



UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS DE RUSSAS
CURSO DE GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

LEVI FREIRE DA ROCHA

**O USO DE APRENDIZADO DE MÁQUINA PARA O MONITORAMENTO DE
ESTRUTURAS DA CONSTRUÇÃO CIVIL**

RUSSAS

2021

LEVI FREIRE DA ROCHA

O USO DE APRENDIZADO DE MÁQUINA PARA O MONITORAMENTO DE
ESTRUTURAS DA CONSTRUÇÃO CIVIL

Trabalho de Conclusão de Curso apresentado ao
Curso de Graduação em Ciência da Computação
do Campus de Russas da Universidade Federal
do Ceará, como requisito parcial à obtenção do
grau de bacharel em Ciência da Computação.

Orientador: Prof. Dr. Alexandre Matos
Arruda

RUSSAS

2021

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária

Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

R574u Rocha, Levi Freire da.

O uso de aprendizado de máquina para o monitoramento de estruturas da construção civil / Levi Freire da Rocha. – 2021.

43 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Russas, Curso de Ciência da Computação, Russas, 2021.

Orientação: Prof. Dr. Alexandre Matos Arruda.

1. Monitoramento de saúde estrutural. 2. Aprendizado de máquina. 3. Algoritmos. 4. Redes neurais artificiais. I. Título.

CDD 005

LEVI FREIRE DA ROCHA

O USO DE APRENDIZADO DE MÁQUINA PARA O MONITORAMENTO DE
ESTRUTURAS DA CONSTRUÇÃO CIVIL

Trabalho de Conclusão de Curso apresentado ao
Curso de Graduação em Ciência da Computação
do Campus de Russas da Universidade Federal
do Ceará, como requisito parcial à obtenção do
grau de bacharel em Ciência da Computação.

Aprovada em:

BANCA EXAMINADORA

Prof. Dr. Alexandre Matos Arruda (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. Esequiel Fernandes Teixeira Mesquita
Universidade Federal do Ceará - UFC

Prof. Ms. Filipe Maciel Roberto
Universidade Federal do Ceará - UFC

Aos meus pais, que tanto se esforçaram para que eu alcançasse os meus objetivos. À Milena Cruz, que acompanhou de perto minha jornada e sempre me incentivou a continuar.

AGRADECIMENTOS

A Deus, pela minha vida, por ter me providenciado saúde quando precisei, por tão bem ter cuidado da minha família quando estive ausente, por ter me dado forças quando quis desistir e por nada ter me deixado faltar durante todo esse tempo. Porque sem Ele, nada disso seria possível.

Aos meus pais, por todo amor, cuidado e dedicação para proporcionar tamanha realização em minha vida, mesmo em meio a tantas dificuldades sempre lutaram para que eu alcançasse meus objetivos. Sempre serei grato por tudo que fazem por mim. Amo-os muito.

À minha namorada Milena Cruz, por apoiar minhas escolhas e me ajudar a progredir. Todo seu amor e dedicação fizeram total diferença quando precisei. Eu a amo demais.

Ao meu amigo Lucas Xavier, pelo companheirismo, brincadeiras e por tanto ter me ensinado. Sua dedicação e humildade o tornam uma pessoa digna de bastante sucesso. Obrigado por tudo, meu amigo.

Ao meu amigo Mikael Alves, pelos conselhos, aventuras e boas conversas. Você sempre me inspirou a ser uma pessoa melhor, aproveitar a vida e a me aventurar pelo mundo. A você, o meu muito obrigado.

À minha amiga Hellen Maria, por tanto ter me apoiado e me ouvido quando precisei. Sua força, coragem e dedicação me ensinaram que valia a pena continuar lutando. Sou muito grato pela sua amizade.

Aos professores Alexandre Arruma e Esequiel Mesquita, por todo suporte e por terem acreditado em mim. Seus conselhos e incentivos foram essenciais para a conclusão desta produção acadêmica.

À cidade e ao *campus* de Russas por, desde o início, tão bem terem me recebido.

Por fim, a todos aqueles que contribuíram direta e indiretamente para a minha graduação.

“O sucesso nasce do querer, da determinação e persistência em se chegar a um objetivo. Mesmo não atingindo o alvo, quem busca e vence obstáculos, no mínimo fará coisas admiráveis”

(José de Alencar)

RESUMO

O Monitoramento de Saúde Estrutural (MSE) caracteriza-se pela utilização de técnicas de coleta e de avaliação de dados de estruturas de engenharia, a fim de identificar e caracterizar seus danos, visando garantir sua segurança e integridade. Apesar de os custos associados ao MSE serem considerados elevados, a longo prazo a detecção com antecedência de danos pode representar significativa economia de recursos e prevenção de acidentes. Dessa forma, este trabalho busca desenvolver um método de monitoramento de estruturas da construção civil, de modo que se utilizem algoritmos de aprendizado de máquina e, com isso, caracterize-se o deslocamento relativo e auxiliar nas decisões de reparos. Para tanto, compararam-se os principais algoritmos de aprendizado de máquina por meio da criação de modelos preditivos a partir da utilização dos dados provenientes do monitoramento de saúde estrutural da igreja de Vila Nova de Foz Côa, em Portugal. Analisaram-se as métricas de desempenho obtidas com as previsões realizadas na base de dados de treinamento e, sobretudo, na base de testes como critérios de avaliação dos modelos. Nesse sentido, indicaram eminentes resultados aqueles criados a partir de algoritmos de redes neurais recorrentes como *Gated Recurrent Unit (GRU)* e *Long Short-Term Memory (LSTM)*. Portanto, os resultados obtidos sugerem quais algoritmos melhor se adaptaram aos dados e, por conseguinte, apresentaram superiores capacidades preditivas dada a base de dados utilizada nos experimentos, o que se configura como um importante estudo na elaboração de uma ferramenta de auxílio no monitoramento de saúde estrutural na construção civil.

Palavras-chave: Monitoramento de saúde estrutural. Aprendizado de máquina. Algoritmos. Redes neurais artificiais.

ABSTRACT

Structural Health Monitoring (SHM) is defined by the use of techniques for the gathering and assessment of data from engineering structures to identify and characterize their damage, aiming to ensure their safety and integrity. Although the costs associated with SHM are considered high, in the long-term, early detection of damage can represent significant savings in resources and accident prevention. Thus, this paper attempts to develop a civil engineering structures monitoring method so that machine learning algorithms are used, and, thereby, the relative displacement and aid in repair decisions are characterized. To this purpose, the main machine learning algorithms were compared, by creating predictive models based on the use of data from the structural health monitoring of the church of Vila Nova de Foz Côa, in Portugal. Were analyzed the performance metrics obtained with the forecasts made in the training database and, mainly, in the base of the test as criteria for evaluating the models. In this sense, those created from recurrent neural network algorithms such as Gated Recurrent Unit (GRU) and Long Short-Term Memory (LSTM) indicated eminent results. Therefore, the results obtained suggest which algorithms best adapted to the data and, consequently, presented superior predictive capacities given the database used in the experiments, which is configured as a relevant study in the elaboration of an aid tool in the structural health monitoring of civil engineering structures.

Keywords: Structural health monitoring. Machine learning. Algorithms. Artificial neural networks.

LISTA DE FIGURAS

Figura 1 – Esquema de um sistema de monitoramento de saúde estrutural utilizando sensores sem fio	18
Figura 2 – Exemplos de problemas de regressão e classificação para a previsão de condições climáticas	20
Figura 3 – Ilustração da lógica por trás do algoritmo Floresta Aleatória	21
Figura 4 – Processo de definição do hiperplano que melhor separa duas classes de dados	22
Figura 5 – Representação esquemática de uma RNN desenrolada	22
Figura 6 – Componentes de uma célula LSTM	23
Figura 7 – Componentes de uma célula GRU	24
Figura 8 – Posicionamento dos sensores na estrutura da Igreja de Foz Côa	26
Figura 9 – Representação gráfica dos dados de temperatura, umidade do ar e deslocamento do monitoramento da igreja de Foz Côa	29
Figura 10 – Representação gráfica do funcionamento do método da janela deslizante . .	30
Figura 11 – Previsões do modelo criado com o algoritmo Floresta Aleatória	35
Figura 12 – Previsões do modelo criado com o algoritmo Máquina de Vetores de Suporte	36
Figura 13 – Camadas ocultas do modelo LSTM	37
Figura 14 – Camadas ocultas do modelo LSTM Bidirecional	37
Figura 15 – Previsões do modelo criado com o LSTM	38
Figura 16 – Previsões do modelo criado com o LSTM Bidirecional	39
Figura 17 – Camadas ocultas do modelo GRU	39
Figura 18 – Previsões do modelo criado com o GRU	40

LISTA DE TABELAS

Tabela 1 – Resumo dos trabalhos relacionados	27
Tabela 2 – Especificações técnicas de hardware do <i>Google Colab</i>	33
Tabela 3 – Métricas de desempenho do algoritmo Floresta Aleatória	34
Tabela 4 – Métricas de desempenho do algoritmo SVM	35
Tabela 5 – Métricas de desempenho do algoritmo LSTM	38
Tabela 6 – Métricas de desempenho do algoritmo GRU	40
Tabela 7 – Quadro resumo das métricas de desempenho dos modelos avaliados	41

LISTA DE ABREVIATURAS E SIGLAS

AM	<i>Aprendizado de Máquina</i>
DNIT	<i>Departamento Nacional de Infraestrutura de Transportes</i>
GRU	<i>Gated Recurrent Unit</i>
LSTM	<i>Long Short-Term Memory</i>
MAE	<i>Mean Absolute Error</i>
MSE	<i>Monitoramento de Saúde Estrutural</i>
RBF	Função de Base Radial ou <i>Radial Basis Function</i>
RNA	<i>Redes Neurais Artificiais</i>
RNN	Redes Neurais Recorrentes ou <i>Recurrent Neural Networks</i>
SVM	Máquina de Vetores de Suporte ou <i>Support Vector Machine</i>

SUMÁRIO

1	INTRODUÇÃO	14
2	OBJETIVOS	16
2.1	Objetivo geral	16
2.2	Objetivos específicos	16
3	FUNDAMENTAÇÃO TEÓRICA	17
3.1	Monitoramento de Saúde Estrutural	17
3.2	Aprendizado de Máquina	18
3.2.1	<i>Aprendizagem Supervisionada</i>	18
3.2.2	<i>Algoritmos de Regressão e Classificação</i>	19
3.3	Floresta Aleatória	20
3.4	Máquina de Vetores de Suporte	21
3.5	Long Short-Term Memory	22
3.6	Gated Recurrent Unit	23
4	TRABALHOS RELACIONADOS	25
4.1	<i>Monitoramento a longo prazo de uma estrutura histórica danificada usando rede de sensores sem fio</i>	25
4.2	<i>Técnicas de aprendizado de máquina para o monitoramento de saúde estrutural</i>	26
4.3	<i>Monitoramento de saúde estrutural de pontes: uma abordagem baseada em Redes Neurais Artificiais livre de modelo para detecção de danos</i>	26
4.4	<i>Comparativo entre os trabalhos relacionados e o proposto</i>	27
5	METODOLOGIA	28
5.1	Aquisição dos dados	28
5.2	Preparação dos dados	29
5.3	Treinamento dos modelos	31
5.4	Método de avaliação	31
6	RESULTADOS E DISCUSSÕES	33
6.1	Resultados Floresta Aleatória	34
6.2	Resultados Máquina de Vetores de Suporte	35
6.3	Resultados LSTM	36

6.4	Resultados GRU	38
6.5	Conclusão	40
7	CONCLUSÕES E TRABALHOS FUTUROS	42
	REFERÊNCIAS	43

1 INTRODUÇÃO

O *Monitoramento de Saúde Estrutural* (MSE) caracteriza-se pela utilização de técnicas de coleta e de avaliação de dados de estruturas de engenharia, a fim de identificar e caracterizar seus danos, visando garantir sua segurança e integridade. Dessa forma, o MSE pode ser aplicado, por exemplo, para realizar o monitoramento de pontes que se encontram no final da vida útil ou em estruturas históricas danificadas que sofreram deformações no decorrer dos anos.

Conforme matéria publicada por Gomes (2018), no jornal Folha de S. Paulo, dados do *Departamento Nacional de Infraestrutura de Transportes* (DNIT) indicam que, das 8000 pontes e viadutos sob sua jurisdição, cerca de 1712, o que corresponde a 21% do total, necessitam de reparos. Essa porcentagem revela um número significativo de pontes que deveriam ter uma atenção especial por parte dos órgãos competentes para prevenir possíveis acidentes decorrentes da falta de manutenção. Além disso, a detecção tardia de danos estruturais pode elevar consideravelmente os custos com reparos, o que pode ser amenizado com o emprego de técnicas de MSE.

Dadas as aplicações e vantagens de se realizar o monitoramento de saúde estrutural na construção civil, deve-se buscar o desenvolvimento de técnicas que auxiliem nesse processo. Assim, Neves *et al.* (2017) propõem a utilização de *Redes Neurais Artificiais* (RNA) para a detecção de danos estruturais em pontes, por meio da utilização de dados de aceleração da superfície de uma ponte simulada como entrada do algoritmo e a realização de previsões das próximas acelerações. Smarsly *et al.* (2016) também utilizam RNA para a detecção de sensores defeituosos que possam comprometer a qualidade do monitoramento de estruturas, proporcionando assim maior confiabilidade e precisão.

O uso de *Aprendizado de Máquina* (AM) pode ser relevante no auxílio do monitoramento de saúde estrutural, uma vez que contribui diretamente para a preservação das estruturas, por meio da realização de previsões ou detecção de sensores defeituosos. Nessa perspectiva, o presente trabalho concentra-se em desenvolver uma técnica de auxílio ao MSE, comparando os principais algoritmos de aprendizado de máquina para a realização de previsões de danos estruturais em estruturas da construção civil, por meio da avaliação das métricas de desempenho obtidas a partir de dados reais e valores previstos por modelos criados com determinados algoritmos.

Os procedimentos metodológicos empregados no desenvolvimento deste trabalho consistem primeiramente na análise e pré-processamento dos dados coletados por sensores no MSE da igreja de Vila Nova de Foz Côa, em Portugal, trabalho desenvolvido por Mesquita *et al.*

(2018). Em seguida, são treinados modelos de aprendizado de máquina a partir dos algoritmos Floresta Aleatória, Máquina de Vetores de Suporte ou *Support Vector Machine* (SVM), *Long Short-Term Memory* (LSTM) e *Gated Recurrent Unit* (GRU). Por fim, os modelos são avaliados a partir das métricas de desempenho calculadas a partir da análise entre os valores reais da série e os valores previstos por cada um dos modelos.

O presente trabalho faz parte do projeto Indústria 4.0 - Transformação Digital na Área de Infraestrutura Rodoviária e Outras Ações de Tecnologia Ligadas à Gerência de Pavimentos do programa Cientista Chefe da Ciência e Inovação na Infraestrutura Viária do Ceará, tendo como entidade responsável a Secretaria da Infraestrutura do Estado do Ceará - SEINFRA/UFC.

As demais seções deste documento estão estruturados da seguinte maneira: o capítulo 2 aponta o objetivo geral e os específicos do desenvolvimento deste trabalho de pesquisa. Logo após, no capítulo 3, serão discutidos os principais conceitos que embasam o projeto. Já no capítulo 4, serão evidenciados os principais trabalhos relacionados a este. O capítulo 5 é focado nos procedimentos metodológicos do desenvolvimento do trabalho. Por último, os capítulos 6 e 7 apontarão os resultados obtidos com o projeto, bem como as conclusões e trabalhos futuros a serem implementados.

2 OBJETIVOS

2.1 Objetivo geral

Desenvolver um método de monitoramento de estruturas da construção civil utilizando algoritmos de aprendizado de máquina.

2.2 Objetivos específicos

- Caracterizar o deslocamento relativo de estruturas da construção civil;
- Utilizar algoritmos de aprendizado de máquina para auxiliar nas decisões de reparos em estruturas da construção civil;
- Comparar os algoritmos de aprendizado de máquina utilizados quanto à adaptação aos dados;
- Analisar os dados obtidos dos algoritmos para a detecção de danos.

3 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, serão introduzidos alguns conceitos essenciais para a melhor compreensão deste trabalho de pesquisa, destacando a aplicação do monitoramento de saúde estrutural em estruturas da construção civil e a definição de aprendizado de máquina. Nessa linha, ressaltase a aprendizagem supervisionada e a diferença entre problemas de regressão e classificação. Além disso, serão apresentados alguns dos principais algoritmos de aprendizado de máquina encontrados na literatura, como Floresta Aleatória, Máquina de Vetores de Suporte, LSTM e GRU.

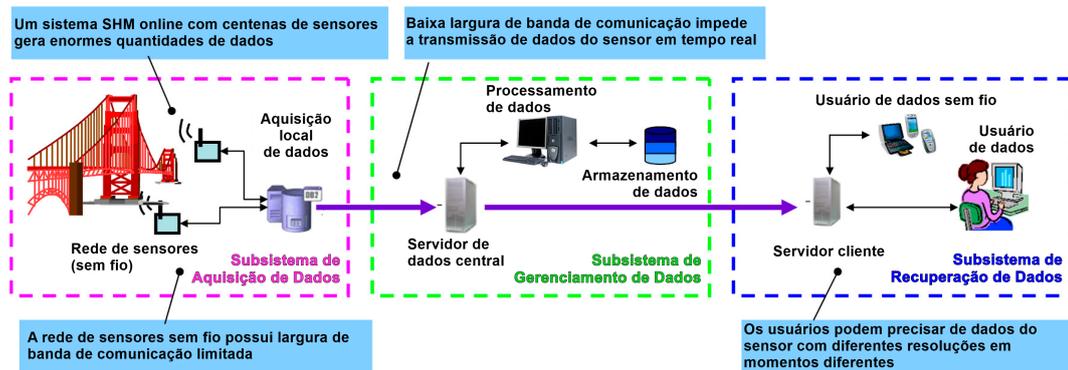
3.1 Monitoramento de Saúde Estrutural

O Monitoramento de Saúde Estrutural (do inglês, *Structural Health Monitoring (SHM)*) consiste na utilização de técnicas para a detecção de danos estruturais. Farrar e Worden (2006, p. 304) definem esse termo como “o processo de implementação de uma estratégia de identificação de danos para a infraestrutura aeroespacial, de engenharia civil e mecânica (SHM)”. Apesar dos diferentes usos do MSE, este trabalho concentra-se principalmente nas aplicações a estruturas de engenharia civil.

Uma das técnicas mais tradicionais para realizar o monitoramento de saúde estrutural, consiste na utilização de sensores instalados em uma estrutura de engenharia que coletam dados em tempo real das propriedades que se deseja analisar, como temperatura, umidade do ar, corrosão, tensão e deslocamento. A Figura 1 demonstra um esquema da utilização de sensores para o monitoramento de saúde estrutural com comunicação sem fio, proposto por Li *et al.* (2015). O esquema divide-se em três subsistemas que se comunicam. No primeiro quadrante pontilhado, ocorre a aquisição dos dados; no segundo, o gerenciamento e, no último, a recuperação dos dados.

O monitoramento de saúde estrutural de pontes consiste em uma importante linha de pesquisa devido à utilidade dessas estruturas para infraestrutura urbana. Em Comisu *et al.* (2017) é proposto um sistema para o monitoramento e avaliação da durabilidade de pontes. No texto, os autores ressaltam a importância de utilizar MSE para monitorar as pontes existentes à medida que elas envelhecem e fazer o acompanhamento da integridade estrutural de novas pontes, levando em consideração o histórico da infraestrutura civil da Romênia.

Figura 1 – Esquema de um sistema de monitoramento de saúde estrutural utilizando sensores sem fio



Fonte: Adaptado de Li *et al.* (2015).

3.2 Aprendizado de Máquina

Aprendizado de máquina (conhecido popularmente também por seu termo em inglês, *Machine Learning*) é um ramo da Inteligência Artificial definido de forma ampla por Mohri *et al.* (2018, p. 14) como “métodos computacionais, que usam a experiência para melhorar o desempenho ou fazer previsões precisas”.

De forma didática, podemos caracterizar a aprendizagem de máquina como sistemas que aprendem a partir de dados. Os algoritmos utilizam como entrada um conjunto de dados, pré-processados ou não, aprendem com esses dados identificando padrões e fornecem informações como saída que podem auxiliar na tomada de decisões.

As técnicas de aprendizagem de máquina têm influenciado positivamente diversas áreas do conhecimento nos últimos anos, principalmente em razão de suas aplicações no reconhecimento de imagem, análise de sentimentos, geração automática de legendas em vídeos, classificação de características, processamento de linguagem natural, serviços médicos, mercado financeiro, dentre várias outras atribuições.

Na construção civil, as técnicas de aprendizado de máquina podem ser utilizadas para detecção e classificação de defeitos de infraestrutura (FENG *et al.*, 2017). Neves *et al.* (2017) propõem uma abordagem de detecção de danos estruturais em pontes, utilizando redes neurais artificiais.

3.2.1 Aprendizagem Supervisionada

Os algoritmos de aprendizado de máquina podem ser classificados de acordo com seu tipo de aprendizagem. Dessa forma, existem três principais classes em que os algoritmos se

encaixam: aprendizagem supervisionada, aprendizagem não supervisionada e aprendizagem por reforço. Este trabalho se concentra na utilização dos algoritmos pertencentes à aprendizagem supervisionada. Desse modo, caso seja do interesse do leitor, as demais classes de algoritmos são definidas em Ayodele (2010).

Na aprendizagem supervisionada, um conjunto de dados de treinamento com as respostas corretas é fornecido. O algoritmo aprende com esses dados de entrada e generaliza-os para qualquer nova entrada possível (MARSLAND, 2014). Como exemplo, na área em que se baseia este trabalho, pode-se citar um conjunto de dados de entrada consistindo na temperatura ambiente e o valor do deslocamento de uma coluna danificada, coletados por sensores sem fio. Para cada entrada referente à temperatura, tem-se como resposta a esse dado o deslocamento relativo observado.

À vista disso, o aprendizado supervisionado é o mais indicado quando se possui dados rotulados e deseja-se encontrar uma função que melhor mapeie os dados de entrada para os de saída em um determinado problema. O modelo criado a partir de um algoritmo será capaz categorizar uma nova entrada ou prever um determinado valor, a depender do tipo de problema, como se explanará na subseção seguinte.

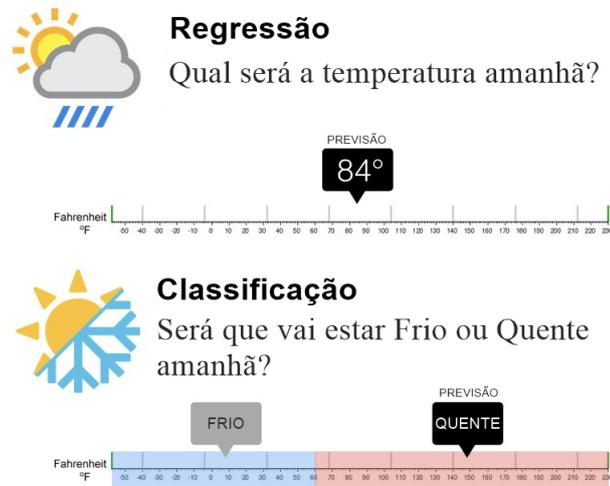
3.2.2 Algoritmos de Regressão e Classificação

A aprendizagem supervisionada é composta por duas classes de algoritmos que se diferenciam quanto à saída esperada. Quando o algoritmo se concentra em categorizar um determinado atributo, lida-se com um problema de classificação. Caso a saída esperada seja um número, a previsão de um valor, trata-se de um problema de regressão. Segundo Raschka (2015), o objetivo da classificação é prever rótulos para novas instâncias baseadas em observações anteriores. Já a regressão se trata da previsão de resultados contínuos.

Com o escopo de exemplificar melhor, considera-se como base a Figura 2. Em um cenário hipotético, no qual se almeja prever qual será a temperatura de amanhã, tem-se um problema de regressão, pois o objetivo é prever o valor da temperatura em °F. Considerando o mesmo exemplo, agora se quer definir se o dia de amanhã será quente ou frio. Esse, por sua vez, trata-se de um problema de classificação, já que o objetivo não é mais prever o valor da temperatura, e sim classificar o dia como quente ou frio.

Neste trabalho, os algoritmos de aprendizado de máquina se concentrarão em problemas relacionados à regressão. Dentre os algoritmos utilizados para regressão e que serão

Figura 2 – Exemplos de problemas de regressão e classificação para a previsão de condições climáticas



Fonte: Adaptado de Cislo (2019).

apresentados na próxima seção, estão Floresta Aleatória, Máquina de Vetores de Suporte, LSTM e GRU.

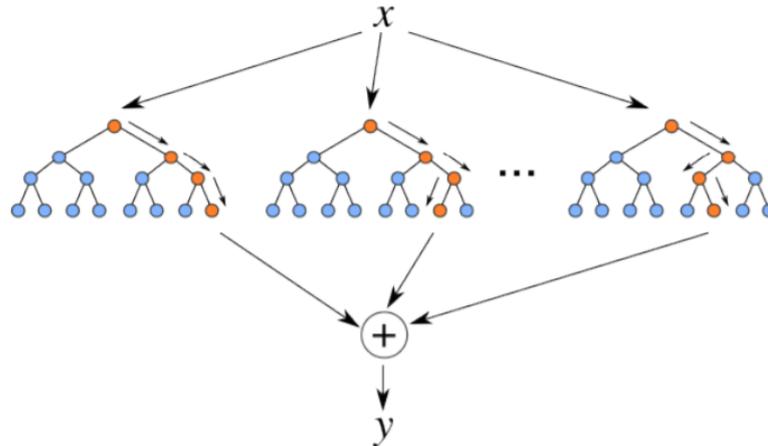
3.3 Floresta Aleatória

O algoritmo de aprendizado supervisionado Floresta Aleatória (do inglês, *Random Forest*) consiste no aperfeiçoamento da utilização dos modelos estatísticos de árvores de decisão, a fim de reduzir o problema de *overfitting*, comumente observado em árvores de decisão bem profundas.

O problema de *overfitting* se caracteriza pelo ajuste demasiado do modelo aos dados de treinamento. Assim, existe o risco da memorização de várias peculiaridades dos dados de treinamento ao invés de encontrar uma regra preditiva geral (DIETTERICH, 1995), o que pode levar a resultados não muito bons para um novo conjunto de dados de entrada.

Dessa maneira, o algoritmo Floresta Aleatória é definido em Breiman (2001, p. 6) como "um classificador que consiste em uma coleção de classificadores estruturados em árvore $\{h(\mathbf{x}, \Theta_k), k = 1, \dots\}$ em que os $\{\Theta_k\}$ são vetores aleatórios distribuídos de forma idêntica e independentes e cada árvore emite um voto unitário para a classe mais popular na entrada \mathbf{x} ". Esse conceito pode ser melhor ilustrado por meio da Figura 3, em que a entrada X é submetida a diferentes árvores de decisão, e a saída Y é resultado da resposta mais popular retornada.

Figura 3 – Ilustração da lógica por trás do algoritmo Floresta Aleatória



Fonte: Lorenzetti e Telöcken (2016).

3.4 Máquina de Vetores de Suporte

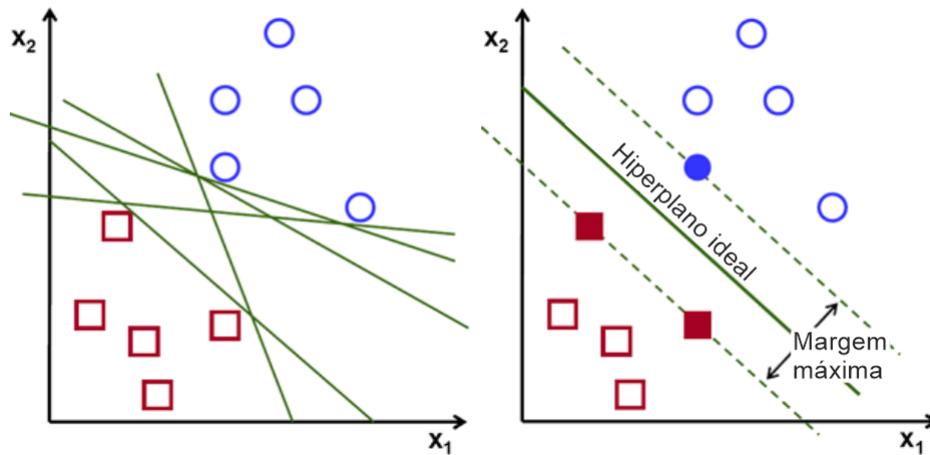
O algoritmo máquina de vetores de suporte (SVM, do inglês, *support vector machines*) tornou-se bastante popular nos últimos anos, por apresentar resultados bastantes satisfatórios para problemas de aprendizado de máquina tanto de classificação como de regressão, apresentando desempenho superior a muitos métodos populares de classificação como árvores de decisão, regressão logística e *naive bayes*.

Originalmente proposto por Cortes e Vapnik (1995), o algoritmo SVM pode ser entendido como o mapeamento dos vetores de entrada para uma superfície n-dimensional, cujo objetivo é encontrar um hiperplano que melhor separe classes de dados. Nesse sentido, deve-se manter a melhor margem de distância para os vetores de suporte.

Na Figura 4, à esquerda, observa-se a definição de diversas retas que separam os dados representados por bolas e quadrados em duas classes. Como a finalidade do algoritmo SVM é encontrar o melhor hiperplano separador das classes, ver-se na figura à direita a definição do hiperplano ideal que mantém a margem máxima em relação aos vetores de suporte representados pelas bolas e quadrados preenchidos.

Nem todos os problemas podem ser linearmente separáveis de forma eficiente, logo o algoritmo SVM faz o uso de funções matemáticas, conhecidas como *kernels*, para fazer o mapeamento dos dados de entrada de natureza não linear para um problema linear. Alguns exemplos de funções *kernel* comuns são: Linear, *Gaussian*, *Polynomial* e *Tangent Hyperbolic*.

Figura 4 – Processo de definição do hiperplano que melhor separa duas classes de dados

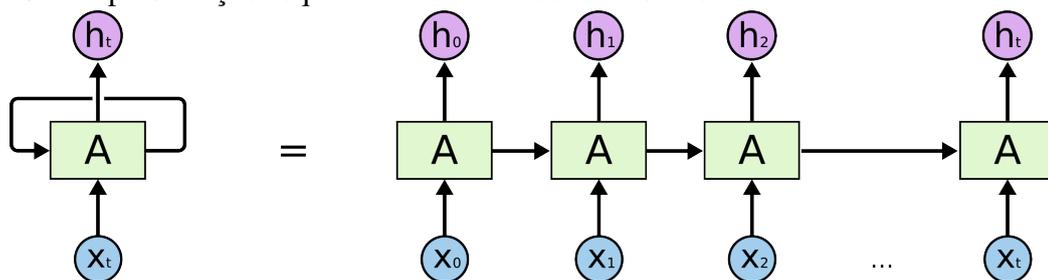


Fonte: Drakos (2018).

3.5 Long Short-Term Memory

As Redes Neurais Recorrentes ou *Recurrent Neural Networks* (RNN) são capazes de aprender dependências entre os dados em problemas de ordem sequencial. Para isso, a saída produzida em uma célula é repassada para a próxima etapa, mantendo em memória as observações anteriores. Isso se dá por intermédio de um mecanismo de recorrência conforme ilustrado na Figura 5. Conforme Olah (2015), as RNNs são indicadas para problemas que lidam com dados em sequência como reconhecimento de fala, modelagem de linguagem, tradução e legendagem de imagens.

Figura 5 – Representação esquemática de uma RNN desenrolada



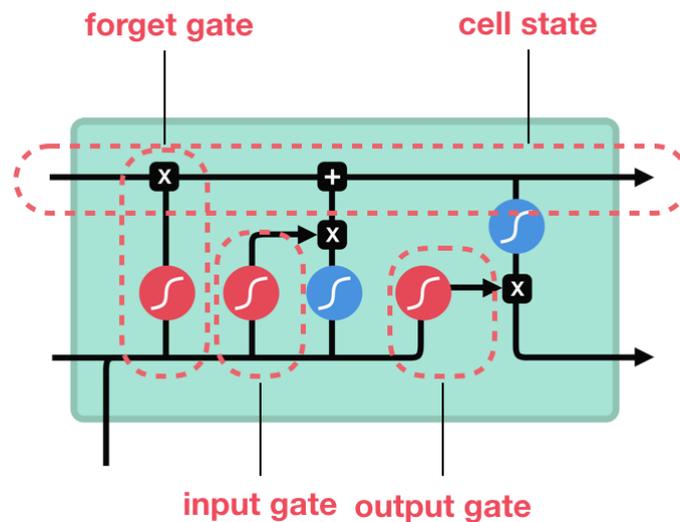
Fonte: Olah (2015).

Entretanto, esse tipo de rede neural apresenta um problema de memória de curto prazo, não sendo indicada para problemas que envolvam grandes sequências de informação. Isso ocorre devido ao desaparecimento do gradiente, responsável pela atualização dos pesos da rede, à proporção que se torna tão pequeno, não contribui de forma efetiva para o aprendizado da rede.

As redes LSTM foram propostas por Hochreiter e Schmidhuber (1997) e têm por objetivo evitar o problema da dependência de longo prazo. Para isso, introduziu-se o conceito de

estado da célula, no qual adicionam-se ou removem-se informações da memória da rede. Para isso, utilizam-se mecanismos intitulados portões (*gates*, em inglês) que coordenam o processo de inclusão ou retirada de informações do estado da célula, bem como a saída para a próxima etapa. A Figura 6 ilustra o estado de uma célula LSTM e os portões: *forget gate*, *input gate* e *output gate*.

Figura 6 – Componentes de uma célula LSTM



Fonte: Adaptado de Phi (2018).

O LSTM é uma arquitetura de rede neural artificial que tem apresentado bons resultados para problemas que lidam com dados em série temporal. O que caracteriza esse tipo de arquitetura é a capacidade de armazenar informações por longos períodos de tempo, evitando assim o problema de dependência de longo prazo. Neste trabalho, os dados coletados pelos sensores se caracterizam como uma série temporal, o que torna o algoritmo LSTM um ótimo candidato a apresentar bons resultados ao trabalhar com esse problema de previsão de danos estruturais, já que se destaca a importância de se considerar as informações passadas para a compreensão do que poderá vir a ocorrer.

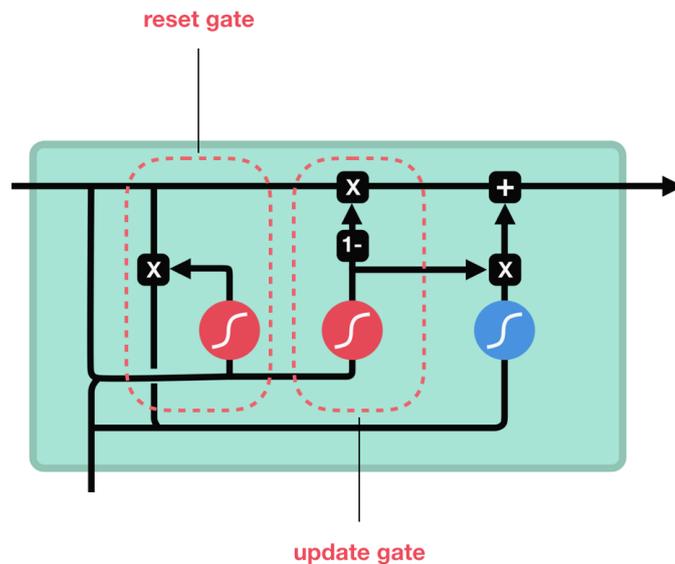
3.6 Gated Recurrent Unit

As redes GRU foram introduzidas em Cho *et al.* (2014) e são outro tipo de RNN que busca resolver o problema da dependência de curto prazo. Elas se caracterizam como uma versão simplificada das redes LSTM, devido a sua relativa diminuição de operações no interior da célula, evidenciada pela redução no número de portões.

Segundo Phi (2018), as GRU se desfizeram do estado da célula existente no LSTM e utilizam o estado oculto para a transferência de informações, bem como faz uso de apenas dois portões: portão de atualização (*update gate*) e o portão de reinicialização (*reset gate*). Isso não necessariamente significa uma perda na capacidade preditiva de modelos quando comparada às redes LSTM, uma vez que pode apresentar resultados bem próximos ou tão bons quanto. Entretanto, essa mudança pode resultar em reduções no tempo de treinamento de modelos, uma vez que há um menor número de operações despendidas no interior da célula.

A Figura 7 evidencia uma célula GRU com seus dois portões usados para adição ou exclusão de informações (*update gate*) e quantas informações anteriores devem ser esquecidas (*reset gate*).

Figura 7 – Componentes de uma célula GRU



Fonte: Phi (2018).

4 TRABALHOS RELACIONADOS

Nesta seção, será apresentado o estado da arte para o monitoramento de saúde estrutural na engenharia civil, destacando a utilização das técnicas de aprendizado de máquina como suporte à tomada de decisões. Na Seção 4.1, é apresentada uma abordagem de monitoramento de uma estrutura histórica danificada, utilizando sensores sem fio. Já na Seção 4.2, é evidenciada a importância de garantir a calibração dos sensores utilizados no MSE, propondo uma estratégia de detecção de sensores com falhas - uso de redes neurais artificiais. Na Seção 4.3 é apresentada uma abordagem de detecção de danos estruturais em pontes - uso de redes neurais artificiais. Por fim, a Seção 4.4 resume as diferenças e semelhanças entre os trabalhos relacionados e o proposto.

4.1 *Monitoramento a longo prazo de uma estrutura histórica danificada usando rede de sensores sem fio*

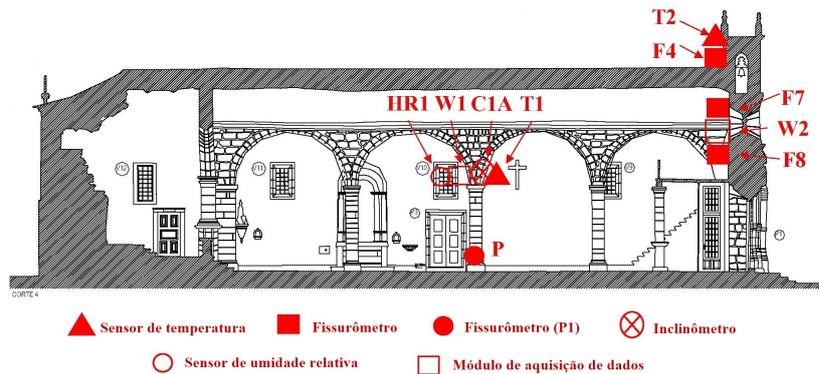
Preservar o patrimônio histórico é de suma importância para conservar a identidade e a história de uma comunidade. Logo, faz-se necessária a utilização de técnicas de monitoramento de saúde estrutural para avaliar e garantir a segurança dessas estruturas tão importantes para a sociedade.

Dessa forma, Mesquita *et al.* (2018) apresentam uma abordagem para realizar o monitoramento de saúde estrutural de uma estrutura histórica danificada do século XVI denominada Igreja de Vila Nova de Foz Coa, em Portugal. Ela sofreu danos estruturais em algumas colunas e na fachada principal, relacionados a abalos sísmicos datados dos anos 1755 e 1969.

Para realizar o monitoramento estrutural da Igreja de Foz Côa, instalou-se estrategicamente em uma coluna danificada e na fachada principal da igreja, um sistema sensorial composto por sensores de temperatura, umidade relativa do ar e deslocamento. A Figura 8 ilustra a distribuição desses sensores na estrutura da igreja, que fizeram a coleta de dados durante o período de um ano, de 13 de março de 2015 a 29 de março de 2016.

A partir dos estudos realizados com os dados coletados pelos sensores, os autores concluíram que os movimentos observados nos elementos monitorados não estão relacionados a questões estruturais, mas sim às influências da temperatura e da umidade relativa do ar. Por fim, inferiu-se, ainda, que os pequenos deslocamentos observados durante o estudo não apresentam

Figura 8 – Posicionamento dos sensores na estrutura da Igreja de Foz Côa



Fonte: Mesquita *et al.* (2018).

risco à integridade da igreja de Vila Nova de Foz Côa.

4.2 Técnicas de aprendizado de máquina para o monitoramento de saúde estrutural

Para que seja realizado o monitoramento de saúde estrutural de uma estrutura de engenharia utilizando sensores sem fio, instalados em seus elementos, é necessária a garantia de que esses sensores sejam fiéis na coleta das informações. Sensores descalibrados podem interferir nos resultados do monitoramento, o que pode comprometer a confiabilidade do sistema MSE.

A fim de realizar uma revisão das técnicas de aprendizado de máquina para o monitoramento de saúde estrutural, Smarsly *et al.* (2016) propõem uma abordagem de aprendizado de máquina para detecção autônoma e descentralizada de falhas em sensores sem fio, com o intuito de facilitar o monitoramento de saúde estrutural confiável e preciso.

A abordagem apresentada pelos autores consiste na utilização de redes neurais artificiais para detectar a existência de falhas em sensores sem fio utilizados no monitoramento de saúde estrutural. Para a realização dos testes, injetaram-se nos dados duas falhas comuns a sensores, após a aplicação da rede neural artificial identificaram-se essas falhas com sucesso, comprovando a eficácia do método proposto.

4.3 Monitoramento de saúde estrutural de pontes: uma abordagem baseada em Redes Neurais Artificiais livre de modelo para detecção de danos

Nem sempre é possível realizar o monitoramento de saúde estrutural de forma prática - instalar sensores em estruturas reais de engenharia, a fim de coletar suas características e se

analisar suas condições estruturais. Dessa forma, para fins de estudo e desenvolvimento de novas técnicas, podem ser realizadas simulações em laboratório do comportamento de estruturas de engenharia sob efeito de determinadas condições definidas no projeto de simulação.

Partindo dessa premissa, Neves *et al.* (2017) propõem uma abordagem de detecção de danos estruturais em pontes, a partir de simulações de passagens de trem em uma ponte ferroviária fictícia, a fim de realizar a coleta de dados para serem utilizados em um modelo preditivo, desenvolvido pelo uso de redes neurais artificiais. Dados esses, que em um cenário real, poderiam ser obtidos a partir de uma rede de sensores sem fio instalada em uma estrutura de engenharia, sem a necessidade do desenvolvimento de modelos numéricos complexos.

A partir desse estudo, identificou-se a possível necessidade de se considerar as condições ambientais, como a temperatura do ambiente, para um processo de detecção de danos mais eficaz, a evidenciar os efeitos que as condições ambientais exercem sobre as estruturas de engenharia. Por conseguinte, o modelo desenvolvido - redes neurais artificiais - poderia apresentar resultados expoentes, com mais precisão de previsão.

4.4 *Comparativo entre os trabalhos relacionados e o proposto*

A Tabela 1 apresenta um quadro resumo dos trabalhos relacionados e o proposto neste trabalho de conclusão de curso. Através desse, evidencia-se as diferenças e semelhanças quanto ao tipo de estrutura estudada, assim como o propósito de desenvolvimento do trabalho e os algoritmos utilizados.

Tabela 1 – Resumo dos trabalhos relacionados

Trabalho	Estrutura	Propósito	Algoritmos
Mesquita <i>et al.</i> (2018)	Estrutura histórica (igreja)	Monitoramento de uma estrutura histórica danificada usando sensores sem fio	Não utiliza
Smarsly <i>et al.</i> (2016)	Pontes	Detecção de falhas em sensores	Redes neurais artificiais
Neves <i>et al.</i> (2017)	Pontes	Detecção de danos estruturais	Redes neurais artificiais
Presente trabalho	Estruturas da construção civil	Detecção de danos e análise comparativa dos algoritmos	Floresta aleatória, SVM, LSTM e GRU

Fonte: Elaborado pelo Autor (2021).

5 METODOLOGIA

Neste capítulo, serão descritos os procedimentos metodológicos realizados para o desenvolvimento de modelos de aprendizado de máquina aplicáveis ao monitoramento de estruturas da construção civil. Na seção 5.1, será abordado o estágio de aquisição dos dados. Já na seção 5.2, será apresentada a reformulação das séries temporais, de forma a serem utilizadas como entrada para os algoritmos de AM. Na seção 5.3, por sua vez, será detalhado o procedimento de treinamento dos modelos. Por fim, a seção 5.4 abordará o processo de avaliação dos resultados obtidos com os modelos criados.

5.1 Aquisição dos dados

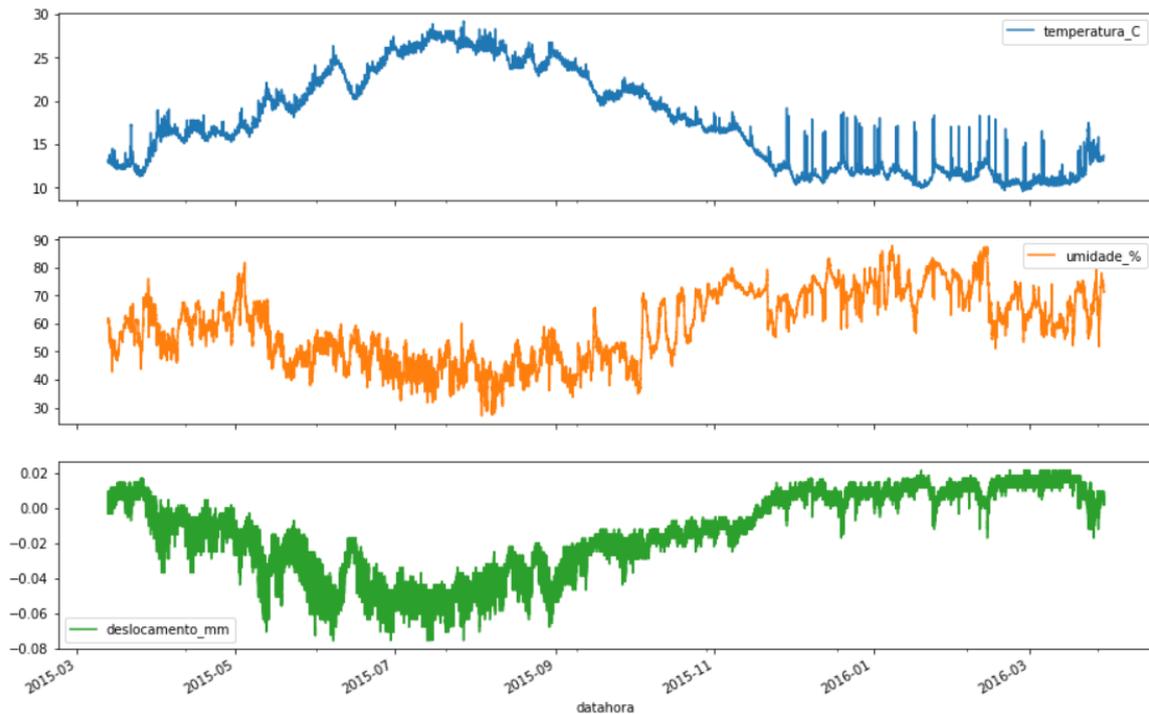
A aquisição dos dados consiste em uma das principais etapas do processo de desenvolvimento de modelos de aprendizado de máquina, pois é nesta fase que será feita a coleta dos dados que serão utilizados como entrada pelos algoritmos para a criação e avaliação dos modelos. Dessa forma, com o propósito de exemplificar melhor como ocorre o processo de coleta de dados para o problema tratado neste trabalho, retoma-se ao quadrado tracejado rosa da Figura 1, que ilustra a obtenção dos dados através de uma rede de sensores sem fio instalados na estrutura de uma ponte. Esses sensores são responsáveis pela aquisição periódica dos atributos estabelecidos no planejamento do sistema de monitoramento daquela estrutura.

Para o desenvolvimento deste trabalho de conclusão de curso, serão utilizados para o treinamento dos modelos, os dados coletados pelo trabalho desenvolvido em Mesquita *et al.* (2018) no monitoramento de saúde estrutural da Igreja de Vila Nova de Foz Côa, em Portugal. O período de coleta corresponde a 13 de março de 2015 a 29 de março de 2016, com a periodicidade de 1 minuto, totalizando 553072 entradas. Para esta pesquisa, serão considerados apenas os dados coletados pelo sensor de temperatura T1, sensor de umidade relativa HR1 e fissurômetro P1, vide Figura 8. Os sensores T1, HR1 e P1 correspondem, respectivamente, à temperatura em graus Celsius ($^{\circ}\text{C}$) no interior da igreja, à umidade relativa do ar (%) e ao deslocamento relativo (mm) na base de uma coluna danificada na nave central da igreja.

A Figura 9 exibe, de forma gráfica, os dados obtidos pelos três sensores anteriormente descritos, em que o eixo horizontal retrata a decorrência do tempo para os três atributos e o eixo vertical os valores absolutos para cada atributo individualmente. A linha azul representa as variações de temperatura, a linha laranja são os valores de umidade do ar e a linha verde os

dados de deslocamento relativo.

Figura 9 – Representação gráfica dos dados de temperatura, umidade do ar e deslocamento do monitoramento da igreja de Foz Côa



Fonte: Elaborado pelo Autor (2020).

Vale ressaltar que, embora os dados aqui utilizados para o treinamento dos modelos limite-se apenas ao monitoramento de uma igreja, mantém-se os objetivos do estudo para qualquer estrutura da construção civil, uma vez que o mesmo pode facilmente ser adaptado a novos conjuntos de dados. O elevado investimento financeiro, a complexidade e o tempo necessário despendido para a aquisição das informações por meio de um sistema de monitoramento de saúde estrutural embasaram a escolha de utilização dos dados do monitoramento da igreja de Foz Côa.

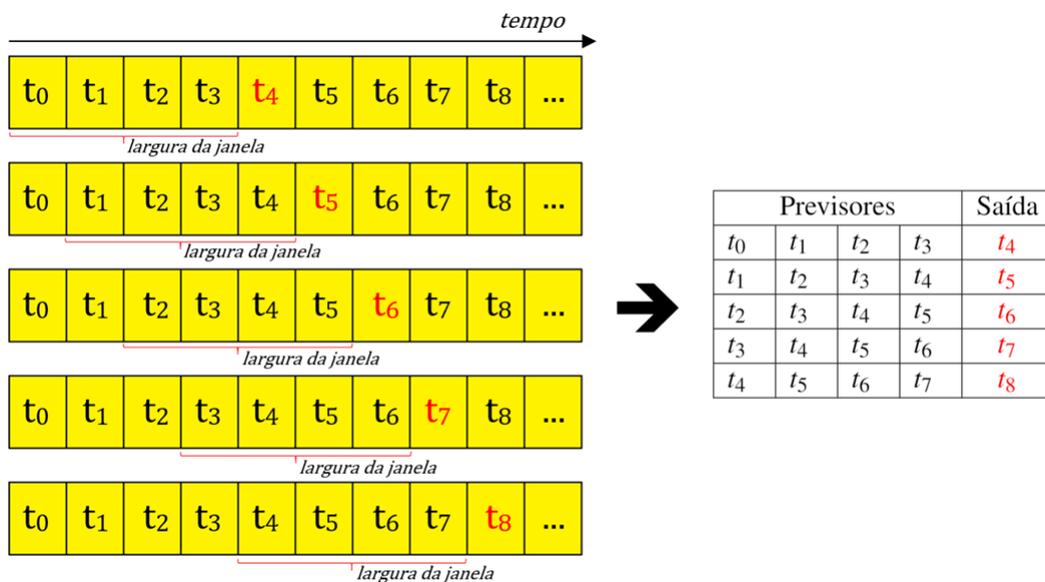
5.2 Preparação dos dados

Devido à natureza sequencial das observações, caracterizada como uma série temporal, é necessária a utilização de uma técnica que transforme as entradas em um formato utilizável por algoritmos de aprendizagem supervisionada, além da necessária manutenção da ordem de coleta dos dados. Dietterich (2002) faz uma revisão das principais técnicas que lidam com dados sequenciais no aprendizado de máquina, entre elas, o método da janela deslizante.

Neste trabalho, realizou-se a reestruturação dos dados utilizando o método da janela

deslizante. Isso se justifica pela necessidade de fragmentar a série temporal em fatias menores, definindo um número de etapas do tempo usados como atributo predictor e a quantidade de valores a serem previstos, atributo meta. A Figura 10 ilustra o processo de reestruturação dos dados através do método anteriormente descrito. Assim, a largura da janela corresponde à quantidade de observações anteriores usadas como entrada para a previsão da próxima etapa do tempo, destacada pelos valores em vermelho na referida imagem. A tabela da direita representa a organização dos dados após a aplicação do método da janela deslizante, em que as etapas anteriores de tempo servirão como predictores para a saída.

Figura 10 – Representação gráfica do funcionamento do método da janela deslizante



Fonte: Elaborado pelo Autor (2021).

A próxima etapa de preparação dos dados se deu pela normalização das entradas dentro do intervalo de 0 e 1, calculado por meio da seguinte fórmula:

$$z = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (5.1)$$

Onde $\min(x)$ e $\max(x)$ correspondem, respectivamente, ao menor e ao maior valor de uma determinada variável. Com esse processo, busca-se a transformação de todas as entradas para a mesma ordem de grandeza, sendo transformada a entrada de menor valor de uma determinada coluna da tabela no número 0, o maior valor em 1 e todos os demais pertencentes a esse intervalo.

O processo de normalização dos dados é de fundamental importância no treinamento de modelos de AM. Dados em escalas diferentes podem comprometer o desempenho do treinamento, uma vez que são feitas operações matemáticas sobre esses e que podem elevar os custos

de processamento e aumentar o tempo de ajuste em dados que não estejam normalizados. Para uma melhor compreensão, Singh e Singh (2020) investigam os impactos da normalização dos dados em problemas de classificação.

5.3 Treinamento dos modelos

O processo de treinamento dos modelos se deu por intermédio da utilização dos recursos disponibilizados pela biblioteca de aprendizado de máquina *Scikit-learning* e pela biblioteca de aprendizado profundo *Keras*. Assim, utilizaram-se os principais algoritmos de aprendizado supervisionado recomendados para a previsão de séries temporais. Em um processo gradual, treinaram-se modelos de teste utilizando algoritmos de mais simples implementação como floresta aleatória, até as mais robustas redes neurais recorrentes, como LSTM e GRU.

Mediante o método da janela deslizante descrito na seção anterior, definiu-se a largura da janela com o valor 15. Isso representa que para cada entrada de valor do atributo deslocamento a ser prevista, serão utilizadas 15 entradas anteriores do atributo de temperatura e umidade relativa do ar. Esse valor é resultante de diversos testes realizados com diferentes algoritmos, uma vez que apresentou as melhores métricas de desempenho.

A definição dos parâmetros de treinamento dos modelos deu-se por diferentes combinações. Por meio de modelos de testes construídos com os algoritmos de aprendizado de máquina, observaram-se os parâmetros que melhor representaram consideráveis acréscimos nos valores das métricas de desempenho.

5.4 Método de avaliação

As métricas de erro constituem uma forma hábil de avaliação de desempenho de modelos de AM. Recorrendo à análise entre os valores reais e os previstos por um modelo treinado, estimou-se sua performance quando submetido a novas entradas e comparado o desempenho entre diferentes abordagens. Dessa forma, este trabalho utilizou como indicadores de desempenho dos modelos criados o *Mean Absolute Error* (MAE) e o Coeficiente de Determinação (R^2), também conhecido como R quadrado.

Como o próprio nome sugere, o MAE calcula o erro médio absoluto entre os valores reais e os previstos. Ela retorna um valor não negativo, que quanto mais próximo de 0 indica que os valores previstos encontram-se bem próximos dos reais. O cálculo do MAE é realizado com a

Fórmula 5.2, em que y_i é o valor da amostra e \hat{y}_i é o valor previsto.

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (5.2)$$

O valor do coeficiente de determinação é obtido com a aplicação da Fórmula 5.4, em que na parte superior da fração calcula-se a soma residual dos erros quadrados, com y_i representando o valor real da amostra e \hat{y}_i o valor previsto pelo modelo ajustado. Já na parte inferior, tem-se a soma dos quadrados da diferença entre as amostras e o valor médio das amostras, representado por $\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$. Essa fórmula retorna valores entre 0 e 1, quanto mais próximo de 1 melhor o modelo conseguiu prever novas entradas em relação aos dados originais, e quanto mais próximo de 0 evidencia uma baixa capacidade preditiva do modelo.

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (5.3)$$

Através dos valores métricos encontrados com a avaliação de modelos, chega-se a conclusões importantes. Por exemplo, qual se adaptou melhor aos dados de treinamento, qual conseguiu realizar as melhores previsões na base de teste ou qual apresentou o maior erro médio absoluto. Complementarmente, a análise gráfica é de suma importância nessa avaliação, pois fornece de forma visual a capacidade de interpretar o desempenho de modelos e identificar tendências e a dispersão entre as entradas reais e as previsões. Dessa forma, determina-se o modelo mais adequado ao problema para usá-lo como auxílio na tomada de decisões de reparos em estruturas da construção civil.

6 RESULTADOS E DISCUSSÕES

Nesta seção, serão apresentados os resultados obtidos com o desenvolvimento dos modelos de aprendizado de máquina para o monitoramento de estruturas, utilizando os dados do trabalho proposto em Mesquita *et al.* (2018), no monitoramento da Igreja de Foz Côa, Portugal. Para o desenvolvimento das etapas de criação dos modelos de aprendizado de máquina, foi utilizado como recurso base o *Google Colab*, uma poderosa ferramenta para a escrita e execução de código *Python* utilizando infraestrutura em nuvem. As especificações técnicas dos principais componentes de hardware disponibilizadas para uso por meio do ambiente da ferramenta são sintetizadas na Tabela 2.

Tabela 2 – Especificações técnicas de hardware do *Google Colab*

CPU	GPU	Memória RAM	Espaço em Disco
2x Intel(R) Xeon(R) @ 2.20GHz	Nvidia Tesla T4 16 GB	12 GB	25 GB

Fonte: Elaborado pelo Autor (2021).

Além de toda capacidade de processamento disponibilizado pelo *Google Colab*, esse ainda dispõe de diversas bibliotecas de processamento de dados e aprendizado de máquina necessárias para a criação de modelos, já instaladas em seu ambiente. Dentre essas bibliotecas disponíveis e que foram utilizadas na elaboração deste trabalho, destacam-se *Pandas* e *Numpy* para preparação dos dados; *Matplotlib*, para a criação de gráficos; *Scikit-learn*, para a criação de modelos de aprendizado de máquina e utilização de métricas de desempenho e a biblioteca *Keras*, para a criação dos modelos de aprendizado profundo.

Todos os modelos avaliados a seguir treinaram-se a partir do uso de 428099 entradas de dados e reestruturaram-se com o método da janela deslizante, utilizando o tamanho da janela igual a 15. Ademais, todos foram avaliados usando a proporção de 80% dos registros para treinamento e 20% para testes, divisão efetuada após a reformulação do dataset. Nas seções seguintes, serão tratados os detalhes de implementação de cada modelo, bem como os resultados alcançados com a utilização de cada um no auxílio do processo de monitoramento de estruturas da construção civil.

6.1 Resultados Floresta Aleatória

Implementou-se o modelo criado a partir do algoritmo floresta aleatória com o uso da classe *RandomForestRegressor* da biblioteca *Scikit-learn*. O principal parâmetro de treinamento definido foi o número de estimadores, que pode ser entendido como o número de árvores de decisão na "floresta".

A Tabela 3 mostra os resultados obtidos com os modelos criados, levando-se em consideração o número de estimadores, as métricas MAE e R^2 na base de treinamento e de testes, assim como o tempo despendido para o treinamento. Observa-se uma expressiva similaridade entre as métricas de desempenho, não existindo uma significativa tendência de crescimento conforme o número de estimadores aumenta. Além disso, percebeu-se a existência de um sobre ajuste dos modelos aos dados de treinamento em relação aos de teste, conforme apontam os coeficientes de determinação calculados em ambos.

Tabela 3 – Métricas de desempenho do algoritmo Floresta Aleatória

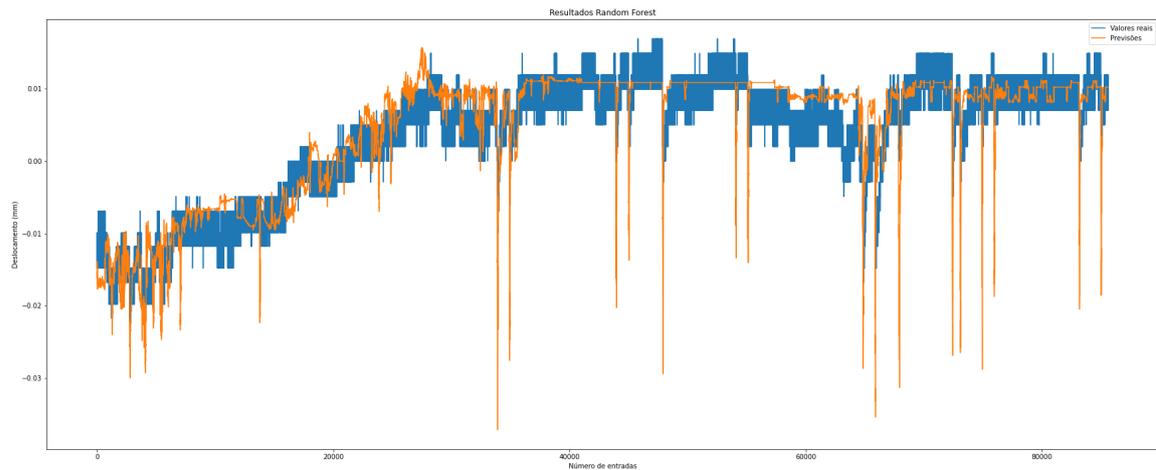
Número estimadores	MAE treinamento	MAE teste	R^2 treinamento	R^2 teste	Tempo (min)
10	0.00084	0.00303	0.994	0.742	1.7
50	0.00076	0.00292	0.996	0.756	8.09
100	0.00075	0.00292	0.996	0.756	16.03
250	0.00074	0.00293	0.996	0.757	40.06

Fonte: Elaborado pelo Autor (2021).

Quando se tratou do tempo necessário para treinamento de cada modelo, notou-se um acréscimo considerável à medida que o número de estimadores também aumentou. Por outro lado, não há um acompanhamento substancial das demais métricas a esses valores. A partir disso, o modelo ajustado com 50 estimadores destacou-se, sobretudo, devido ao tempo de treinamento inferior, dada a similaridade das demais métricas.

A Figura 11 demonstra, de forma gráfica, as previsões realizadas com o modelo de 50 árvores na base de dados de teste, em que a linha em azul representa os valores reais e a linha laranja os valores que o modelo conseguiu prever. Nesse viés, observou-se que ele conseguiu capturar a tendência dos dados. Não obstante, teve dificuldades em capturar os valores picos de deslocamento.

Figura 11 – Previsões do modelo criado com o algoritmo Floresta Aleatória



Fonte: Elaborado pelo Autor (2021).

6.2 Resultados Máquina de Vetores de Suporte

Para a criação do modelo a partir do algoritmo SVM utilizou-se a classe *SVR* da biblioteca *Scikit-learn*, uma versão especializada para problemas de regressão. Para isso, foram testados três diferentes *kernels*, a fim de obter aquele que melhor se adaptava aos dados, e por conseguinte, realizava as melhores previsões. São eles: Função de Base Radial ou *Radial Basis Function* (RBF), Polinomial ou *Poly* e Sigmoide.

Em decorrência da restrição de hardware e da ultrapassagem do limite de tempo de uso fornecido pelo *Google Colab*, foram encontradas dificuldades para o treinamento de modelos com os *kernels poly* e sigmoide. Isso significa que não foi possível obter resultados conclusivos, optando-se por analisar apenas os resultados do *kernel* RBF.

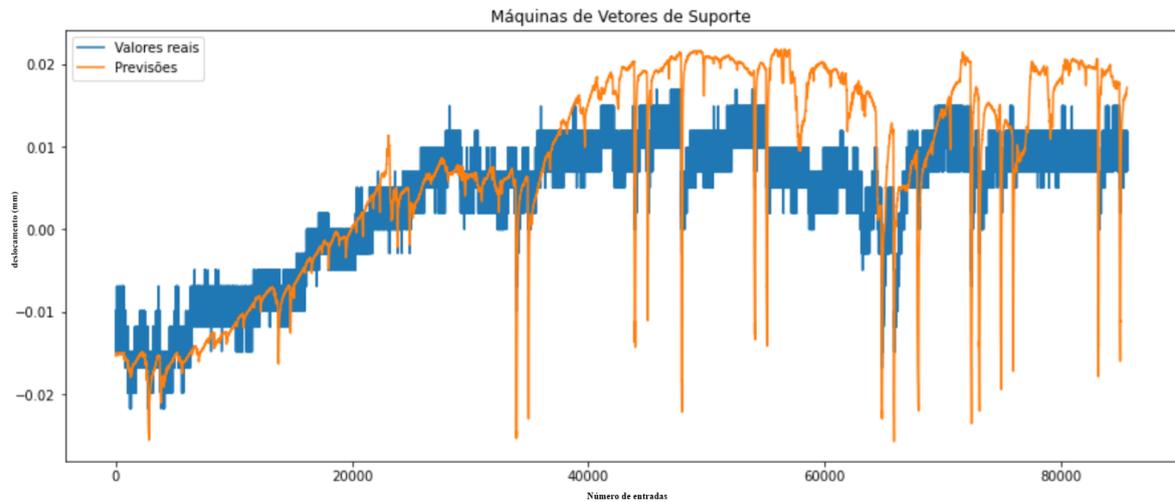
Assim, na Tabela 4 estão dispostas as métricas alcançadas com o modelo criado com o algoritmo. No geral, nota-se um considerável ajuste do modelo aos dados de treinamento, o que não é observado nas previsões utilizando os dados de teste, com apenas 0.224 na métrica R^2 . O gráfico da Figura 12 evidencia a baixa capacidade preditiva do modelo criado a partir do *kernel* RBF. Nessa esteira, verifica-se uma certa captura de tendência, porém uma alta discrepância entre os valores reais e os previstos, intensificada a partir dos 40000 registros.

Tabela 4 – Métricas de desempenho do algoritmo SVM

<i>Kernel</i>	MAE treinamento	MAE teste	R^2 treinamento	R^2 teste	Tempo (m)
RBF	0.00574	0.00569	0.852	0.224	38.68

Fonte: Elaborado pelo Autor (2021).

Figura 12 – Previsões do modelo criado com o algoritmo Máquina de Vetores de Suporte



Fonte: Elaborado pelo Autor (2021).

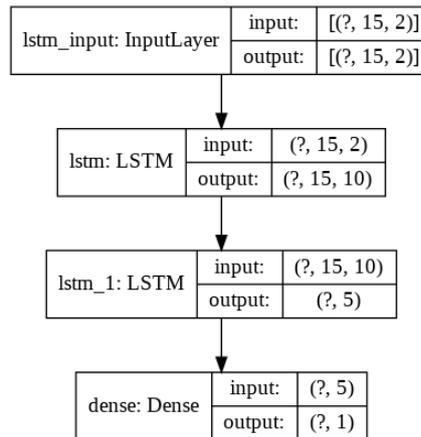
6.3 Resultados LSTM

Nesta seção, serão apresentados os resultados obtidos com o algoritmo LSTM treinado, a usar duas abordagens diferentes. Na primeira, criou-se o modelo de forma tradicional e direta, como descrito na seção 3.5. No segundo método, utilizou-se uma variação denominada de LSTM Bidirecional. Esta consiste na criação de duas redes LSTM paralelas, uma recebendo as entradas de forma direta e a outra na ordem inversa. Desta forma, possibilita-se que as redes mantenham informações tanto do passado como do futuro.

Para o desenvolvimento dos modelos, fez-se uso da biblioteca de aprendizado profundo *Keras*. No método tradicional, duas camadas ocultas do tipo empilhada com 10 e 5 células LSTM, respectivamente, e uma camada densa de saída com a função de ativação *sigmoid* foram usadas. Já na abordagem bidirecional foram empregadas duas camadas ocultas com 32 e 16 unidades LSTM bidirecionais. Acrescido a isso, para o ajuste dos modelos, considerou-se o otimizador *Adam*, a função de perda erro quadrático médio, utilizando 100 épocas de treinamento e tamanho do lote 32. As Figuras 13 e 14 representam, de forma gráfica, a configuração das camadas no treinamento do modelo LSTM tradicional e no bidirecional, na sequência.

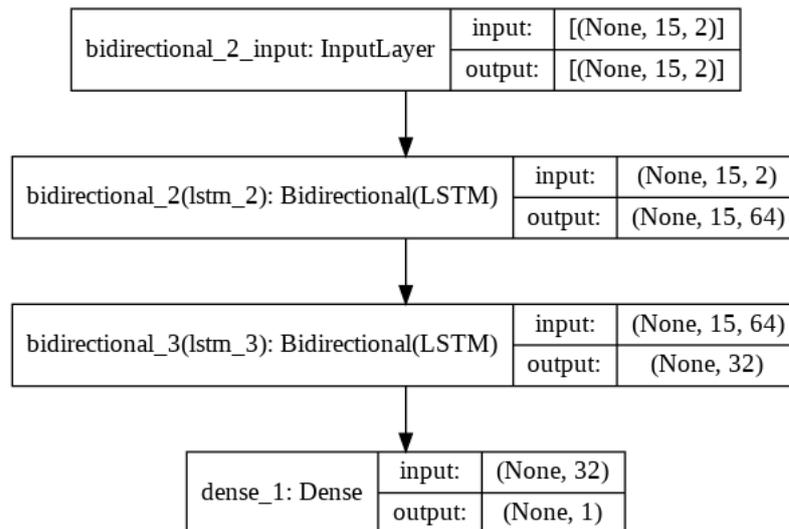
A definição dos parâmetros utilizados nesta etapa deve-se a sucessivos testes realizados com diferentes configurações de treinamento. Assim, verificou-se que os modelos conseguiram adaptar-se melhor aos dados usando um número mais reduzido de camadas e unidades LSTM. Quando treinados com uma maior quantidade de camadas ocultas e células de memória, observou-se um ganho expressivo no tempo de treinamento, bem como redução no

Figura 13 – Camadas ocultas do modelo LSTM



Fonte: Elaborado pelo Autor (2021).

Figura 14 – Camadas ocultas do modelo LSTM Bidirecional



Fonte: Elaborado pelo Autor (2021).

valor das métricas de desempenho.

Na Tabela 5, observa-se a performance obtida com as previsões realizadas pelos modelos na base de dados de treinamento e de teste a partir das métricas de desempenho. Ao se comparar os resultados obtidos com os do algoritmo floresta aleatória, observou-se um aumento na métrica MAE na base de treinamento, evidenciando um não sobre ajuste como no último, o que reflete também na redução do coeficiente de determinação na base de treinamento. Já na base de teste, nota-se um aumento significativo no coeficiente de determinação na abordagem tradicional, o que pode refletir em uma capacidade de previsão superior quando novas entradas forem submetidas ao modelo.

Apesar de a abordagem bidirecional adicionar uma segunda rede LSTM, tornando possível armazenar na memória observações do passado e do futuro, ela não apresentou desempe-

Tabela 5 – Métricas de desempenho do algoritmo LSTM

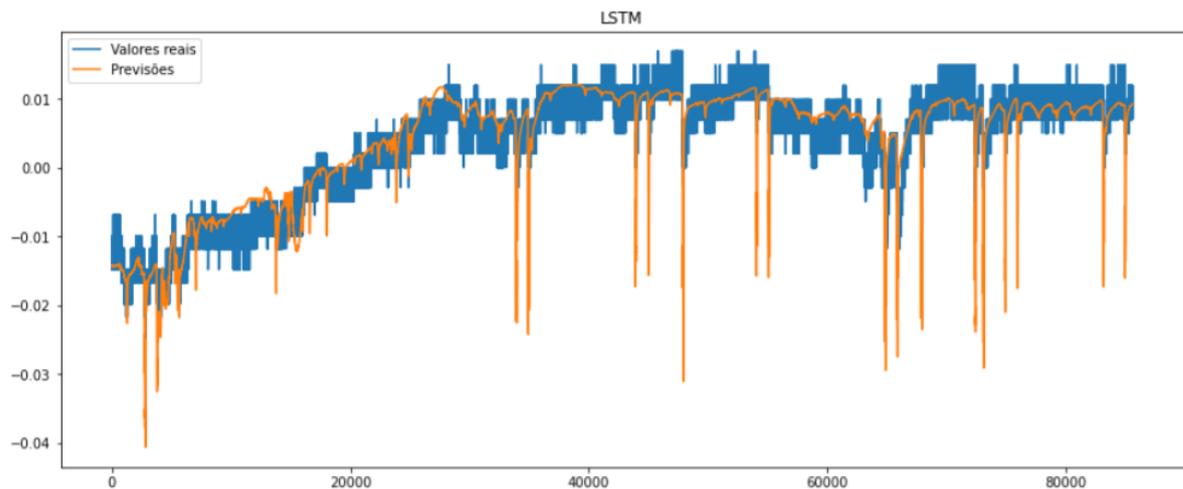
Abordagem	MAE treinamento	MAE teste	R^2 treinamento	R^2 teste	Tempo (m)
Tradicional	0.00432	0.00246	0.893	0.820	110.7
Bidirecional	0.00631	0.00310	0.790	0.774	338.3

Fonte: Elaborado pelo Autor (2021).

no superior ao método tradicional, como evidenciam as métricas de desempenho. É importante ressaltar também o custo de tempo necessário para o treinamento do modelo, que devido a sua rede adicional, necessitou de aproximadamente três vezes mais tempo para a execução do ajuste.

O gráfico da Figura 15 possibilita excelente percepção da capacidade preditiva do modelo tradicional. Desse modo, ele consegue capturar bem a tendência dos dados e apresenta uma menor variação nas previsões em relação ao modelo criado com o algoritmo floresta aleatória. Além disso, a previsão dos dados de pico apresentou notável performance perante ao da última subseção.

Figura 15 – Previsões do modelo criado com o LSTM



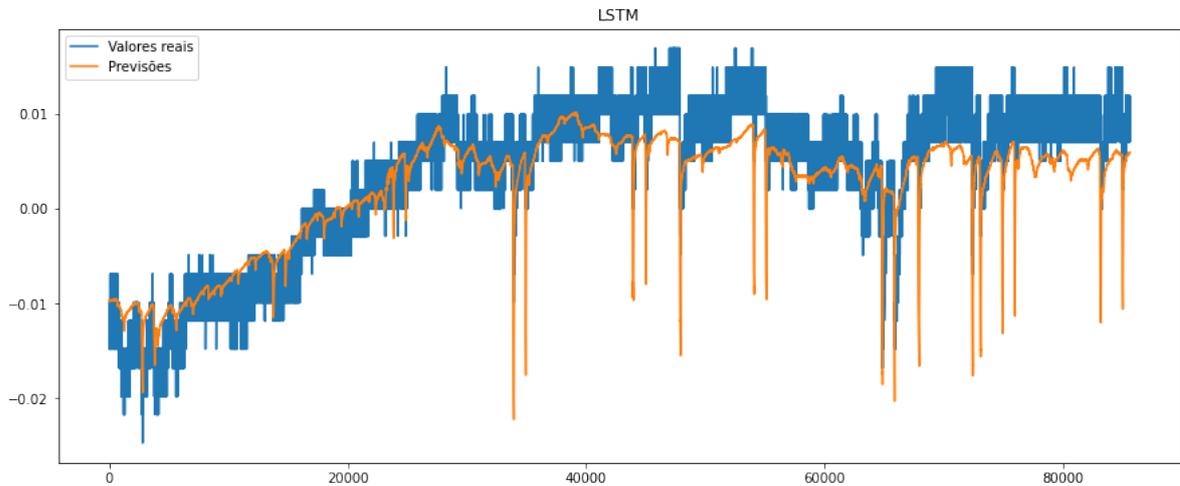
Fonte: Elaborado pelo Autor (2021).

Em contrapartida, no gráfico da Figura 16 percebe-se a inferior capacidade preditiva do modelo bidirecional, indicada pelas previsões abaixo dos valores reais, acentuadas principalmente a partir dos 40000 registros de teste.

6.4 Resultados GRU

Após o desenvolvimento do modelo LSTM e a comprovação de sua performance, decidiu-se pela realização de um novo experimento com redes neurais recorrentes. Assim,

Figura 16 – Previsões do modelo criado com o LSTM Bidirecional

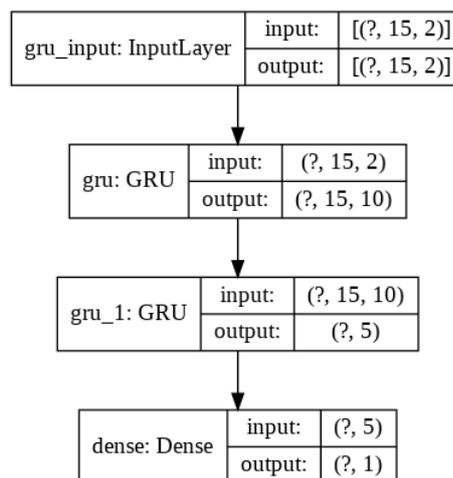


Fonte: Elaborado pelo Autor (2021).

utilizou-se o algoritmo GRU com configurações semelhantes ao LSTM para analisar se haveria algum ganho de desempenho em sua capacidade preditiva.

Novamente, fez-se uso da biblioteca *Keras* e os parâmetros de treinamento para o ajuste dos modelos foram o otimizador Adam, a função de perda Erro Quadrático Médio, utilizando 100 épocas de treinamento e tamanho do lote 32. Contou-se com duas camadas ocultas com 10 e 5 células do tipo GRU, como mostra a Figura 17.

Figura 17 – Camadas ocultas do modelo GRU



Fonte: Elaborado pelo Autor (2021).

Conforme se observa na Tabela 6, não houve ganhos reais de desempenho na capacidade preditiva deste modelo em relação ao LSTM. Todavia, as métricas de desempenho expressam valores significativamente melhores que dos modelos floresta aleatória, máquina de vetores de suporte e que a abordagem bidirecional do LSTM. Por meio da análise gráfica da

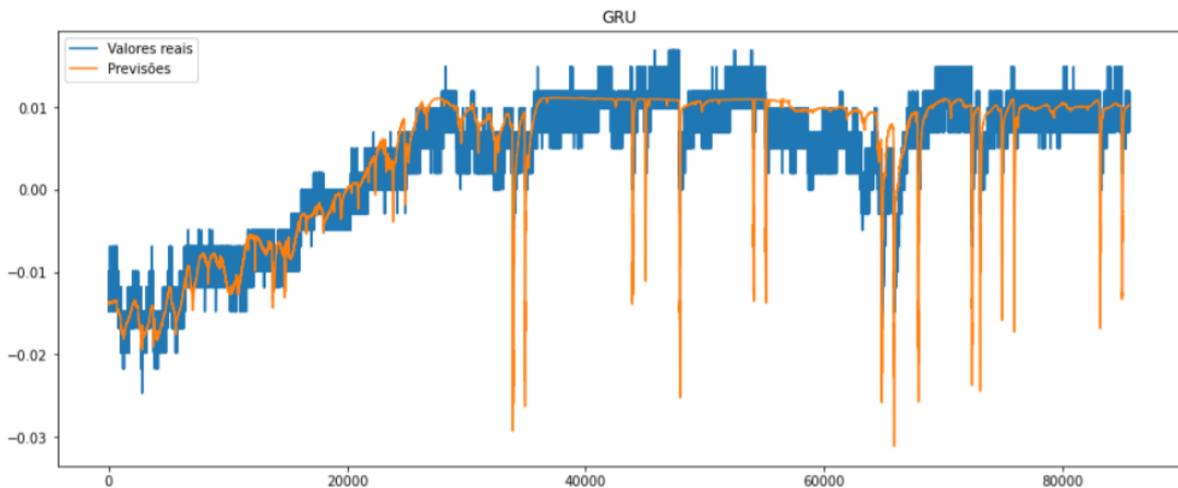
Figura 18, observam-se menores oscilações evidenciadas principalmente nas primeiras 20000 entradas.

Tabela 6 – Métricas de desempenho do algoritmo GRU

MAE treinamento	MAE teste	R^2 treinamento	R^2 teste	Tempo (m)
0.00440	0.00247	0.892	0.815	75.2

Fonte: Elaborado pelo Autor (2021).

Figura 18 – Previsões do modelo criado com o GRU



Fonte: Elaborado pelo Autor (2021).

6.5 Conclusão

Os resultados demonstrados neste capítulo foram consequentes de diversos experimentos realizados com diferentes abordagens e metodologias de tratamento de dados e aprendizado de máquina. Apesar das limitações de *hardware* para testes mais aprimorados de otimização de parâmetros, avaliaram-se diferentes métodos até se alcançarem os presentes resultados.

Como exemplo, tentou-se realizar redução de dimensionalidade com a Análise de Componentes Principais (PCA) e a suavização dos dados recorrendo ao cálculo de médias móveis, a fim de obter um melhor ajuste dos modelos aos dados de treinamento. Isso se deu ao observar-se o elevado nível de variação nos valores coletados. Contudo, a aplicação das técnicas não garantiu um expressivo aumento nas métricas de desempenho.

Outrossim, buscou-se dividir as entradas conforme as datas das quatro estações do ano na cidade Foz Côa, em Portugal, para prever cada estação individualmente e analisar seus

resultados. A estratégia mostrou-se falha ao analisar-se as métricas obtidas com a previsão dos valores nas estações verão e inverno, devido aos baixos valores das métricas e evidenciada por meio da análise gráfica. As previsões realizadas nas estações primavera e outono mostraram-se razoáveis, ainda assim, não suficientes para se destacarem diante as demais.

Nenhuma dessas abordagens surtiu efeito significativo no aprimoramento da capacidade preditiva dos modelos, optando-se pela não utilização delas. O curto período de coleta das informações sobressai como possível limitante para a reduzida capacidade preditiva dos modelos de previsão das estações do ano, bem como a baixa eficiência na previsão em várias etapas.

Dessa forma, a Tabela 7 resume o desempenho dos principais modelos avaliados neste trabalho por ordem crescente de desempenho, levando-se em consideração principalmente as métricas referentes ao conjunto de dados de teste. Destacam-se, portanto, os algoritmos de redes neurais recorrentes LSTM e GRU.

Tabela 7 – Quadro resumo das métricas de desempenho dos modelos avaliados

Algoritmo	MAE treinamento	MAE teste	R^2 treinamento	R^2 teste	Tempo (min)
SVM	0.00574	0.00569	0.852	0.224	38.68
Floresta aleatória	0.00076	0.00292	0.996	0.756	8.09
GRU	0.00440	0.00247	0.892	0.815	75.2
LSTM	0.00432	0.00246	0.893	0.820	110.7

Fonte: Elaborado pelo Autor (2021).

7 CONCLUSÕES E TRABALHOS FUTUROS

De forma concisa, apresentou-se neste trabalho do que se trata o monitoramento de saúde estrutural, demonstrando como o uso de modelos de aprendizado de máquina podem auxiliar na tarefa de caracterização do deslocamento relativo em estruturas da construção civil que possuem algum tipo de comprometimento estrutural.

Sob esse ângulo, compararam-se algumas abordagens para a utilização de algoritmos de aprendizado de máquina para a previsão de séries temporais, provenientes de sensores instalados em estruturas de engenharia. Dessa forma, os modelos desenvolvidos no decorrer do estudo buscaram comparar os principais algoritmos utilizados quanto à adaptação aos dados obtidos com o monitoramento da Igreja de Vila Nova de Foz Côa, em Portugal.

Por intermédio da análise gráfica e das métricas de desempenho obtidas com a submissão dos dados para o treinamento e validação dos modelos, observou-se considerável adaptação aos dados de treinamento, bem como notável capacidade de previsão de novas entradas por meio dos dados de teste. Nesse sentido, fica clara a capacidade de previsão dos modelos, dando ênfase aos algoritmos de redes neurais recorrentes, GRU e LSTM.

Não se pode deixar de mencionar, indubitavelmente, que se percebeu a capacidade de adaptação dos modelos aos dados utilizados, apesar da limitação quanto à quantidade de observações coletadas no monitoramento da Igreja de Foz Côa. Devido ao período de coleta, as variações de temperatura e umidade do ar e a influência desses nos deslocamentos relativos da estrutura, refletiram observações que capturaram cada estação do ano apenas uma vez.

De outro modo, isso significa que, em virtude da necessidade de preservação da ordem cronológica das entradas, o modelo utiliza para treinamento os dados coletados durante a primavera, o verão e o outono. Outrossim, tenta realizar previsões na base de teste, composta essencialmente por dados coletados durante o inverno, em que não se submeteram aos modelos anteriormente.

Portanto, a partir das limitações supracitadas, propõe-se como trabalho futuro a obtenção de uma nova amostra de dados mediante o monitoramento de saúde estrutural de alguma estrutura da construção civil e que possua o período de observação de, no mínimo dois anos, suficiente para capturar as estações do ano pelo menos duas vezes. Ademais, recomenda-se o estudo de outros modelos criados com redes neurais convolucionais, *Seq2seq* ou ARIMA, a utilizar uma abordagem focada na otimização dos parâmetros de treinamento.

REFERÊNCIAS

- AYODELE, T. O. Types of machine learning algorithms. **New advances in machine learning**, InTech, v. 3, p. 19–48, 2010.
- BREIMAN, L. Random forests. **Machine learning**, Springer, v. 45, n. 1, p. 5–32, 2001.
- CHO, K.; MERRIËNBOER, B. V.; GULCEHRE, C.; BAHDANAU, D.; BOUGARES, F.; SCHWENK, H.; BENGIO, Y. Learning phrase representations using rnn encoder-decoder for statistical machine translation. **arXiv preprint arXiv:1406.1078**, 2014.
- CISLO, P. **Regression vs Classification**. 2019. Acesso em: 22 de out. de 2019. Disponível em: <<https://i.pinimg.com/originals/71/8e/6a/718e6a40e1782bead960e58d3c52663b.png>>.
- COMISU, C.-C.; TARANU, N.; BOACA, G.; SCUTARU, M.-C. Structural health monitoring system of bridges. **Procedia engineering**, Elsevier, v. 199, p. 2054–2059, 2017.
- CORTES, C.; VAPNIK, V. Support-vector networks. **Machine learning**, Springer, v. 20, n. 3, p. 273–297, 1995.
- DIETTERICH, T. Overfitting and undercomputing in machine learning. **ACM computing surveys**, Citeseer, v. 27, n. 3, p. 326–327, 1995.
- DIETTERICH, T. G. Machine learning for sequential data: A review. In: SPRINGER. **Joint IAPR international workshops on statistical techniques in pattern recognition (SPR) and structural and syntactic pattern recognition (SSPR)**. [S.l.], 2002. p. 15–30.
- DRAKOS, G. **Support Vector Machine vs Logistic Regression**. 2018. Acesso em: 15 de out. de 2019. Disponível em: <<https://towardsdatascience.com/support-vector-machine-vs-logistic-regression-94cc2975433f>>.
- FARRAR, C. R.; WORDEN, K. An introduction to structural health monitoring. **Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences**, The Royal Society London, v. 365, n. 1851, p. 303–315, 2006.
- FENG, C.; LIU, M.-Y.; KAO, C.-C.; LEE, T.-Y. Deep active learning for civil infrastructure defect detection and classification. In: **Computing in Civil Engineering 2017**. [S.l.: s.n.], 2017. p. 298–306.
- GOMES, P. 1 em 5 pontes ou viadutos precisa de reforma; veja regras para manutenção. **Folha de S.Paulo**, 2018. São Paulo, ano 98, n. 32.656, 30 ago. 2018. Cotidiano, p. B4.
- HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. **Neural computation**, MIT Press, v. 9, n. 8, p. 1735–1780, 1997.
- LI, J.; DENG, J.; XIE, W. Damage detection with streamlined structural health monitoring data. **Sensors**, Multidisciplinary Digital Publishing Institute, v. 15, n. 4, p. 8832–8851, 2015.
- LORENZETT, C. D. C.; TELÖCKEN, A. V. Estudo comparativo entre os algoritmos de mineração de dados random forest e j48 na tomada de decisão. **Simpósio de Pesquisa e Desenvolvimento em Computação (SPDC)**, v. 2, n. 1, 2016.
- MARSLAND, S. **Machine learning: an algorithmic perspective**. [S.l.]: Chapman and Hall/CRC, 2014.

MESQUITA, E.; ARÊDE, A.; PINTO, N.; ANTUNES, P.; VARUM, H. Long-term monitoring of a damaged historic structure using a wireless sensor network. **Engineering Structures**, Elsevier, v. 161, p. 108–117, 2018.

MOHRI, M.; ROSTAMIZADEH, A.; TALWALKAR, A. **Foundations of machine learning**. [S.l.: s.n.], 2018.

NEVES, A.; GONZÁLEZ, I.; LEANDER, J.; KAROUMI, R. Structural health monitoring of bridges: a model-free ann-based approach to damage detection. **Journal of Civil Structural Health Monitoring**, Springer, v. 7, n. 5, p. 689–702, 2017.

OLAH, C. **Understanding lstm networks**. 2015. Acesso em: 23 de fev. de 2021. Disponível em: <<http://colah.github.io/posts/2015-08-Understanding-LSTMs/>>.

PHI, M. **Illustrated Guide to LSTM's and GRU's: A step by step explanation**. 2018. Acesso em: 11 de fev. de 2021. Disponível em: <<https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21>>.

RASCHKA, S. **Python machine learning**. [S.l.]: Packt Publishing Ltd, 2015.

SINGH, D.; SINGH, B. Investigating the impact of data normalization on classification performance. **Applied Soft Computing**, Elsevier, v. 97, p. 105524, 2020.

SMARSLY, K.; DRAGOS, K.; WIGGENBROCK, J. Machine learning techniques for structural health monitoring. In: **Proceedings of the 8th European Workshop on Structural Health Monitoring (EWSHM 2016), Bilbao, Spain**. [S.l.: s.n.], 2016. p. 5–8.