUÊNCIA DA AMPLITUDE DOS OUTLIERS INSERIDOS SOBRE O FUNCIONAMENTO DO ALGORITMO DE CORREÇÃO SPLINE.

Faixa de variação da amplitude dos outliers (em módulo)	Métricas de Avaliação				Pontuação- spline						
	REQM	EAM	ERM	MAPE	p1	p2	p3	p4	total	CO_2e(mg)	mês-árvore
de 0 a 1% de Pmax	0,0957	2,2149	39,68%	1,17%	0	0	0	1	3	$4,76x10^{-2}$	$5,19x10^{-8}$
de 0 a 25% de Pmax	0,0898	0,7829	18,15%	1,21%	1	1	1	0	8	$4,76x10^{-2}$	$5,19x10^{-8}$
de 0 a 50% de Pmax	0,1164	1,3883	40,77%	1,34%	0	0	0	0	0	$5,08x10^{-2}$	$5,14x10^{-8}$
de 0 a 75% de Pmax	0,1312	2,5799	40,78%	1,34%	0	0	0	0	0	$5,08x10^{-2}$	$5,54x10^{-8}$
de 0 a 100% de Pmax	0,1250	1,3883	40,77%	1,42%	0	0	0	0	0	$5,39x10^{-2}$	$5,88x10^{-8}$
de 0 a 200% de Pmax	0,1534	2,8422	40,77%	1,46%	0	0	0	0	0	$4,76x10^{-2}$	$5,19x10^{-8}$
u ±3dp	0,1322	2,8403	40,77%	1,40%	0	0	0	0	0	$4,76x10^{-2}$	$5,19x10^{-8}$

Fonte: Elaborada pelo autor.

A Tabela 26 resume os melhores resultados obtidos pelos algoritmos de correção proposto, o baseado em interpolação linear, nearest e spline, sob a ótica da variação da amplitude dos outliers inseridos. Através da Tabela 26, é possível observar que, dentre estes cenários, o algoritmo baseado em interpolação linear obteve menor RMSE e MAPE, o spline obteve menor EAM e o algoritmo proposto obteve menor ERM.

TABELA 26: COMPARAÇÃO ENTRE OS MELHORES CENÁRIOS: AUTOENCODERS X 3 ALGORITMO DE CORREÇÃO TRADICIONAIS.

Algoritmo	Faixa de variação da amplitude dos	Métricas de Avaliação - Geral	Pontuação

	outliers (em módulo)	RMSE	EAM	ERM	MAPE	p 1	p 2	p 3	p 4	total	CO_2e(mg)	mês-árvore
Baseado em Autoencoders	de 0 a 25% de Pmax	0,1001	0,7987	16,76%	1,42%	1	1	1	1	11	$9,52x10^{-2}$	$1,04x10^{-7}$
Baseado em Interpolação Linear	de 0 a 25% de Pmax	0,0869	0,8545	17,93%	1,09%	1	0	1	1	9	$3,17x10^{-2}$	$3,46x10^{-8}$
Nearest	de 0 a 25% de Pmax	0,1371	1,4367	24,09%	1,61%	1	0	0	1	7	$4,13x10^{-2}$	$4,50x10^{-8}$
Spline	de 0 a 25% de Pmax	0,0898	0,7829	18,15%	1,21%	1	1	1	0	8	$4,76x10^{-2}$	$5,19x10^{-8}$

Fonte: Elaborada pelo autor.

Considerando as Tabelas 19 e 23, 20 e 24, e 21 e 25, foram calculas a média da pegada de carbono e da quantidade de meses-árvores necessários para sequestro do carbono gerado pela execução dos algoritmos baseado em interpolação linear, nearest e spline, obtendo-se 0,0305mg de CO_2e e $3,33x10^{-8}$ meses-árvores, 0,0491mg de CO_2e e $5,27x10^{-8}$, e 0,0520mg de CO_2e e $5,64x10^{-8}$ meses-ávores, respectivamente.

4.5 UM AUTOENCODER PARA CADA DIA DA SEMANA

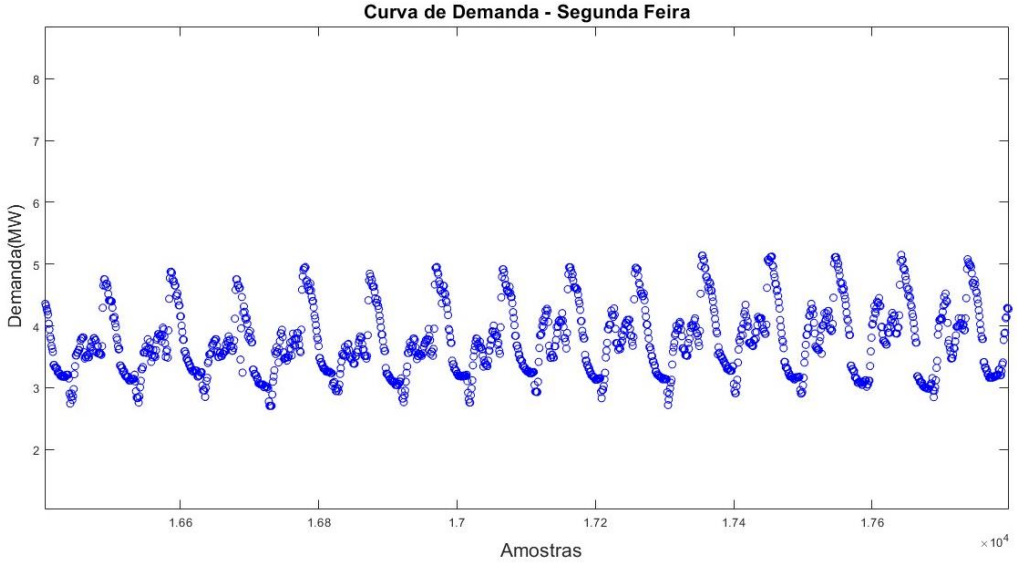
Com o intuito de obter melhora nos índices de detecção e correção já alcançados, foi promovida uma modificação no algoritmo proposto.

Primeiramente, a curva de demanda de potência foi dividida em 7 curvas menores, 1 para cada dia da semana, tendo em vista que o perfil das curvas pode ser diferente para diferentes dias da semana, o que pode levar, por exemplo, ao aumento do índice de falsos positivos previstos pelo sistema.

Por conseguinte, há um algoritmo de detecção, assim como o de correção de outliers, para cada dia da semana, que foram treinados com a respectiva curva de demanda. Por exemplo, há um subsistema de detecção e um de correção responsáveis por detectar e reconstruir possíveis outliers da terça feira, que foram treinados com a curva que contém dados ordenados apenas de terças-feiras.

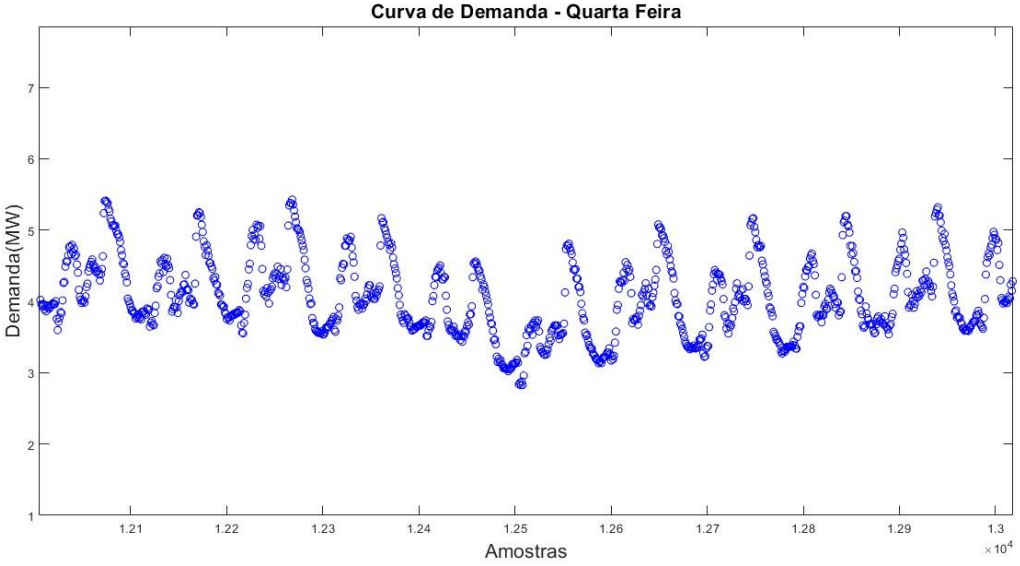
Portanto, toda metodologia foi mantida, com a diferença de que agora, o sistema como um todo é formado por 7 subsistemas e cada instância é submetida ao seu respectivo subsistema. Por exemplo, se determinada instância corresponde a dados de uma segunda feira, esta será submetido ao subsistema de detecção e de correção da segunda.

FIGURA 27: CURVA DE DEMANDA ORIGINAL PARA A SEGUNDA FEIRA



Fonte: Elaborada pelo autor.

FIGURA 28: CURVA DE DEMANDA ORIGINAL PARA A QUARTA FEIRA



Fonte: Elaborada pelo autor.

Como exemplo, trechos das curvas de demandas originais para a segunda e quarta feira podem ser vistas nas Figuras 27 e 28.

Em seguida, o algoritmo de detecção modificado foi submetido aos cenários já descritos no capítulo 3, para os quais avaliou-se a influência da variação do número de outliers inseridos. Os resultados podem ser vistos na Tabela 27.

TABELA 27: AVALIANDO A INFLUÊNCIA DO NÚMERO DE OUTLIERS INSERIDOS SOBRE O FUNCIONAMENTO DO ALGORITMO DE DETECÇÃO PROPOSTO MODIFICADO.

Nº de outliers em relação a L	Métricas de avaliação - 1 AE por dia					
	Acurácia	Precisão	Recall	f-score	CO_2e(mg)	mês-árvore
2%	99,45%	93,53%	77,94%	85,03%	$7{,}93\times10^{-2}$	$8{,}65\times10^{-8}$
4%	99,17%	94,08%	84,71%	89,15%	$8{,}25\times10^{-2}$	$9{,}00\times10^{-8}$
5%	98,99%	96,00%	83,25%	89,17%	$7{,}93\times10^{-2}$	$8{,}65\times10^{-8}$
6%	99,22%	94,30%	92,61%	93,45%	$7{,}62\times10^{-2}$	$8{,}31\times10^{-8}$
8%	97,71%	94,99%	75,47%	84,11%	$8{,}25\times10^{-2}$	$9{,}00\times10^{-8}$
10%	96,82%	89,67%	77,09%	82,91%	$1{,}11\times10^{-1}$	$1{,}21\times10^{-7}$

Fonte: Elaborada pelo autor.

Comparando as Tabelas 5 e 27, é possível ver que o algoritmo de detecção proposto modificado obteve maior Acurácia, Recall e f-score em 3 cenários (para NO = 2%, 4%, e 10% de L). O algoritmo proposto modificado obteve precisão inferior em praticamente todos os cenários, exceto para o cenário no qual NO = 8% de L.

Outra diferença consiste no fato de que o algoritmo de detecção proposto atinge seu melhor desempenho para o cenário no qual NO = 5% de L, enquanto o algoritmo de detecção proposto modificado atinge seu melhor desempenho para NO = 6% de L. O valor máximo dos parâmetros também é diferente. Enquanto o algoritmo proposto cosegue alcançar Acurácia e f-score de 99,74% e 97,43%, respectivamente, os valores máximos dos mesmos parâmetros alcançados pelo algoritmo proposto modificado é de 99,22% e 93,45%, respectivamente.

Posteriormente, o algoritmo de detecção modificado, considerando NO = 6% de L, cenário com melhor desempenho no experimento anterior, foi submetido aos cenários onde avaliou-se a influência da variação da amplitude dos outliers. Os resultados estão expostos na Tabela 28.

Comparando as Tabelas 6 e 28, observa-se que a metodologia proposta obtém Acurácia, Precisão, Recall e f-score igual ou superior em todos os cenários, exceto para o cenário cuja faixa de variação da amplitude dos outliers é de u ± 3dp.

TABELA 28: AVALIANDO A INFLUÊNCIA DA AMPLITUDE DOS OUTLIERS INSERIDOS SOBRE O FUNCIONAMENTO DO ALGORITMO DE DETCÇÃO PROPOSTO MODIFICADO.

Faixa de variação da amplitude dos outliers (em módulo)	Métricas de avaliação – 1AE por dia					
	Acurácia	Precisão	Recall	f-score	CO_2e(mg)	mês-árvore
de 0 a 1% de Pmax	100,00%	100,00%	100,00%	100,00%	$1,05x10^{-1}$	$1,14x10^{-7}$
de 0 a 25% de Pmax	99,97%	99,96%	99,54%	99,75%	$1,14x10^{-1}$	$1,25x10^{-7}$
de 0 a 50% de Pmax	99,66%	97,84%	96,45%	97,14%	$1,02x10^{-1}$	$1,11x10^{-7}$
de 0 a 75% de Pmax	98,78%	96,72%	82,62%	89,12%	$8,25x10^{-2}$	$9,00x10^{-8}$
de 0 a 100% de Pmax	96,11%	63,48%	83,42%	72,09%	$9,52x10^{-2}$	$1,04x10^{-7}$
de 0 a 200% de Pmax	98,60%	91,07%	85,17%	88,02%	$9,52x10^{-2}$	$1,04x10^{-7}$
u±3dp	98,68%	88,44%	89,77%	89,10%	$1,14x10^{-1}$	$1,25x10^{-7}$

Fonte: Elaborada pelo autor.

Depois disto, o algoritmo de correção proposto modificado foi submetido aos cenários onde o número de outliers inseridos é alterado. Os resultados podem ser consultados na Tabela 29, onde observa-se que o melhor desempenho foi atingido para NO = 8% de L.

TABELA 29: AVALIANDO A INFLUÊNCIA DA AMPLITUDE DOS OUTLIERS INSERIDOS SOBRE O FUNCIONAMENTO DO ALGORITMO DE CORREÇÃO PROPOSTO MODIFICADO.

Nº de outliers em relação a L	Métricas de Avaliação autoencoder				Pontuação - 1AE por dia						
	RMSE	EAM	ERM	MAPE	p1	p2	p3	p4	total	CO_2e(mg)	mês-árvore
2%	0,1581	2,4641	39,25%	1,78%	0	0	1	0	2	$1,43x10^{-1}$	$1,56x10^{-7}$
4%	0,1841	2,9129	40,15%	1,78%	0	0	0	0	0	$1,43x10^{-1}$	$1,56x10^{-7}$
5%	0,1667	2,9129	40,15%	1,70%	0	0	0	0	0	$1,46x10^{-1}$	$1,59x10^{-7}$
6%	0,1587	2,9243	39,25%	1,70%	0	0	1	0	2	$1,33x10^{-1}$	$1,45x10^{-7}$
8%	0,1297	2,2941	40,15%	1,61%	1	1	0	1	9	$1,78x10^{-1}$	$1,94x10^{-7}$
10%	0,1427	2,4641	39,25%	1,74%	0	0	1	0	2	$1,30x10^{-1}$	$1,42x10^{-7}$

Fonte: Elaborada pelo autor.

Comparando as Tabelas 17 e 29, observa-se que o algoritmo de correção proposto modificado obtém menores valores de RMSE e MAPE em apenas 2 cenários, para NO = 2% e 8% de L. O algoritmo de correção proposto modificado obtém menor EAM em todos os cenários, exceto para NO = 6% de L. O algoritmo proposto modificado obtém menor ERM em apenas 2 cenários, para NO = 2% e 6%.

Finalmente, o algoritmo de correção proposto modificado foi submetido a avaliação do ponto de vista da variação da amplitude dos outliers. Os resultados podem ser vistos na Tabela 30.

Comparando as Tabelas 18 e 30, observa-se que o algoritmo proposto obtém menor RMSE que o algoritmo proposto modificado em, praticamente, todos os cenários, exceto para o cenário no qual a faixa de variação da amplitude dos outliers é de u± 3dp.

TABELA 30: AVALIANDO A INFLUÊNCIA DA AMPLITUDE DOS OUTLIERS INSERIDOS SOBRE O FUNCIONAMENTO DO ALGORITMO DE CORREÇÃO PROPOSTO MODIFICADO.

Faixa de variação da amplitude dos outliers (em módulo)	Métricas de Avaliação - autoencoder				Pontuação						
	RMSE	EMA	ERM	MAPE	p1	p2	p3	p4	total	CO_2e(mg)	mês-árvore
de 0 a 1% de Pmax	0,1217	2,4641	42,34%	1,57%	0	1	0	0	2	$1,55x10^{-1}$	$1,70x10^{-7}$
de 0 a 25% de Pmax	0,1148	2,4641	34,02%	1,54%	1	1	1	1	11	$1,71x10^{-1}$	$1,87x10^{-7}$
de 0 a 50% de Pmax	0,1420	2,4641	39,25%	1,74%	0	1	0	0	2	$1,46x10^{-1}$	$1,59x10^{-7}$
de 0 a 75% de Pmax	0,1502	2,6891	40,15%	1,74%	0	0	0	0	0	$1,46x10^{-1}$	$1,59x10^{-7}$
de 0 a 100% de Pmax	0,1651	2,9129	35,19%	1,74%	0	0	0	0	0	$1,46x10^{-1}$	$1,59x10^{-7}$
de 0 a 200% de Pmax	0,1649	2,9129	38,19%	1,72%	0	0	0	0	0	$1,71x10^{-1}$	$1,87x10^{-7}$
u±3dp	0,1827	2,9129	40,15%	2,00%	0	0	0	0	0	$1,46x10^{-1}$	$1,59x10^{-7}$

Fonte: Elaborada pelo autor.

Observa-se também o algoritmo proposto obtém menor erro absoluto médio na maior parte dos cenários, exceto para os cenários nos quais a faixa de variação das amplitudes dos outliers é de 0 a 100 de Pmax, e de u ±3dp. O algoritmo proposto obtém menor MAPE para, praticamente, todos os cenários, exceto em 2 (quando a faixa de variação da amplitude dos outliers é de 0 a 75%, e de Pmax e para u±3dp)

De uma forma geral, dividir a curva de demanda em curvas por dia da semana não melhorou os índices para os algoritmos de detecção e correção.

Uma hipótese é que, ao dividir o banco de dados por 7, a rede de detecção, por exemplo, recebeu menos labels de outliers na etapa de treinamento (a quantidade de exemplo também ficou dividia por 7), o que pode causar essa pequena queda no desempenho do algoritmo.

Para avaliar esta hipótese, seria necessário um banco de dados com mais amostras. Por exemplo, seria interessante realizar testes com um banco de dados com o número de amostras 7 vezes maior que o número de amostras atual. Assim, a quantidade de outliers inseridos para treinamento de cada sub-rede responsável por cada dia da semana seria, aproximadamente, a mesma que a inserida nos testes iniciais, para o algoritmo proposto, facilitando a comparação.

Do ponto de vista do impacto ambiental, foram calculados, em média, considerando os dados da Tabela 27, 28 e 29 e 30, a pegada de carbono e a quantidade de meses-árvores necessária para que ocorra o sequestro deste carbono, para os algoritmos de detecção e correção propostos modificados, obtendo-se 0,0937mg de CO_2e e $1,02x10^{-7}$ meses-árvore, 0,1503mg de CO_2e e $1,64x10^{-7}$ meses-árvore, respectivamente.

TABELA 31: AVALIANDO O IMPACTO AMBIENTAL PARA OS ALGORITMOS DE DETECÇÃO TESTADOS NESTE TRABALHO.

Algoritmo de Detecção	Valores Médios	
	CO_2e(mg)	**mês-árvore**
Algoritmo proposto - 1 Autoencoder	$7,41x10^{-2}$	$8,08x10^{-8}$
Algoritmo proposto modificado - 7 Autoencoders	$9,37x10^{-2}$	$10,23x10^{-8}$
Desvio Padrão	$23,59x10^{-2}$	$64,81x10^{-8}$
MAD	$5,055x10^{-2}$	$5,52x10^{-8}$
i-forest	1,2423	$1,35x10^{-6}$

Fonte: Elaborada pelo autor.

TABELA 32: AVALIANDO O IMPACTO AMBIENTAL PARA OS ALGORITMOS DE CORREÇÃO TESTADOS NESTE TRABALHO.

Algoritmo de Correção	Valores Médios	
	CO_2e(mg)	**mês-árvore**
Algoritmo proposto - 1 Autoencoder	$9,45x10^{-2}$	$11,1x10^{-8}$
Algoritmo proposto modificado - 7 Autoencoders	$1,50x10^{-2}$	$1,64x10^{-8}$
Interpol Linear	$3,05x10^{-2}$	$3,33x10^{-8}$
Nearest	$4,91x10^{-2}$	$5,27x10^{-8}$
Spline	$5,2x10^{-2}$	$5,64x10^{-8}$

Fonte: Elaborada pelo autor.

A Tabela 31 reune os valores médios da pegada de carbono e da quantidade de meses-árvores necessária para que ocorra o sequestro deste carbono, considerando todos os algoritmos de detecção testados neste trabalho.

Analisando a Tabela 31, observamos que o algoritmo de detecção que apresenta menor impacto ambiental é baseado em MAD. Em contrapartida, o i-forest apresenta o maior impacto ambiental, com diferença de duas ordens de grandeza para o baseado em MAD. O algoritmo proposto obteve a segunda menor pegada de carbono.

A Tabela 32 reúne os valores médios da pegada de carbono e da quantidade de meses-árvores necessária para que ocorra o sequestro deste carbono, considerando todos os algoritmos de correção testados neste trabalho.

Analisando a Tabela 32, observa-se que o algoritmo que apresentou a maior pegada de carbono, e consequentemente a maior quantidade de meses-árvores, foi o algoritmo de correção proposto. Entretanto, o algoritmo que apresentou menor impacto ambiental foi o algoritmo proposto modificado.

As Tabelas 31 e 32 podem nos ajudar a estimar o impacto ambiental dos algoritmos envolvidos para uma situação cotidiana. Tomemos como exemplo um banco de dados com $4x10^{12}$ amostras que é gerado ao longo de um período e que deve ser submetido a metodologia de detecção proposta, com as redes já treinadas. Tendo em mente que as Tabelas foram confeccionadas considerando a quantidade de amostras de teste (20% de 200000 amostras) que é de 40000 amostras = $4x10^{4}$. Logo, o algoritmo de detecção proposto vai produzir uma pegada de carbono de $(4x10^{12})÷(4x10^{4}) \times 7,41x10^{-2}$mg de CO_2e = $7,41x10^{6}$mg de CO_2e = 7,41kg de CO_2e. De forma análoga, seriam necessários $(4x10^{12})÷(4x10^{4}) \times 8,08x10^{-8}$ meses-árvore = 8,08 meses-árvores para sequestro desse carbono produzido.

Um aspecto importante é que para estimar o impacto ambiental dos algoritmos propostos, foi considerado o tempo de execução dos algoritmos com as redes neurais artificiais já treinadas. Na prática, se o intervalo de tempo necessário para retreinar a rede neural for muito grande de forma que o período de tempo efetivo de execução dos programas seja muito maior, podemos desconsiderar o tempo de treinamento, como fizemos acima. No entanto, se as redes neurais aritificais precisam ser retreinadas com certa frequência de tal forma que o tempo de treinamento passe a ser significativo frente ao período de tempo que programa passa sendo executado, o tempo de treinamento deve ser considerado para fins de cálculo.

5 CONCLUSÃO

5 CONCLUSÃO

Este trabalho desenvolveu uma metodologia para detecção e correção de outliers baseado em autoencoders através do qual, foi possível avaliar o uso de autoencoders nas tarefas de detecção e correção de outliers em curvas de demanda elétrica, considerando dados reais de uma subestação.

Através do ajuste de parâmetros dos autoencoders, foi possível obter melhora no desempenho dos algoritmos de detecção e correção de outliers. Esta etapa também possibilitou selecionar a interpolação linear como melhor estratégia de passagem de dados, dentre as estratégias testadas.

Variando-se o número de outliers inseridos na etapa de treinamento, os testes sugerem que há um valor ótimo, ou uma faixa de valores ótima, para os quais o algoritmo de detecção proposto obtém melhor desempenho, resultado extremamente importante, visto que este é um parâmetro controlável.

Do ponto de vista da amplitude dos outliers inseridos, os resultados sugerem que há um aumento do erro médio quadrático com o aumento da amplitude dos outliers inseridos, para o algoritmo de correção proposto.

Durante os testes, o menor e maior MAPE obtidos pela metodologia proposta foram de 1,42% e 2,07%. Considerando o pior cenário, os resultados representam uma diminuição de 50,60% em comparação com ANDRADE et al. (2020), que obteve média dos erros relativos de 4,19%.

Considerando o cenário no qual os algoritmos foram treinados com o número de outliers equivalente a 5% do total de amostras, a metodologia proposta alcançou acurácia, precisão e f-score de 99,74%, 98,12% e 96,75%, respectivamente. Este resultado representa um aumento de 1,58% em relação a acurácia alcançada por (LIN; WANG, 2020), que foi de 98,19% assim como representa um incremento de cerca de 7% sobre a precisão e de 10% sobre o recall alcançados em (RADAIDEH et al., 2022), que foram de 91% e 88%, respectivamente.

Por outro lado, os resultados mostram que é possível substituir o sistema de correção baseado em autoencoders por outro baseado em interpolação linear, ou interpolação cubica segmentada spline, que obtiveram melhor desempenho, segundo o método de avaliação desenvolvido. Porém, caso o erro percentual máximo seja um

fator crítico no sistema, a metodologia de correção de outliers baseada em autoencoders apresentou melhores resultados, desse ponto de vista.

O desenvolvimento de uma metodologia que auxilia na seleção de parâmetros do autoencoder de forma a selecionar o modelo com melhor desempenho assim como o desenvolvimento de um sistema de pontuação capaz de eleger, dentro de um conjunto de configurações de um modelo estimativo, a que apresenta menores índices de erro se mostraram eficientes, tendo em vista os resultados alcançados.

Como trabalhos futuros desta pesquisa, é proposto:

- utilizar um banco de dados 7 vezes maior para testes com os algoritmos de detecção e correção propostos modificados, isto é, o algoritmo que contém um subsistema de detecção e detecção para cada dia da semana;
- avaliar o uso de outro tipo de autoencoder para o algoritmo de correção de outliers proposto, como os Variacionais ou redes GAN (Generative Adversarial Network), na tentativa de melhorar seu desempenho.
- Viabilizar uma forma de disponibilizar o algoritmo desenvolvido como, por exemplo, através de uma biblioteca do MATLAB, o que pode aumentar a visibilidade do trabalho, possibilitando que outros pesquisadores façam comparações e citem o trabalho, contribuindo para o desenvolvimento científico na área.
- Embarcar a metodologia desenvolvida em um hardware e aplica-la a máquinas e equipamentos industriais com o viés de detectar os outliers, isolá-los e analisa-los em busca de algum padrão relacionado a algum aspecto do funcionamento da máquina e/ou equipamento. Por exemplo, determinado padrão de ocorrência de outliers pode sugerir algum mal funcionamento ou pode sugerir alguma falha que está prestes a acontecer. Este hardware pode funcionar, por exemplo, como uma espécie de “caixa preta”, que contém informações que podem auxiliar a identificar o que aconteceu com a máquina e/ou equipamento antes de um defeito.
- Pode-se levar em consideração na metodologia de seleção de parâmetros e/ou no sistema de pontuação desenvolvidos, uma métrica que considere o gasto energético do algoritmo e, consequentemente, seu impacto ambiental. Por exemplo, no F-score, que faz uso do Recall e da Precisão, suponha que após uma análise, infere-se que para determinada aplicação, o Recall é menos

importante que o gasto energético. Pode-se, então, substituir a energia gasta na execução do algoritmo ou o gás carbônico equivalente pelo Recall na formula do F-score, que é uma média harmônica, criando-se uma nova métrica que leva em consideração o gasto energético e, consequentemente, o impacto sobre o aquecimento global.

REFERÊNCIAS

ABEDIN, Z.; BARUA, M.; PAUL, S.; AKTHER, S.; CHOWDHURY, R.; CHOWDHURY, M. S. U. "**A model for prediction of monthly solar radiation of different meteorological locations of Bangladesh using artificial neural network data mining tool**," 2017 International Conference on Electrical, Computer and Communication Engineering (ECCE), 2017, pp. 692-697.

AGGARWAL, C.C. (2017) **Outliers Analysis**, 2nd Edition, Springer, Cham, Switzerland.

ALLA, S.; ADARI, S.K. (2019) **Beginning Anomaly Detection Using Python-Based Deep Learning: With Keras and PyTorch**, Apress, New York, Unated States of America, ISBN: 978-1-4842-5176-8.

ANDRADE, P.H.M. "**Metodologia para Detecção e Correção de Outliers em Curvas de Potência de Subestação Utilizando Técnicas de Inteligência Artificial**", Dissertação de Mestrado, Dep. Eng. Elétrica, UFPB, 2018.

ANDRADE, P.H.M; VILLANUEVA, J.M.M., MACEDO, H.D. "**An Outliers Processing Module Based on Artificial Intelligence for Substations Metering System,**" in IEEE Transactions on Power Systems, vol 35, no. 5, pp.3400-3409, Sep. 2020.

BARAI, G.R.; KRISHNAN, S.; VENKATESH, B. "**Smart metering and functionalities of smart meters in smart grid - a review**," 2015 IEEE Electrical Power and Energy Conference (EPEC), 2015, pp. 138-145.

BARMAN, B. K.; YADAV, S. N.; KUMAR, S.; GOPE, S. "**IOT Based Smart Energy Meter for Efficient Energy Utilization in Smart Grid**," 2018 2nd International Conference on Power, Energy and Environment: Towards Smart Technology (ICEPE), 2018, pp. 1-5.

CHOI, J. S.; LEE S; CHUN S. J. "**A Queueing Network Analysis of a Hierarchical Communication Architecture for Advanced Metering Infrastructure"** in IEEE Transactions on Smart Grid, vol. 12, no. 5, pp. 4318-4326, Sept. 2021, doi: 10.1109/TSG.2021.3088879.

CHREN, S.; ROSSI, B.; PITNER, T. "**Smart grids deployments within EU projects: The role of smart meters**," 2016 Smart Cities Symposium Prague (SCSP), 2016, pp. 1-5.

DEVORE, J.L.; CORDEIRO, ANDRADE, M.T. **Probabilidade e estatística: para engenharia e ciências**. Cengage Learning Edições Ltda., 2014.

FREITAS, I.W.S. "**Um Estudo Comparativo de Técnicas de Detecção de Outliers no Contexto de Classificação de Dados**", Dissertação de Metrado, Dep. De Informática da UERN, 2019.

GÉRON, A. (2017) "**Hands-On Machine Learning With Scikit-Learn and TensorFlow**" O'Reilly, ISNB: 978-1-491-96229-9.

INTEL, "**Processador Intel Core I5-4210U",** Documentação oficial da Intel. Disponível em https://www.intel.com.br/content/www/br/pt/products/sku/81016/intel-core-i54210u-processor-3m-cache-up-to-2-70-ghz/specifications.html. Acesso em 03 de maio de 2023.

LANNELONGUE, L.; GREALEY, J., INOUYE, M., "**Green Algorithms: Quantifying the Carbon Footprint of Computation"**. Adv. Sci. 2021, 2100707.

LIGHT, J. "**Energy usage profiling for green computing**," 2017 International Conference on Computing, Communication and Automation (ICCCA), Greater Noida, India, 2017, pp. 1287-1291, doi: 10.1109/CCAA.2017.8230017.

LIN, Y.; WANG, J. "**Probabilistic Deep Autoencoder for Power System Measurement Outlier Detection and Reconstruction**," in IEEE Transactions on Smart Grid, vol. 11, no. 2, pp. 1796-1798, March 2020.

LISOWSKI, M.; MASNICKI R.; MINDYKOWSKI, J. **"PLC-Enabled Low Voltage Distribution Network Topology Monitoring,"** in IEEE Transactions on Smart Grid, vol. 10, no. 6, pp. 6436-6448, Nov. 2019, doi: 10.1109/TSG.2019.2904681.

MATHWORKS, "**Train Stacked Autoencoders for Image Classification"** Documentação Oficial do Matlab. Disponível em <https://la.mathworks.com/help/deeplearning/ug/train-stacked-autoencoders-for-image-classification.html>. Acesso em: 08 set. 2022.

MEHROTRA, K. G,; MOHAN, C.K.; HUANG, H. (2017) "**Anomaly Detection Principles and Algorithms"**, Springer, Cham, Switzerland, ISBN: 978-3-319-67526-8.

MOHANTY, P. K.; REZA, M.; KUMAR, P.; KUMAR, P. "**Implementation of Cubic Spline Interpolation on Parallel Skeleton Using Pipeline Model on CPU-GPU Cluster**," 2016 IEEE 6th International Conference on Advanced Computing (IACC), Bhimavaram, India, 2016, pp. 747-751, doi: 10.1109/IACC.2016.143.

MOSBAH, H.; EL-HAWARY, M. "**Multilayer artificial neural networks for real time power system state estimation**," 2015 IEEE Electrical Power and Energy Conference (EPEC), 2015, pp. 344-351.

MICROSOFT, "**Windows 10"**, site oficial do Sistema Operacional Windows, disponível em <https://www.microsoft.com/pt-br/windows/?r=1>. Acesso em 03 de maio de 2023.

NASCIMENTO, R.M.; OENING, A.P.; MARCÍLIO, D.C.; AOKI, A.R.; DE PAULA ROCHA, E.; SCHIOCHET, J.M. "**Outliers' detection and filling algorithms for smart metering centers,"** PES T&D 2012, 2012, pp. 1-6.

NEAGU B. C.; GRIGORAŞ, G.; SCARLATACHE, F. "**Outliers discovery from Smart Meters data using a statistical based data mining approach**," 2017 10th International Symposium on Advanced Topics in Electrical Engineering (ATEE), Mar. 2017, pp. 555-558.

NETO, J.T.C.; ANDRADE, P.H.M; VILLANUEVA, J.M.M.; SANTOS, F.A.O. "**Big Data Analytics of Smart Grids using Artificial Intelligence for the Outliers Correction at Demand Measurements,**" 2018 3rd International Symposium on Instrumentation Systems, Circuits and Transducers (INSCIT), Aug. 2018.

NETO, J.T.C. "**Processamento de Valores Atípicos em Redes Elétricas Inteligentes Baseado em Algoritmos Neuro-Fuzzy**", Dissertação de Mestrado, Dep. Eng. Elétrica, UFPB, 2018.

O.N.S. "**O que é ONS ?,**" Site oficial da ONS, Disponível em <http://www.ons.org.br/paginas/sobre-o-ons/o-que-e-ons>. Acesso em: 08 set. 2022.

RADAIDEH, M.; PAPPAS, C.; WALDEN, J.; LU, D.; VIDYARATNE, L.; BRITTON, T.; COUSINEAU, S. (2022). "**Time series anomaly detection in power electronics signals with recurrent and ConvLSTM autoencoders.**" Digital Signal Processing 130, 103704.

REZA, M. S.; CIOBOTARU, M.; AGELIDIS, V. G. "**Power System Frequency Estimation by Using a Newton-Type Technique for Smart Meters**," in IEEE Transactions on Instrumentation and Measurement, vol. 64, no. 3, pp. 615-624, March 2015, doi: 10.1109/TIM.2014.2347671.

SETZER, V. "**Dado, Informação, Conhecimento e Competência**", em *DataGramaZero – Revista de Ciência da Informação*, vol. 1, no. 0, pp. 1-10, Dez 1999.

SILVA, R.D.S. “**Contextualização do Setor Elétrico Brasileiro e o Planejamento da Infraestrutura no Longo Prazo**” Nota Técnica no. 69, Diretoria de Estudos e Políticas Setoriais de Inovação e Infraestrutura, IPEA, 2020.

SUN, L.; ZHOU, K.; ZHANG, X.; YANG S. "**Outlier Data Treatment Methods Toward Smart Grid Applications**," in IEEE Access, vol. 6, pp. 39849-39859, 2018.

TING, K.M. (2011). Confusion Matrix. In: Sammut, C., Webb, G.I. (eds) “**Encyclopedia of Machine Learning”**. Springer, Boston, MA.

GUNGOR V. C.; SAHIN D.; KOCAK, T.; ERGUT, S.; BUCCELLA, C.; FELLOW, C. C,; HANCKE, G.P. **"A Survey on Smart Grid Potential Applications and Communication Requirements,"** in IEEE Transactions on Industrial Informatics, vol. 9, no. 1, pp. 28-42, Feb. 2013, doi: 10.1109/TII.2012.2218253.

ZHONG, Y.; LI, Q.; HUANG, D.; HE, B.; SUN, L.; WANG, X. "**A Neural Network Approach to Wind Speed Prediction**," 2020 Asia Energy and Electrical Engineering Symposium (AEEES), 2020, pp. 788-794.

ZHOU, J.; HU, R. Q.; QIAN, Y. **"Scalable Distributed Communication Architectures to Support Advanced Metering Infrastructure in Smart Grid,"** in IEEE Transactions on Parallel and Distributed Systems, vol. 23, no. 9, pp. 1632-1642, Sept. 2012, doi: 10.1109/TPDS.2012.53.

APÊNDICE A

A Tabela 33 apresenta os resultados de todos os experimentos realizados com a finalidade de selecionar o número de entradas **q**, o número de neurônios da 1ª camada oculta **NNC1**, e da 2ª camada oculta **NNC2**, para os quais o algoritmo de detecção de outliers proposto obteve o melhor desempenho.

TABELA 33: SELEÇÃO DO NÚMERO DE ENTRADAS E DO NÚMERO DE NEURÔNIO DAS CAMADAS OCULTAS DO SUBSISTEMA DE DETECÇÃO.

Configuração do Autoencoder			**Métricas de Avaliação**			
q	**NNC1**	**NNC2**	**Acurácia**	**Precisão**	**Recall**	**F-score**
3	2	1	94,98%	*	0,00%	*
5	4	3	94,98%	*	0,00%	*
5	4	2	94,98%	*	0,00%	*
5	4	1	97,98%	99,34%	60,15%	74,93%
5	3	2	94,98%	*	0,00%	*
5	3	1	94,98%	*	0,00%	*
5	2	1	94,98%	*	0,00%	*
7	6	5	97,77%	100,00%	55,70%	71,55%
7	6	4	98,11%	98,67%	63,15%	77,01%
7	6	3	94,98%	*	0,00%	*
7	6	2	94,98%	*	0,00%	*
7	6	1	98,13%	99,06%	63,45%	77,35%
7	5	4	98,02%	98,02%	61,75%	75,77%
7	5	3	94,98%	*	0,00%	*
7	5	2	94,98%	*	0,00%	*
7	5	1	97,79%	100,00%	55,95%	71,75%
7	4	3	97,73%	100,00%	54,85%	70,84%
7	4	2	94,98%	*	0,00%	*
7	4	1	98,03%	96,84%	62,80%	76,19%
7	3	2	94,98%	*	0,00%	*
7	3	1	94,98%	*	0,00%	*
7	2	1	94,98%	*	0,00%	*
9	8	7	97,78%	100,00%	55,75%	71,59%
9	8	6	94,98%	*	0,00%	*
9	8	5	98,12%	98,90%	63,20%	77,12%
9	8	4	98,01%	98,71%	61,20%	75,56%
9	8	3	98,03%	98,41%	61,70%	75,85%
9	8	2	98,00%	99,02%	60,70%	75,26%
9	8	1	98,11%	98,98%	63,10%	77,07%
9	7	6	97,83%	100,00%	56,90%	72,53%

9	7	5	97,94%	100,00%	59,05%	74,25%
9	7	4	97,87%	100,00%	57,60%	73,10%
9	7	3	97,73%	100,00%	54,75%	70,76%
9	7	2	98,12%	97,64%	64,10%	77,39%
9	7	1	94,98%	*	0,00%	*
9	6	5	97,71%	100,00%	54,50%	70,55%
9	6	4	97,77%	100,00%	55,70%	71,55%
9	6	3	94,98%	*	0,00%	*
9	6	2	94,98%	*	0,00%	*
9	6	1	98,02%	98,95%	61,15%	75,59%
9	5	4	98,10%	98,29%	63,35%	77,04%
9	5	3	97,94%	100,00%	59,00%	74,21%
9	5	2	97,78%	100,00%	55,80%	71,63%
9	5	1	94,98%	*	0,00%	*
9	4	3	94,98%	*	0,00%	*
9	4	2	94,98%	*	0,00%	*
9	4	1	94,98%	*	0,00%	*
9	3	2	97,84%	100,00%	57,05%	72,65%
9	3	1	98,01%	99,67%	60,55%	75,33%
9	2	1	94,98%	*	0,00%	*
11	10	9	99,74%	98,47%	96,40%	97,42%
11	10	8	98,11%	99,13%	62,90%	76,97%
11	10	7	99,68%	97,46%	96,05%	96,75%
11	10	6	98,13%	98,61%	63,75%	77,44%
11	10	5	97,94%	100,00%	59,00%	74,21%
11	10	4	97,73%	100,00%	54,80%	70,80%
11	10	3	97,97%	99,17%	60,05%	74,81%
11	10	2	98,06%	97,82%	62,75%	76,45%
11	10	1	98,30%	99,33%	66,60%	79,74%
11	9	8	97,83%	94,24%	60,55%	73,73%
11	9	7	98,02%	98,17%	61,75%	75,81%
11	9	6	98,03%	98,64%	61,70%	75,92%
11	9	5	97,77%	100,00%	55,60%	71,47%
11	9	4	98,17%	99,77%	63,80%	77,83%
11	9	3	97,86%	100,00%	57,50%	73,02%
11	9	2	97,79%	100,00%	56,10%	71,88%
11	9	1	97,78%	100,00%	55,80%	71,63%
11	8	7	97,95%	100,00%	59,10%	74,29%
11	8	6	94,98%	*	0,00%	*
11	8	5	97,83%	100,00%	56,80%	72,45%
11	8	4	97,78%	100,00%	55,80%	71,63%
11	8	3	94,98%	*	0,00%	*
11	8	2	98,08%	98,97%	62,45%	76,58%

11	8	1	94,98%	*	0,00%	*
11	7	6	98,06%	97,52%	63,00%	76,55%
11	7	5	97,80%	100,00%	56,25%	72,00%
11	7	4	98,10%	99,44%	62,50%	76,76%
11	7	3	94,98%	*	0,00%	*
11	7	2	94,98%	*	0,00%	*
11	7	1	97,96%	98,69%	60,20%	74,78%
11	6	5	97,80%	100,00%	56,15%	71,92%
11	6	4	97,88%	100,00%	57,80%	73,26%
11	6	3	97,82%	100,00%	56,60%	72,29%
11	6	2	97,83%	99,91%	56,85%	72,47%
11	6	1	97,82%	100,00%	56,70%	72,37%
11	5	4	98,09%	99,21%	62,55%	76,72%
11	5	3	98,01%	99,27%	60,90%	75,49%
11	5	2	94,98%	*	0,00%	*
11	5	1	98,11%	98,90%	63,15%	77,08%
11	4	3	97,95%	100,00%	59,10%	74,29%
11	4	2	94,98%	*	0,00%	*
11	4	1	97,92%	100,00%	58,65%	73,94%
11	3	2	98,05%	99,59%	61,40%	75,97%
11	3	1	97,78%	100,00%	55,90%	71,71%
11	2	1	97,82%	100,00%	56,55%	72,25%
13	12	11	97,86%	100,00%	57,45%	72,98%

Fonte: Elaborada pelo autor.

APÊNDICE B

A Tabela 34 apresenta os resultados de todos os experimentos realizados com a finalidade de selecionar o número de entradas **q** e o número de neurônios da camada oculta **NNCO** para os quais o algoritmo de correção de outliers proposto obteve o melhor desempenho.

TABELA 34: SELECIONANDO O NÚMERO DE ENTRADAS E O NÚMERO DE NEURÔNIOS DA CAMADA OCULTA DO AUTOENCODER DO SUBSISTEMA DE CORREÇÃO.

Configuração do Autoencoder		**Métricas de Avaliação**				**Sistema de Pontuação**				
q	**NNCO**	**RMSE**	**EAM**	**ERM**	**MAPE**	**P1**	**P2**	**P3**	**P4**	**total**
3	2	0,1502	1,3001	38,18%	2,07%	0	0	0	0	0
3	1	0,1766	1,2715	37,29%	2,65%	0	1	1	0	4
5	4	0,1452	1,3929	40,14%	1,71%	1	0	0	1	7
5	3	0,1474	1,3810	39,80%	1,80%	0	0	0	0	0
5	2	0,1514	1,3564	39,09%	1,97%	0	0	0	0	0
5	1	0,1841	1,3461	38,79%	2,45%	0	0	0	0	0
7	6	0,1532	1,4624	42,14%	1,78%	0	0	0	0	0
7	5	0,1571	1,4732	42,45%	1,84%	0	0	0	0	0
7	4	0,1573	1,4663	42,25%	1,88%	0	0	0	0	0
7	3	0,1585	1,4549	41,93%	1,95%	0	0	0	0	0
7	2	0,1613	1,4303	41,22%	2,05%	0	0	0	0	0
7	1	0,2024	1,4205	40,94%	2,50%	0	0	0	0	0
9	8	0,1760	1,5241	43,92%	2,14%	0	0	0	0	0
9	7	0,1719	1,5226	43,88%	2,05%	0	0	0	0	0
9	6	0,1706	1,4901	42,94%	2,08%	0	0	0	0	0
9	5	0,1673	1,5194	43,78%	2,00%	0	0	0	0	0
9	4	0,1782	1,5242	43,92%	2,18%	0	0	0	0	0
9	3	0,1760	1,4865	42,83%	2,17%	0	0	0	0	0
9	2	0,1828	1,4818	42,70%	2,29%	0	0	0	0	0
9	1	0,2272	1,4712	42,39%	2,69%	0	0	0	0	0
11	10	0,1586	1,4674	42,29%	1,87%	0	0	0	0	0
11	9	0,1672	1,5144	43,64%	2,02%	0	0	0	0	0
11	8	0,1852	1,5260	43,97%	2,27%	0	0	0	0	0
11	7	0,1882	1,5404	44,39%	2,32%	0	0	0	0	0
11	6	0,1910	1,5406	44,40%	2,36%	0	0	0	0	0
11	5	0,1872	1,5307	44,11%	2,30%	0	0	0	0	0
11	4	0,1860	1,4756	42,52%	2,33%	0	0	0	0	0
11	3	0,2005	1,5072	43,43%	2,52%	0	0	0	0	0
11	2	0,2567	1,4950	43,08%	2,99%	0	0	0	0	0
11	1	0,2524	1,4956	43,10%	2,95%	0	0	0	0	0

Fonte: Elaborada pelo autor.