UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE TECNOLOGIA E GEOCIÊNCIAS
DEPARTAMENTO DE ENGENHARIA CIVIL
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA CIVIL

BRUNA BRITO LIBERAL

APRENDIZADO DE MÁQUINA SUPERVISIONADO PARA CLASSIFICAÇÃO DE NUVENS DE PONTOS DE GRANDES ÁREAS

Recife
2024

BRUNA BRITO LIBERAL

APRENDIZADO DE MÁQUINA SUPERVISIONADO PARA CLASSIFICAÇÃO DE NUVENS DE PONTOS DE GRANDES ÁREAS

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

Área de concentração: Construção Civil.

Orientador: Prof. Dra. Rachel Perez Palha.

Recife

2024

.Catalogação de Publicação na Fonte. UFPE - Biblioteca Central

Liberal, Bruna Brito.
Aprendizado de máquina supervisionado para classificação de nuvens de pontos de grandes áreas / Bruna Brito Liberal. - Recife, 2024.
70f.: il.

Dissertação (Mestrado), Universidade Federal de Pernambuco, Centro de Tecnologia e Geociências, Programa de Pós-Graduação em Engenharia Civil, 2024.
Orientação: Rachel Perez Palha.

1. Nuvens de pontos; 2. BIM; 3. Classificação; 4. Aprendizado de máquina; 5. Random Forest. I. Palha, Rachel Perez. II. Título.

UFPE-Biblioteca Central CDD 624

BRUNA BRITO LIBERAL

APRENDIZADO DE MÁQUINA SUPERVISIONADO PARA CLASSIFICAÇÃO DE NUVENS DE PONTOS DE GRANDES ÁREAS

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Civil da Universidade Federal de Pernambuco, Centro de Tecnologia e Geociências, como requisito parcial para a obtenção do título de Mestre em Engenharia Civil. Área de concentração: Construção Civil.

Aprovada em: 05/08/2024.

BANCA EXAMINADORA

_____

Profa. Dra. Rachel Perez Palha
Universidade Federal de Pernambuco

_____

Prof. Dr. Arnaldo Manoel Pereira Carneiro
Universidade Federal de Pernambuco

_____

Prof. Dr. Fabiano Rogério Corrêa
Universidade de São Paulo

_____

Prof. Dr. Conrado de Souza Rodrigues
Centro Federal de Educação Tecnológica de Minas Gerais

Dedico este trabalho às minhas avós, Eunice e Maria, que partiram durante a minha pesquisa e que sempre torceram por um futuro brilhante para mim.

AGRADECIMENTOS

A Deus, por sempre iluminar meus caminhos.

À Professora Dra. Rachel Perez Palha, minha orientadora, por todo conhecimento compartilhado, pela disponibilidade, pelo apoio incondicional e pelo estímulo para superar os desafios que apareceram durante o desenvolvimento da pesquisa.

A Arthur Henrique da Costa e Silva e a Gustavo de Hollanda Cavalcanti Soares, por todas as sugestões e suporte dados ao desenvolvimento deste trabalho.

Aos meus pais, Auxiliadora e Bismarck, pelo amor incondicional e por sempre colocarem meus estudos em primeiro lugar, muitas vezes abdicando de projetos pessoais para investir na minha educação. A certeza de que sempre posso contar com vocês torna a minha caminhada muito mais leve.

A Thales Castro, por todo amor e companheirismo, e por sempre me incentivar a ir cada vez mais longe, desde o ensino médio. A vida é muito mais fácil do teu lado.

À Luma, por sempre tornar leves os momentos de angústia.

À minha avó Maria (*in memoriam*), por sempre ter sido um colo cheio de amor e por todas as orações que sempre me abençoaram fortemente. À minha avó Eunice (*in memoriam)*, por ter sido um exemplo de mulher forte e por todo o carinho disfarçado em forma de toalhinhas bordadas com meu nome. Não esperava concluir essa etapa da minha vida sem vocês, mas tenho certeza de que sempre estarão olhando por mim.

A Roseane Castro, Giullia e Kauã, por toda torcida e apoio.

A Milena Gomes, por todo suporte emocional desde o início do mestrado.

Aos amigos que fiz durante o mestrado, Cláudia, Eduardo, Fernanda, Josivan e Marília, por todo incentivo. Agradeço também aos amigos que sempre me incentivaram desde a graduação, em especial a Álamo, Beatriz, Clarissa, Felipe, Gabriel, Hayla, Marcello, Nattally e Pedro.

A TPF Engenharia e, em especial, a Arthur Silva, Bárbara Vilar, Juliana Scanoni e Priscila Camelo, por todo apoio e suporte para desenvolvimento dessa iniciativa de inovação. Agradeço também a Abmael Júnior, Andresa Dornelas, Dália Katz e Roseane Soares, que me deram suporte nos momentos em que precisei me ausentar para participar das atividades do mestrado.

RESUMO

O uso de tecnologias de obtenção de nuvens de pontos para auxílio na elaboração de modelos BIM é cada vez mais presente na indústria AECO (Arquitetura, Engenharia, Construção e Operação), entretanto, a modelagem a partir de uma nuvem de pontos pode ser muito trabalhosa e pouco intuitiva. Neste trabalho, inicialmente foi realizada uma análise bibliométrica da literatura existente acerca do BIM e da classificação de nuvens de pontos, para uma melhor contextualização dos estudos que estão sendo desenvolvidos nesse âmbito. A partir disso, a pesquisa proposta visa dar um passo inicial na automatização do processo de modelagem, por meio do desenvolvimento de um algoritmo de aprendizado de máquina supervisionado baseado em Random Forest, capaz de classificar nuvens de pontos de grandes áreas a partir da inserção de um conjunto de dados de nuvens pré-classificadas. O algoritmo desenvolvido é capaz de classificar regiões das nuvens de pontos em três categorias principais: terreno, edificações e vegetação, sendo especialmente útil para apoiar a construção de modelos de cidades ou obras de infraestrutura. Os dados utilizados para treinamento do modelo foram obtidos por meio do GeoSampa, uma plataforma de dados abertos da cidade de São Paulo, onde é possível obter imagens aéreas e nuvens de pontos classificadas de todo o município. Para elaboração do algoritmo, foi utilizado o *Visual Studio Code* e, para visualização dos dados das nuvens de pontos, foi utilizado um *software* de visualização de nuvens de pontos, o *CloudCompare*. O estudo mostrou que o algoritmo desenvolvido foi útil para a identificação das classes, pois foram obtidas métricas de desempenho que indicaram que o aprendizado de máquina supervisionado se comportou de maneira satisfatória. A partir dos resultados obtidos foi observado que é necessário atenção com a qualidade dos dados de entrada. A principal contribuição do algoritmo é a possibilidade de reconhecer e separar as nuvens de pontos em diferentes classes, além de ser capaz de classificar nuvens de pontos obtidas por fotogrametria. Isso é especialmente útil para os casos em que se deseja modelar apenas determinados elementos de uma nuvem de pontos extensa, pois o algoritmo permite a separação de suas classes; ou quando é necessário representar determinados elementos no modelo sem sua informação semântica, como um conjunto de árvores.

Palavras-chave: Nuvens de pontos, BIM, Classificação, Aprendizado de Máquina, Random Forest.

ABSTRACT

The use of technologies for obtaining point clouds to aid in the creation of BIM models is increasingly present in the AECO (Architecture, Engineering, Construction and Operation) industry, however, modeling from a point cloud can be very laborious and unintuitive. In this work, initially a bibliometric analysis of the existing literature on BIM and point cloud classification was carried out, for a better contextualization of the studies that are being developed in this field. Based on this, the proposed research aims to take an initial step in automating the modeling process, through the development of a supervised machine learning algorithm based on Random Forest, capable of classifying point clouds over large areas by inserting a dataset of pre-classified clouds. The developed algorithm is capable of classifying point cloud regions into three main categories: terrain, buildings and vegetation, being especially useful for supporting the construction of city models or infrastructure works. The data used to train the model was obtained through GeoSampa, an open data platform in the city of São Paulo, where it is possible to obtain aerial images and classified point clouds from the entire city. To develop the algorithm, Visual Studio Code was used and, to visualize the point cloud data, point cloud visualization software, CloudCompare, was used. The study showed that the developed algorithm was useful for class identification, as performance metrics were obtained that indicated that supervised machine learning behaved satisfactorily. From the results obtained, it was observed that attention is needed with the quality of the input data. The main contribution of the algorithm is the possibility of recognizing and separating point clouds into different classes, in addition to being able to classify point clouds obtained by photogrammetry. This is especially useful for cases where you want to model only certain elements of an extensive point cloud, as the algorithm allows the separation of its classes; or when it is necessary to represent certain elements in the model without their semantic information, such as a set of trees.

Keywords: Point clouds, BIM, Classification, Machine Learning, Random Forest.

LISTA DE FIGURAS

LISTA DE QUADROS

SUMÁRIO

1 INTRODUÇÃO

O interesse em Building Information Modeling (BIM) tem crescido amplamente nos últimos anos, pois esta metodologia contribui, entre outras coisas, para otimizar o ciclo de vida dos projetos e reduzir custos de forma geral (Bensalah et al., 2019). A metodologia BIM envolve diversos fatores que abrangem o processo construtivo desde o planejamento da obra até a pós-construção. Paralelo a isso, o uso de escaneamento a laser para criar nuvens de pontos em diferentes fases de um projeto também tem sido explorado na construção civil para otimizar processos (Wang e Kim, 2019).

A utilização de nuvens de pontos 3D produzidas por scanners a laser para gerar modelos BIM está se tornando uma cada vez mais presente na literatura, sendo utilizada nas fases de construção, reabilitação e manutenção de instalações em áreas que vão desde plantas de projeto até preservação histórica (Bosché et al., 2015). De acordo com Wang et al. (2015), a sinergia entre BIM e *LiDAR* (*Light Detection and Ranging*) abre novas possibilidades no campo da detecção de defeitos de construção e controle de qualidade em tempo real.

Na literatura é observado que o uso de nuvens de pontos dentro da AECO (Arquitetura, Engenharia, Construção e Operação) pode ter diferentes finalidades, como obtenção de informações ligadas à topografia, georreferenciamento de elementos, elaboração de projetos *as-built* e monitoramento do andamento da obra (Ebrahimi et al., 2023; Ma et al., 2022; O'Donnell et al., 2019; Puri & Turkan, 2020; Yin et al., 2023). A nuvem de pontos surge nesse cenário como um suporte aos levantamentos feitos de forma manual, pois permite a visualização de dados referentes ao espaço escaneado de forma digital e tridimensional, sendo possível inclusive medir a distância entre pontos e superfícies da nuvem obtida.

As nuvens de pontos obtidas nos escaneamentos são de grande valia para a elaboração de modelos *as-built*, tendo em vista que o levantamento é feito de maneira rápida e com grande volume de informações. Igor et al. (2023) apontam que a modelagem construtiva integrada a outras tecnologias tem o potencial de remodelar completamente o ambiente construído. Desde que o sensor seja posicionado em pontos estratégicos durante o levantamento, não há necessidade de revisitar o espaço para conseguir elaborar o projeto, pois as medições podem ser feitas no próprio visualizador da nuvem de pontos.

Apesar de suas vantagens, o processo de transformar uma nuvem de pontos em um modelo BIM apresenta algumas dificuldades na prática, pois não é tão intuitivo, sendo muitas vezes custoso e demorado (Bassier; Vergauwen, 2020). Enquanto no processo "tradicional" a

modelagem em *softwares* BIM pode ser feita a partir de um projeto em formato DWG, em que as linhas e vértices estão bem definidos, com a nuvem de pontos essa não há nenhuma superfície bem definida, apenas pontos aglomerados. A depender da especificação do equipamento utilizado para obtenção da nuvem de pontos, o resultado da nuvem de pontos pode ser ruidoso, pois quanto maior a precisão do equipamento, maior a densidade – e acurácia – da nuvem (Kim & Lee, 2023).

Este trabalho busca otimizar a classificação de nuvens de pontos de grandes áreas por meio de um algoritmo de aprendizado de máquina supervisionado, sendo este um primeiro passo para futura modelagem automática de modelos BIM a partir de nuvens de pontos, que ainda se apresenta um desafio na literatura (Sacks et al., 2017).

1.1. JUSTIFICATIVA

No âmbito AECO, é importante reduzir gastos e economizar tempo e recursos humanos, Junto a isso, investir em inovação está se tornando uma necessidade, pois traz vantagem competitiva às empresas e proporciona soluções mais eficazes aos problemas existentes. A utilização de tecnologias facilitadoras de processos dentro da construção civil, como é o caso do uso de nuvem de pontos para levantamentos, está cada vez mais presente. Por esses motivos, ainda que a utilização de nuvens de pontos dentro da engenharia ainda não esteja difundida, é necessário investir nessa tecnologia e entender suas peculiaridades, para que ela possa ser explorada em sua totalidade.

No mercado, apesar de haver soluções proprietárias consolidadas na classificação automática de nuvens de pontos, soluções de código aberto apresentam importantes vantagens competitivas, permitindo alternativas sem grandes impactos econômicos. Além disso, soluções *open source* também são dispositivos propícios para a produção descentralizada e colaborativa, intensificando o processo de inovação (Heron et al., 2013).

1.2. OBJETIVOS

Nesta seção do trabalho, estarão listados o objetivo geral e os objetivos específicos ligados ao desenvolvimento da pesquisa.

1.2.1 Objetivo Geral

O objetivo geral deste trabalho é realizar a classificação de nuvens de pontos de grandes áreas com uso de um algoritmo de aprendizado de máquina supervisionado.

1.2.2 Objetivos Específicos

Dentre os objetivos específicos deste trabalho estão:

a) Criar e alimentar um *dataset* de nuvens de pontos pré-classificadas para que sirvam de base de aprendizado do algoritmo a ser desenvolvido;
b) Desenvolver um algoritmo de aprendizado de máquina que absorva a informações do *dataset* criado e seja capaz de armazenar o conhecimento adquirido para que seja replicado;
c) Inserir nuvens de pontos não classificadas no algoritmo e obter como *output* a mesma nuvem classificada por meio do aprendizado de máquina;
d) Analisar a acurácia da classificação das nuvens inseridas e ser capaz de exportar a nuvem classificada para utilização em *softwares* variados.

1.3. ORGANIZAÇÃO DO TRABALHO

Este trabalho está subdividido em 6 capítulos, conforme estrutura detalhada nesta seção. Ao final do trabalho, também estão listadas as referências bibliográficas utilizadas.

- Capítulo 1: apresenta a introdução, justificativa e objetivos do trabalho desenvolvido;
- Capítulo 2: apresenta o referencial teórico utilizado como base para o desenvolvimento da pesquisa, apresentando os conceitos básicos necessários ao entendimento. É apresentado um breve contexto do cenário BIM do uso de nuvens de pontos na construção civil, além do algoritmo *Random Forest*;
- Capítulo 3: contém a descrição da metodologia e dos softwares utilizados para realização da pesquisa, além de apresentar a fonte de dados utilizada para aquisição das nuvens de pontos;
- Capítulo 4: apresenta o *script* de classificação de nuvens de pontos desenvolvido na pesquisa por meio de pseudocódigos;

- Capítulo 5: apresenta os resultados obtidos por meio da aplicação do *script* nas nuvens de pontos compara os efeitos que diferentes parâmetros têm nesses resultados. Também possui as discussões a partir da observação dos resultados obtidos e traz as limitações presentes na realização da pesquisa;
- Capítulo 7: traz as considerações finais e as principais contribuições do trabalho, além das perspectivas futuras para próximos trabalhos.

2 REFERENCIAL TEÓRICO

Nesta seção serão tratados os principais temas necessários ao entendimento da pesquisa realizada, com foco no *Building Information Modeling*; no uso das nuvens de pontos dentro da indústria da construção civil e os modos de aquisição dessas nuvens; e no algoritmo de classificação Random Forest. Também foi realizada uma revisão bibliométrica para avaliação dos estudos mais recentes desenvolvidos referentes à classificação de nuvens de pontos.

2.1. BUILDING INFORMATION MODELING (BIM)

O BIM, sigla para *Building Information Modeling*, é uma tecnologia que possibilita a elaboração de um modelo de construção rico em informações semânticas de forma digital, que permite gerenciar todo o ciclo de vida de um projeto antes mesmo do início da obra. Para Azhar (2011), um modelo BIM é caracterizado por informações de geometria, espaço, informações geográficas, quantidades e propriedades dos elementos do projeto, com estimativas de custo e materiais. Isto é, é uma ferramenta poderosa de gerenciamento pois compila diversas informações úteis para um melhor controle dos projetos.

O desenvolvimento da tecnologia BIM teve início na década de 1970, com os primeiros projetos sendo desenvolvidos por meio de CAD (*Computer Aided Design*), não só na engenharia civil, mas em diversos setores (Volk et al., 2014). A indústria da construção civil demorou a adotar de fato os desenhos parametrizados em relação às demais indústrias, pois existiu uma certa resistência a adotar novas tecnologias (Sacks et al., 2018). Ainda de acordo com Sacks et al. (2018), essa resistência pode estar relacionada ao fato de que a construção no canteiro não se beneficiou substancialmente a partir da automação, dado ao fato de que a produtividade em campo é muito ligada ao trabalho manual.

Apesar dessas questões, o uso do BIM vem sendo incentivado por meio de “mandatos BIM” em todo o mundo, tendo sido os primeiros documentos publicados pela Noruega, Dinamarca e Finlândia, antes mesmo do ano de 2010 (Sacks et al., 2018). No Brasil, a implementação da metodologia BIM tem recebido ainda mais destaque devido ao Decreto nº 10.306 datado de 2 de abril de 2020. Esse Decreto discorre sobre a obrigatoriedade da utilização do BIM na execução de obras e serviços de engenharia realizados por órgãos e entidades da administração pública federal (Brasil, 2020).

O Decreto ainda estabelece que, a partir de 1 de janeiro de 2021, o BIM deve ser utilizado no desenvolvimento de projetos de arquitetura e engenharia, sendo construções novas,

ampliações ou reabilitações (Brasil, 2020). Dado esse cenário de incentivo à adoção de técnicas ligadas ao BIM, é importante o desenvolvimento e o aperfeiçoamento de novas tecnologias que surgem a partir dos desafios em se adotar um novo modo de gerenciar os projetos.

Succar (2009) define o BIM como uma junção de três campos distintos que conversam entre si, o campo da tecnologia, o campo dos processos e o campo das políticas, que podem ser observados na Figura 1. O campo das tecnologias envolve equipamentos, sistemas de comunicação, *softwares*, *hardwares* e tecnologias complementares ao BIM; já o campo dos processos envolve pessoas, modelos, componentes, fornecedores e outros; o campo das políticas envolve as melhores práticas, centros de pequisa, normas, regulamentos, órgãos reguladores e outros (Succar, 2009).

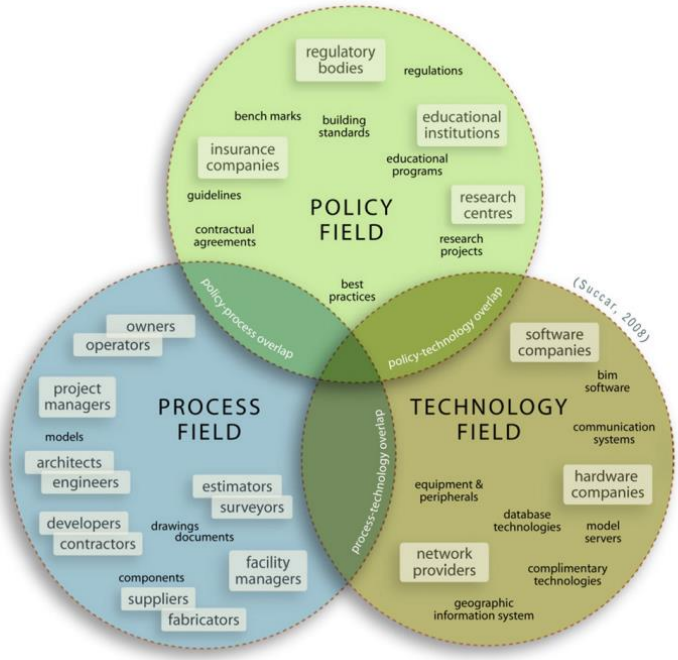

Figura 1: Campos BIM

Fonte: Succar (2009).

A partir desse conjunto, é possível analisar que o BIM não se trata apenas de uma tecnologia ou uma plataforma ou um modelo, mas sim de uma ferramenta de gestão completa, capaz de envolver todas as fases do ciclo de vida de um projeto. De acordo com Wang e Chen (2023), o BIM como método de gestão deve ser aplicado continuamente num projeto para concretizar o seu enorme potencial. Essa ideia é reforçada por Sacks et al. (2018), que reforça que um dos benefícios mais importantes do BIM é a coordenação ativa que o construtor pode alcançar quando todos os projetistas utilizam o modelo BIM em seus respectivos projetos.

Dada a importância da utilização da tecnologia para gestão das infraestruturas urbanas, o conceito BIM também foi expandido para as cidades, sendo chamado de *City Information*

Modeling (CIM). Alguns trabalhos utilizam a mesma terminologia BIM para se referir aos modelos 3D parametrizados das cidades, como em "BIM *for infraestructure*", comumente utilizado pela *Autodesk, Inc*. Amorim (2015) discute sobre as diferentes conceituações existentes sobre o CIM e conclui que uma visão CIM precisa contemplar necessariamente uma visão de BIM, da mesma forma que o BIM engloba o conceito de CAD.

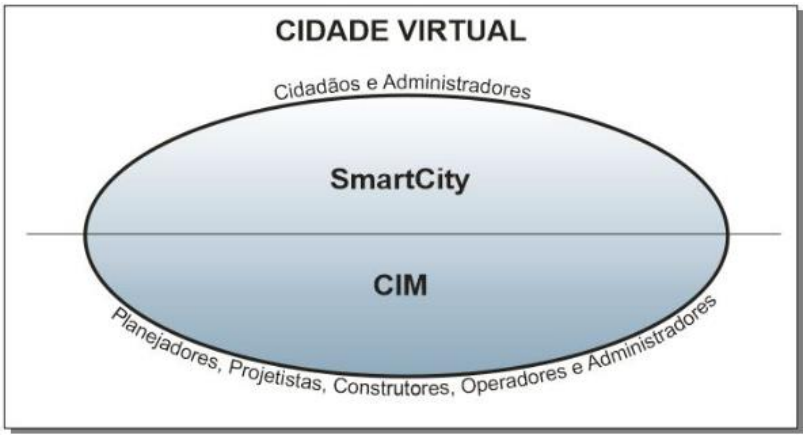

Figura 2: Definição de uma cidade virtual (CIM).

Fonte: (Amorim, 2015).

O CIM é de grande valia para os órgãos de manutenção das cidades, pois de acordo com Lopes (2019), um modelo de cidade paramétrico e organizado pode ser um agente centralizador de informações, prevendo situações ajustadas à realidade. Além disso, o CIM possui forte sinergia com sistemas de informação geográfica (SIG).

O BIM, no geral, trouxe para a indústria AECO diversos benefícios em relação ao gerenciamento do ciclo de vida dos projetos. Dentre esses benefícios, estão a redução de erros durante a fase de execução do projeto, redução de custos, otimização do cronograma e outros. Pode-se dizer que a indústria está passando por um grande processo de transformação digital, pois o BIM não só vem mudando a forma como os projetos são feitos, mas também a comunicação entre as partes interessadas no projeto.

A adoção do BIM por si só permite maior eficiência do projeto, mas esse ganho pode ser aumentado se integrado com outras tecnologias emergentes, como a realidade virtual, e a digitalização a laser para aquisição de informações *as-built* ou sistemas autônomos para monitoramento automatizado (Abreu et al., 2023).

2.2. NUVEM DE PONTOS NA CONSTRUÇÃO CIVIL

Ao longo do tempo, outras tecnologias existentes foram sendo agregadas à construção civil, como é o caso das nuvens de pontos. Nuvens de pontos são um conjunto de pontos definidos em um espaço métrico 3D e, de acordo com (Bello et al. (2020), se tornaram um dos formatos de dados mais significativos para representação 3D e estão ganhando popularidade crescente como resultado da maior disponibilidade de dispositivos de aquisição.

As nuvens de pontos fornecem uma grande gama de possibilidades dentro da construção civil. As aplicações vão desde um simples registro da obra como uma imagem em 3D, até a construção dos chamados *Tours Virtuais*, que permitem que o usuário seja inserido na obra mesmo sem estar nela, inclusive com o auxílio de óculos de realidade aumentada (Luleci et al., 2024). Além disso, está bastante presente na literatura como uma ferramenta para preservação histórica de edificações (Escudero, 2023).

Nuvens de pontos podem ser obtidas por diferentes métodos, como *scanners* terrestres ou aéreos, que podem ser utilizados em Veículos Aéreo Não Tripulados (VANTs), popularmente conhecidos como drones. Além disso, também há a possibilidade de se obter nuvens de pontos por meio de fotogrametria. As tecnologias abordadas nesse trabalho limitam-se ao escaneamento realizado por *LiDAR* e à fotogrametria, sendo cada um desses métodos relevantes para diferentes objetivos, que serão detalhados nesta seção.

Estudos mais recentes investem em investigar a construção de modelos BIM de forma automática a partir de nuvens de pontos, mas isso ainda é tido como um desafio a ser superado (Sacks et al., 2017). Na literatura as nuvens são muito utilizadas para verificação do avanço da obra em projetos de construção, como em Ojeda et al. (2024), que utilizou nuvens de pontos como ponte para avaliação do progresso físico da obra de um laboratório, por meio da comparação da nuvem gerada com o modelo BIM projetado.

Nuvens de pontos também são úteis para engenharia geotécnica, pois permitem a medição automatizada e periódica de locais amplos por meio do uso de VANTs, como no estudo desenvolvido por Lee et al. (2023). Também é bastante utilizada em projetos rodoviários, como no estudo desenvolvido por Hiasa & Kawamoto (2023), em que foi criado um modelo digital de terreno (DTM) para projetos de rodovias 3D.

Seguindo a mesma linha, Pérez et al. (2022) realiza uma análise comparativa de um levantamento realizado por topografia tradicional e um por meio de nuvens de pontos obtida por drone para projetos rodoviários. Foi observado que a junção do levantamento topográfico

clássico com os dados provenientes do levantamento do drone geraram um produto bem-sucedido (Pérez et al., 2022).

Outro uso bastante comum é no campo da preservação histórica e restauração de edifícios históricos, sendo inclusive um campo específico dentro do BIM, conhecido como *Heritage BIM* ou (H-BIM). Escudero (2023) estuda um processo automatizado de transformar nuvens de pontos em modelos tridimensionais para representar edifícios históricos, para facilitar projetos de restauro e gestão do patrimônio cultural.

Croce et al. (2023a) reforça que atualmente a reconstrução de modelos H-BIM a partir de digitalização a laser ou fotogrametria é um processo manual, demorado e excessivamente subjetivo, mas o surgimento de técnicas de inteligência artificial (IA) está oferecendo novas maneiras de interpretar, processar e elaborar dados brutos de levantamento digital.

2.2.1 LiDAR (Light Detection and Ranging)

O *LiDAR* é um método de sensoriamento que se utiliza feixes de luz para medir a distância entre os objetos no ambiente e o sensor, sendo a distância determinada pelo tempo de reflexão entre o pulso do *laser* e o sensor. Um esquema simples do funcionamento da aquisição de dados por *LiDAR* pode ser observado na Figura 3. Essa tecnologia é utilizada em diversos setores, e na engenharia civil é muito utilizada para criação de modelos 3D de ambientes reais. O *LiDAR* é considerado uma das tecnologias de sensores mais cruciais para veículos autônomos, robôs artificialmente inteligentes e reconhecimento de veículos aéreos não tripulados (Albano, 2019).

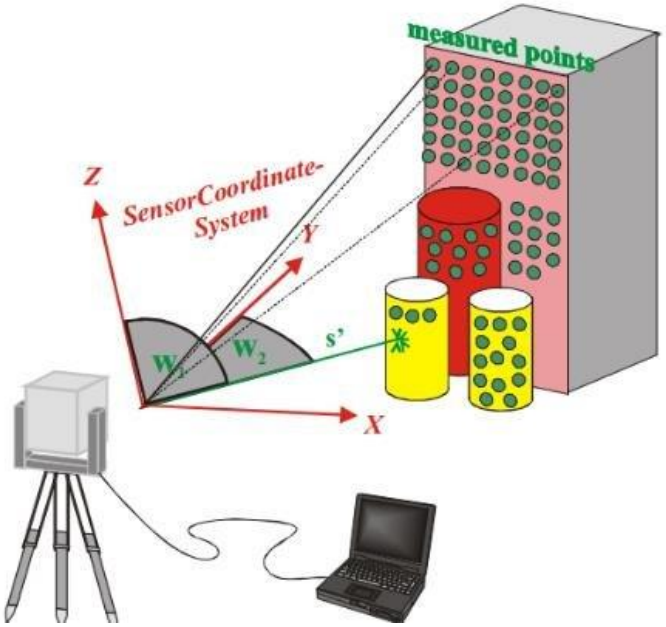

Figura 3: Esquema do escaneamento a laser terrestre.

Fonte: Khomsin et al., (2019).

De acordo com Tang et al. (2010), o processo de criação de um modelo *as-built* a partir de *lasers scanners* consiste em três etapas: coleta de dados, pré-processamento de dados e modelagem do BIM. Na etapa de coleta de dados, o *laser scanner* é utilizado para medir a distância entre os pontos da estrutura existente e o sensor do equipamento utilizado com grande acurácia, tendo como produto uma nuvem de pontos densa (Tang et al., 2010).

No pré-processamento de dados, pode-se realizar um tratamento da nuvem de pontos para exclusão de informações desnecessárias ao projeto e também para redução do tamanho do arquivo. Já na modelagem BIM, a imagem 3D formada pela nuvem de pontos auxilia na transformação de informações não paramétricas em objetos paramétricos (Tang et al., 2010). Um dos problemas da utilização de nuvem de pontos para modelagem BIM é que o processo pode ser bastante manual, demorado e propenso a erros (Anil et al., 2011).

Uma grande vantagem da adoção do escaneamento a *laser* para levantamento em alternância aos procedimentos de rotina tradicionais é a grande quantidade de dados que podem ser coletados de forma rápida e que permitem uma visualização de imagens claras em *softwares* de visualização de nuvens de pontos. Essa ideia é corroborada por Moyano et al. (2022), ao afirmar que os dados de nuvens de pontos 3D provenientes de técnicas de aquisição de dados, como a fotogrametria e o *LiDAR*, desempenham um papel importante devido à grande quantidade de informação que fornecem.

Mihić et al. (2023) acrescentam que o *LiDAR* começou a ser incorporado recentemente em alguns *smartphones* e *tablets*, mas essa tecnologia em dispositivos portáteis ainda não possui precisão o suficiente para gerar nuvens de pontos que possam ser usadas para controle de qualidade e quantidade. Outro modo de aquisição de nuvens de pontos comumente utilizado na engenharia civil é a fotogrametria, que possui diferenças metodológicas e morfológicas em relação ao *LiDAR* e será apresentado na continuidade desta seção.

2.2.2 Fotogrametria

A fotogrametria consiste na construção de imagens, modelos tridimensionais ou mapas a partir da aquisição de fotografias. Na engenharia civil, as fotografias obtidas para fotogrametria são normalmente obtidas por meio de VANTs. Para a criação de modelos 3D por meio da fotogrametria, são utilizadas imagens estereoscópicas, que se baseiam na paralaxe, que é resultante da diferença angular da visada de um mesmo objeto (Santos et al., 2022). Ou seja, duas fotos são capturadas de ângulos diferentes, a fim de obter a profundidade dos elementos presentes na imagem.

Como um exemplo de como são capturadas as imagens, pode-se observar a Figura 4, onde o drone faz um voo e parte do ponto O1 para o O2 capturando imagens que possuem sobreposição entre si. A junção de imagens capturadas com a câmera do drone de forma perpendicular à superfície com as imagens feitas com a câmera com um determinado ângulo, sobrepostas, permitem a obtenção de nuvens de pontos.

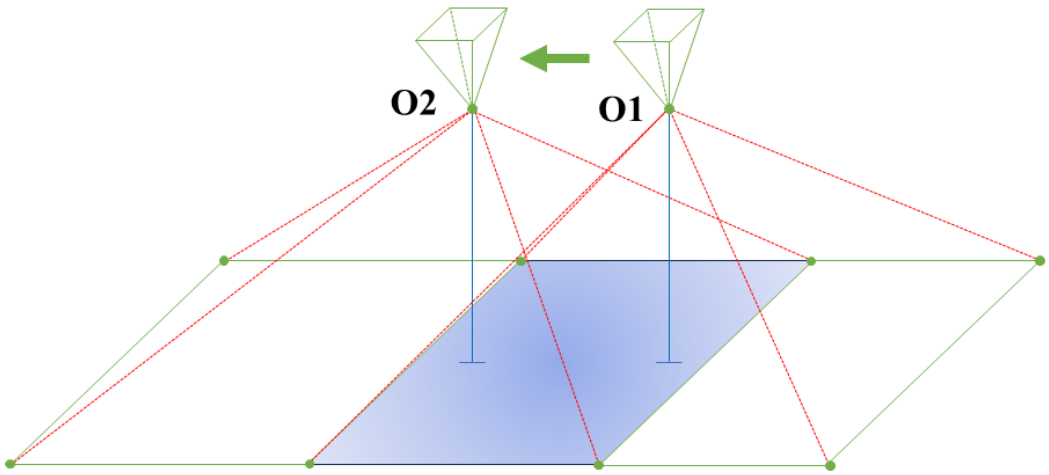

Figura 4: Esquema de aquisição de imagens fotogramétricas.

Fonte: A autora.

Hammad et al. (2021) aponta que, na construção civil, as pesquisas têm focado na utilização de VANTs em conjunto com BIM, não só para modelagem 3D, mas também para simulações de cronograma da obra. O autor acrescenta que os dados obtidos pelos levantamentos de imagens podem ser integrados a modelos BIM para um melhor planejamento do projeto, com base nas condições do solo e na locação de outros elementos. Isso de certa forma remete ao uso do *LiDAR* mencionado anteriormente para aquisição de dados de topografia e, ainda que as tecnologias sejam diferentes, podem ser utilizadas de forma semelhante.

Mihić et al. (2023) aponta que a fotogrametria pode ser utilizada para tanto para a geração de nuvens de pontos quanto para visão computacional, enquanto a varredura a *laser* só pode ser usada para gerar dados de nuvens de pontos. Também acrescentam que, para que nuvem de pontos 3D fotogramétrica seja utilizável, as imagens precisam ter qualidade adequada em termos de resolução e desfoque e precisam ter pelo menos 50-60% de sobreposição para permitir a execução da recriação da realidade usando a fotogrametria como técnica (Mihić et al., 2023).

As configurações de voo de drones para obtenção de nvens de pontos podem ser ajustadas por meio de *softwares* específicos voltados para VANTs. O *DroneDeploy*, por exemplo, permite que haja um planejamento do voo de determinada área com dados prévios do tempo de voo, quantas imagens serão capturadas e a resolução das imagens que serão obtidas a partir da altura de voo desejada. Nesse caso, quanto menor a altura do voo, maior a resolução das imagens.

2.3. SCAN-TO-BIM

O desenvolvimento de modelos BIM é comum para desenvolvimentos de projetos ainda em fase de planejamento, em que são observadas as possíveis interferências entre disciplinas e onde há espaço para modificações ainda em fases iniciais, promovendo a interoperabilidade entre disciplinas ainda na fase de planejamento. Em projetos de reforma, operação e manutenção, no entanto, especialmente em edificações mais antigas, é comum que não haja modelos BIM a serem consultados. Por esse motivo, quando há a necessidade de desenvolvimento de um modelo BIM de uma edificação existente, o processo de levantamento de informações pode ser trabalhoso e demorado.

Para dar suporte à resolução dos problemas decorrentes dos casos em que o modelo BIM é inexistente ou há informações de construção imprecisas, desatualizadas ou ausentes,

tecnologias de captura da realidade, como a digitalização a laser de ambientes, passaram a ser adotadas na indústria AECO (Wang et al., 2019). Nesse cenário, o uso de nuvens de pontos serve como um suporte ao levantamento dessas informações para elaboração de projetos.

Ainda que seja um cenário promissor, de acordo com Mahmoud et al. (2024), as abordagens atuais desenvolvidas para o processo de digitalização para BIM dependem de procedimentos manuais ou semiautomáticos e não aproveitam de forma suficiente os dados semânticos das nuvens de pontos, o que causa ineficiência nos modelos BIM. Isso ocorre porque a informação proveniente das nuvens de pontos não tem significado dentro dos modelos BIM sem que haja um processo que transforme os elementos capturados pelo equipamento de escaneamento a laser em elementos BIM com informação semântica.

Dentro desse contexto surge o *Scan-to-BIM*, que é um termo que se refere à integração das tecnologias de escaneamento a laser com o desenvolvimento de modelos BIM de forma automática a partir de nuvens de pontos. De acordo com Lee et al. (2020), um modelo 3D construído a partir de tecnologias de escaneamento a laser contém apenas informações sobre o formato geométrico da estrutura, sendo necessária a atribuição de informação semântica às nuvens. Os autores indicam que o processo *Scan-to-BIM* requer a extração de parâmetros das nuvens de pontos para obter dados numéricos como altura, comprimento e largura, e que as pesquisas desenvolvidas na literatura se concentram em formas universais (pilares, vigas, etc) (Lee et al., 2020).

Bosché et al. (2015) aponta que um grande indicativo do valor do *Scan-to-BIM* é o crescimento exponencial do mercado de *hardwares* e *softwares* voltados para a digitalização a laser na última década. Apesar das diversas vantagens de realizar levantamentos escaneamento a laser dentro para posterior modelagem BIM, também deve ser mencionado que esse não é um processo simples e pode ser demorado e pouco intuitivo.

Para facilitar a extração de informação semântica das nuvens de pontos para uso dentro do contexto BIM, modelos de algoritmo de aprendizagem de máquina vêm sendo utilizados na literatura com diferentes funcionalidades, como a segmentação semântica e segmentação de objetos (Wang et al., 2022). Os métodos de aprendizado de máquina são usados de forma semelhante em um contexto específico de classificação de objetos de nuvens de pontos (Diab et al., 2022). Especialmente para esse contexto de reconhecimento de padrões, características ou objetos em nuvens de pontos, diferentes metodologias baseadas em algoritmos distintos são utilizadas para processar digitalmente os dados das nuvens de pontos (Xu et al., 2021).

2.4. ALGORITMOS DE APRENDIZADO DE MÁQUINA

Normalmente, os algoritmos de aprendizado de máquina podem ser classificados em três grupos de acordo com a natureza do feedback fornecido ao programa. O primeiro desses grupos é chamado de “aprendizagem supervisionada”, onde o termo supervisionado indica que o algoritmo recebe exemplos de saídas desejadas a partir dos dados de entrada (Ayodele, 2010). O segundo tipo é denominado “aprendizagem não supervisionada” e agrupa programas que recebem apenas o conjunto de dados de entrada, sem qualquer classificação prévia ou indicação da saída desejada (Ayodele, 2010). Por fim, o terceiro grupo é denominado “aprendizado por reforço” e consiste no aprendizado baseado em um sistema de recompensa-punição, onde o algoritmo recebe incentivos para reforçar associações ou desfazê-las (Ayodele, 2010).

Embora algoritmos de diferentes grupos possam ser usados para classificar dados de uma nuvem de pontos, o conceito por trás da metodologia utilizada é fundamentalmente diferente. Na aprendizagem supervisionada, por exemplo, o objetivo central é aprender e generalizar um conjunto de regras que correlacionam dados de entrada e saída. Na aprendizagem não supervisionada, o objetivo central pode girar em torno da descoberta de novas correlações entre dados que não foram descobertos anteriormente. Além disso, essas particularidades impactam diretamente nos parâmetros operacionais, como custo computacional, velocidade de processamento e precisão.

Por exemplo, Zhang & Rabbat (2018) propõem um algoritmo baseado em redes neurais convolucionais gráficas (Graph-CNN) para classificar objetos da nuvem de pontos. Embora as CNNs possam potencialmente apresentar maior precisão, estão comumente associadas à maior complexidade computacional, resultando em maior custo computacional (Zhang & Rabbat (2018). Além disso, as CNNs geralmente dependem de conjuntos de dados maiores e são mais suscetíveis ao *overfitting.*

O *overfitting* ocorre quando o modelo funciona perfeitamente no conjunto de treinamento, mas se ajusta mal ao conjunto de testes, isso porque um modelo superajustado tem dificuldade em lidar com informações do conjunto de teste, que podem ser diferentes daquelas do conjunto de treinamento (Decuyper et al., 2019). Essas limitações são amenizadas com o uso de algoritmos como o *Random Forest*, uma vez que propriedades inerentes ao algoritmo como crescimento sem poda e randomização dos classificadores atuam diretamente sobre esses fatores (Rodriguez-Galiano et al., 2015).

Apesar das limitações quanto ao nível de complexidade, alguns trabalhos baseados em CNNs utilizam estratégias auxiliares para mitigar esse alto custo computacional. Han et al.

(2023), por exemplo, usaram uma rede neural convolucional reduzida associada a um codificador e decodificador para gerar desenhos CAD para arquitetura paisagística com dados de realidade tridimensionais (3D) digitalizados via drone, câmera e *LiDAR*. Esta abordagem torna a solução mais eficiente em termos computacionais, mas não isenta o modelo da ocorrência de *overfitting.* Além disso, essa simplificação deve ser realizada com cautela, pois pode diminuir drasticamente a precisão do modelo.

Outro algoritmo de aprendizado supervisionado amplamente utilizado no processamento de nuvem de pontos é a máquina de vetores de suporte (SVM). Shirowzhan et al. (2019), por exemplo, usam um SVM para processar imagens *LiDAR* com o objetivo de detectar mudanças em edifícios. Este algoritmo utiliza uma técnica de aprendizagem estatística para resolver problemas de separação de classes e pode apresentar uma complexidade computacional menor que a CNN, especialmente para conjuntos de dados maiores.

Apesar disso, o desempenho do algoritmo SVM é vulnerável a amostras rotuladas incorretamente e conjuntos de dados de entrada desequilibrados, incluindo aqueles com grandes disparidades numéricas entre classes (Sothe et al., 2020). Essas limitações podem inviabilizar o SVM em algumas situações específicas, como na identificação de vegetação em conjuntos de nuvens de pontos em grandes centros urbanos, ou em áreas com classificação ruidosa.

2.4.1 RANDOM FOREST

O Random Forest (RF) consiste em um algoritmo de aprendizado de máquina supervisionado amplamente utilizado para classificação de dados e foi inicialmente proposto em 2001 por Breiman (2001b). Essa classificação pode ser definida como o ato de rotular de forma coerente elementos de um grupo e, especificamente no caso do RF, é realizada por meio de classificadores independentes (Parmar et al., 2019).

Os classificadores são utilizados para gerar estruturas chamadas árvores de decisão e atuam sobre subconjuntos aleatórios dos dados de treinamento, justificando o nome do algoritmo (Breiman, 2001). A intenção central desta metodologia é permitir previsões generalizadas para um conjunto de dados a partir da combinação do processamento desses subconjuntos (Breiman, 2001b; Parmar et al., 2019).

Metaforicamente, o RF pode ser comparado a um processo de tomada de decisão por votação, onde a decisão final é tomada com base nas opiniões individuais de vários especialistas (árvores de decisão). Esses indivíduos opinam com base em um conjunto de informações

(características) aleatórias sobre o problema. Suas opiniões são combinadas ao final do processo para compor a decisão final mais robusta. Na Figura 5 está indicado visualmente esse processo.

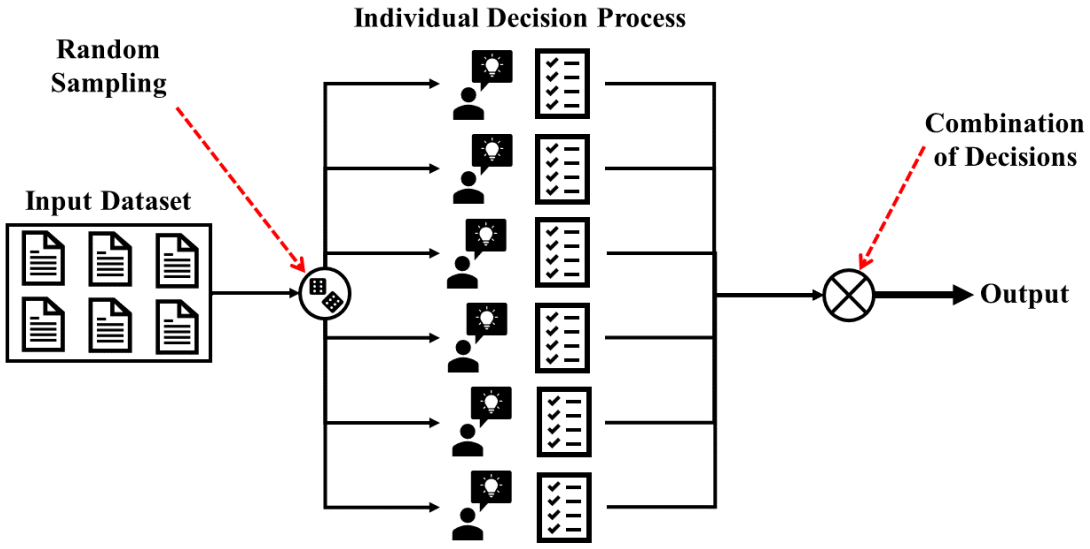

Figura 5: Processo do Random Forest ilustrado.

Fonte: A autora.

As árvores de decisão são construídas a partir da análise de um conjunto de exemplos de treinamento onde os rótulos de classe são conhecidos (exemplo: R, G, B, X, Y, Z) e são então aplicadas para classificar exemplos que não foram vistos anteriormente (Kingsford & Salzberg, 2008). (Kingsford & Salzberg, 2008) acrescentam ainda que, se as árvores de decisão forem treinadas com dados de alta qualidade, são capazes de fazer previsões com alto grau de precisão.

De maneira mais precisa, o *Random Forest* recebe, além do conjunto de treino, as características consideradas para análise e o número de árvores de decisão. A partir dessas informações, o algoritmo implementa um processo conhecido como *bootstrapping* que consiste na execução de uma reamostragem estatística a partir de uma pré-amostragem aleatorizada com reposição (Breiman, 2001; Rodriguez-Galiano et al., 2015). Esse processo ocorre para cada árvore, sendo um dos pontos chaves por trás da alta eficiência computacional do RF, pois permite aumentar artificialmente o conjunto de treinamento sem a necessidade de adquirir e processar novos conjuntos de dados. Além disso, essa expansão no conjunto de treinamento também é especialmente útil para mitigar a possibilidade de ocorrência de *overfitting* (Rodriguez-Galiano et al., 2015).

Complementarmente, cada uma das árvores produzidas durante o processo de *bootstrapping* é composta por estruturas chamadas de nós que recebem um subconjunto aleatório das características de interesse. Esse segundo processo aleatório promove o aumento da "diversidade da floresta" que, por sua vez, também contribui significativamente para a eficiência do algoritmo (Rodriguez-Galiano et al., 2015). Por fim, também é válido salientar que a combinação do mecanismo de *bootstrapping* com a diversidade entre as árvores torna o RF especialmente robusto à ocorrência de ruído e outliers nos dados (Rodriguez-Galiano et al., 2015). O algoritmo abaixo sintetiza o funcionamento do RF ao passo que a Figura 6 ilustra a estrutura de árvores obtidas a partir da execução do Algoritmo 1.

Algoritmo 1: Random Forest

Inputs: Training set X: {(x1, y1), (x2, y2), …, (xn, yn)}, features to consider at each split (F) and the number of trees in the forest (N).

```
1.  RandomForest (X, F, N)
2.          K ← ∅
3.          for i = 1 to N:
4.                  X^(i) ← statistical resampling using random sampling with replacement of X
5.                  k_i ← RandomizedTreeLearn(X^(i), F)
6.                  K ← K ⋃ k_i
7.          return K
8.
9.  RandomizedTreeLearn(X, F)
10.     At each node:
11.             select the best features from a small subset of F
12.             return the learned tree
```

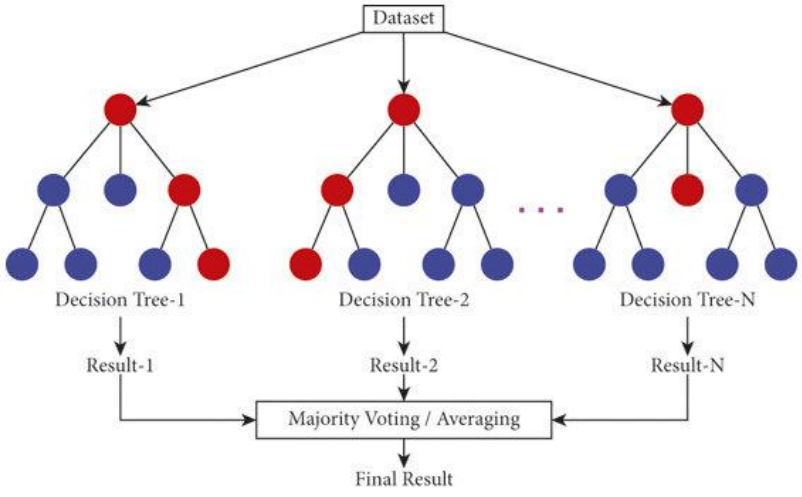

Figura 6: Esquema de funcionamento do Random Forest.

Fonte: Khan et al. (2021).

Com o crescimento do aprendizado de máquina e da exploração da inteligência artificial, diversas bibliotecas e ferramentas de segmentação de imagem se popularizaram e estão disponíveis em código aberto em diversas linguagens de programação, como é o caso do Python que contém a *RandomForestClassifier*. Essas bibliotecas facilitam o desenvolvimento de algoritmos por serem pacotes de códigos prontos, que podem ser importados para serem utilizados em algoritmos de forma simplificada.

2.4.2 MÉTRICAS PARA AVALIAÇÃO DE MODELOS DE APRENDIZADO

A avaliação precisa de um algoritmo de aprendizado de máquina é fundamental para planejar o algoritmo do classificador e a melhoria do desempenho nos resultados obtidos (Heydarian et al., 2022). A avaliação da eficiência de algoritmos de aprendizado é comumente realizada por meio de algumas métricas, como precisão, acurácia, *recall* e *f-score*, e são importantes para entender o comportamento dos modelos. No Quadro 1 estão listadas as métricas utilizadas nesse trabalho, bem como suas fórmulas e descrições.

Quadro 1: Métricas de avaliação de classificação.

Métrica	Fórmula	Descrição
Accuracy (acc)	$\frac{tp + tn}{tp + fp + tn + fn}$	Em geral, a métrica de precisão mede a proporção de previsões corretas sobre o número total de instâncias avaliadas
Precision (p)	$\frac{tp}{tp + fp}$	A precisão é usada para medir os padrões positivos que são previstos corretamente a partir do total de padrões previstos em uma classe positiva
Recall (r)	$\frac{tp}{tp + fn}$	É usado para medir a fração de padrões positivos que são classificados corretamente.
F-measure (FM)	$\frac{2 * tp}{2 * tp + fp + fn}$	Essa métrica representa a média harmônica entre os valores de *recall* e precisão
Averaged Accuracy	$\frac{\Sigma_{i=1}^{l} \frac{tp_i + tn_i}{tp_i + fn_i + fp_i}}{l}$	A eficácia média de todas as classes
Averaged Precision	$\frac{\Sigma_{i=1}^{l} \frac{tp_i}{tp_i + fp_i}}{l}$	A média de precisão por classe

Averaged Recall	$\frac{\Sigma_{i=1}^{l} \frac{tp_i}{tp_i + fn_i}}{l}$	A média de *recall* por classe
Averaged F-measure	$\frac{2 * p_M * r_M}{p_M * r_M}$	A média de F-measure por classe

Fonte: Adaptado de Hossin e Sulaiman (2015).

Sendo:

tp: true positive (verdadeiros positivos);

fp: *false positives* (falsos positivos);

fn: false negatives (falsos negativos);

tn: true negatives (verdadeiros negativos);

M: macro-averaging (média macro);

i: cada classe.

A partir da acurácia (*accuracy*), a qualidade do modelo pode ser avaliada com base na porcentagem de previsões corretas sobre o total de instâncias existentes (Hossin e Sulaiman, 2015). Dentre as métricas existentes, Hossin e Sulaiman (2015) apontam que a acurácia é a mais utilizada na prática, tanto para problemas de classificação binária quanto para problemas que envolvem múltiplas classes.

De acordo Foody (2023), o *recall* é amplamente utilizado em áreas como ciência da computação e aprendizado de máquina, e indica quão bem o classificador rotula como positivos os casos que realmente são positivos, mostrando a capacidade de identificar corretamente os casos que são positivos. Outras métricas também amplamente utilizadas são o *precision* e o *F-measure* (ou *F1-score*); o *precision* seria uma métrica alternativa ao *recall*, que também foca nos casos positivos; já o o *F-measure* combina essencialmente as informações das duas métricas anteriores em uma média harmônica (Foody, 2023).

Nas tarefas de classificação multiclasse, onde cada uma das instâncias existentes só pode ser rotulada como uma classe, a matriz de confusão é uma ferramenta poderosa para avaliação de desempenho, quantificando a sobreposição de classificação (Heydarian et al., 2022). A matriz de confusão se trata de uma matriz quadrada em que as linhas representam a classe real das instâncias e as colunas a classe prevista (Caelen, 2017).

2.5. ANÁLISE BIBLIOMÉTRICA

A análise bibliométrica é uma metodologia que fornece informações acerca de um determinado tema por meio da análise de um conjunto de dados científicos. Para avaliar o cenário do tema pesquisado dentro da literatura, foi realizada uma análise bibliométrica simples, que possibilitou a observação das tendências e possibilidades que envolvem a classificação de nuvens de pontos e o BIM.

O primeiro passo para desenvolver a análise bibliométrica foi a definição da plataforma de dados e os termos de pesquisa que comporiam os filtros. A plataforma utilizada na análise foi a *SCOPUS*, que pertence a *Elsevier*, e as palavras-chave utilizadas foram: "BIM", "*classification*" e "*point cloud*". A definição das palavras-chave utilizadas buscou abranger o maior número de trabalhos ligados à pesquisa desenvolvida, sendo considerados também as variações dos termos utilizados, como "*building information modeling*" e "*point clouds*".

Os critérios de busca também foram limitados a trabalhos em inglês, que fossem artigos e especificamente da área de engenharia, o que resultou em 50 artigos para compor o conjunto de dados a ser avaliado. Para avaliação dos dados, foi utilizado o software de código aberto, *VOSViewer*, que permite a construção de mapas de co-ocorrência a partir de informações relacionadas do conjunto de dados.

Para aumentar a precisão do processo, uma nova etapa de filtragem listou todas as palavras-chave existente nos artigos e aquelas que possuíam termos equivalentes, porém escritos de forma diferente (como em *3d modeling* e *3d modelling*), foram adaptadas para serem reconhecidas como o mesmo termo. Após esse processo, foi construído o mapa de palavras que pode ser visto na Figura 7, que mostra não só as ligações entre as palavras-chave existentes nos trabalhos, mas também a tendência dos temas durante os anos a partir de uma escala de cor.

No mapa de correlação entre as palavras-chave é possível observar a predominância dos assuntos BIM, *classification* e *point cloud*, tendo em vista que foram as palavras utilizadas para obtenção do conjunto de artigos a ser analisado. O tamanho dos nós tem relação com a quantidade de vezes que a palavra-chave apareceu nos artigos, sendo a presença de "*architectural design*" forte no conjunto de dados.

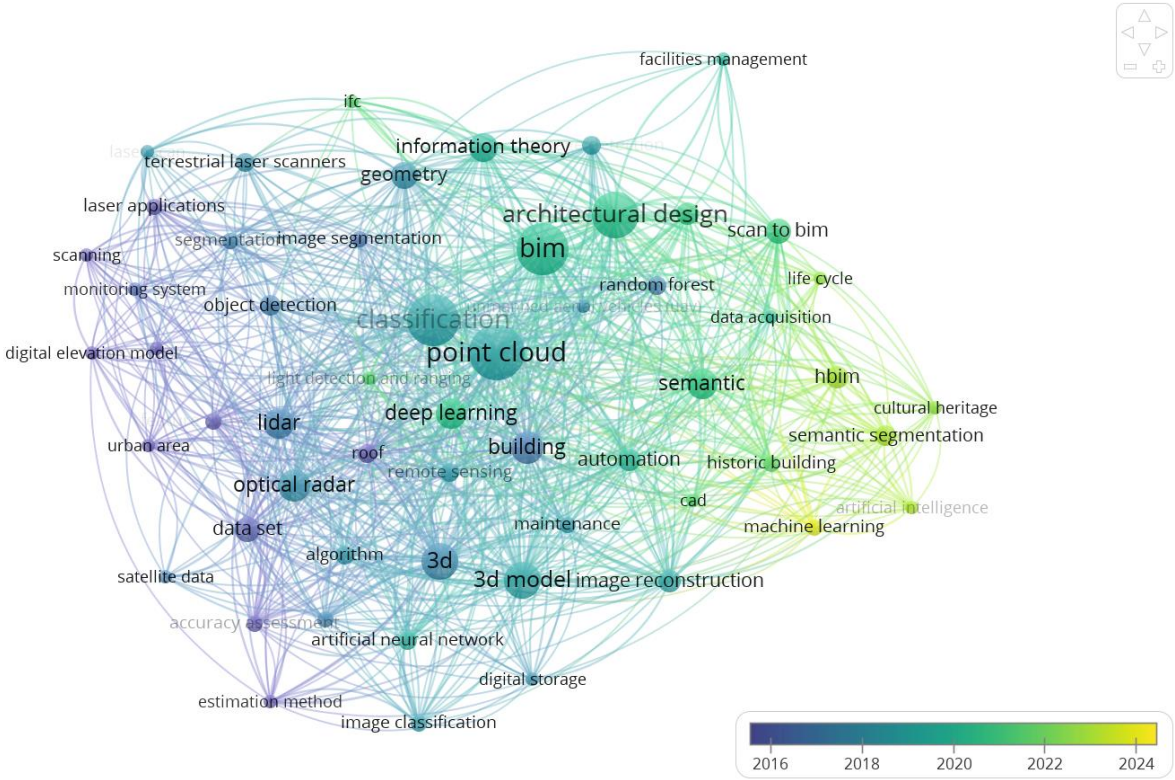

Figura 7: Correlação entre palavras-chave.

Fonte: A autora.

Pode-se observar como em 2016 havia uma maior tendência de estudos envolvendo aplicações de *laser scanner* para obtenção das nuvens de pontos, indicando uma maior preocupação com a classificação de objetos em si, com alguns trabalhos indicando o interesse em detecção de objetos e segmentação de imagens. Dimitrov e Golparvar-Fard (2014) propõe um método de classificação de materiais presentes em elementos de nuvens de pontos a partir de um banco de imagens construído em um canteiro de obras, com foco no monitoramento automático do progresso da construção.

Nesse mesmo sentido, Han & Golparvar-Fard (2015) também usam nuvens de pontos para monitoramento da construção e foi observada uma grande precisão para a inferência no estado da obra a partir do reconhecimento automático de materiais da nuvem por meio de um banco de dados de imagem. À medida que os anos avançam, as pesquisas passam a se voltar para a processamento a partir das nuvens de pontos, pois supõe-se que a etapa de levantamento já está bem estabelecida na literatura. O foco nos anos mais recentes passa a ser a utilização de *deep learning*, *machine learning*, inteligência artificial e segmentação semântica das nuvens.

Bahreini e Hammad (2024), por exemplo, utilizaram um método de segmentação semântica baseado em *deep learning* para reconhecer patologias específicas em nuvens de pontos e, posteriormente, essas patologias seriam levadas aos modelos BIM. Um ponto a ser

ressaltado neste trabalho é que apesar da alta precisão no reconhecimento das patologias, a informação semântica delas no modelo BIM precisou ser feita de modo manual. Esse é um problema recorrente nos estudos analisados, pois automatizar a modelagem com toda informação necessária aos elementos BIM a partir das nuvens de pontos ainda é um desafio na literatura.

Lee et al. (2023) avaliou a precisão do levantamento realizado por *LiDAR* acoplado a um drone e concluiu que foi útil para monitorar recalques no solo de uma grande área remota. Os resultados de avaliação dos recalques cumulativos ao longo de vários acompanhamentos periódicos mostraram diferenças 13 cm, que os autores avaliaram como um erro razoável em comparação com outros trabalhos semelhantes (Lee et al., 2023).

Ainda que não tenha sido incluído nos critérios de busca iniciais para abranger uma quantidade maior de artigos, o termo "*Random Forest*" também consta no mapa. É possível observar a correlação das palavras-chave a partir do nó "*Random Forest*" na Figura 8, onde pode-se destacar "*UAV*" que se refere a veículos aéreos não tripulados (drones), *scan-to-BIM*, segmentação de imagens, *LiDAR* e geometria.

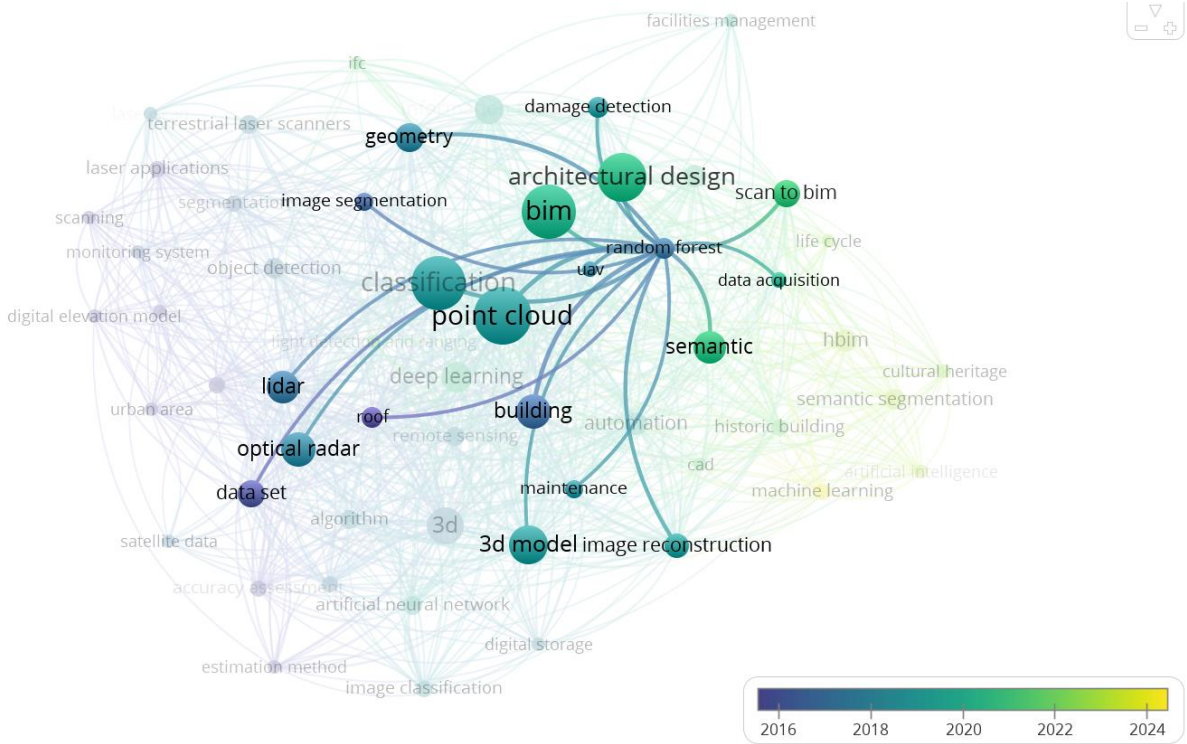

Figura 8: Principais correlações a partir do Random Forest.

Fonte: A autora.

É possível observar a relação do uso do RF com aquisição de dados, classificação, segmentação de imagens, manutenção, geometria, detecção de danos e outros. Xu et al. (2018)

utilizaram o RF para classificação de pontos em modelos cilíndricos e cubóides de um andaime, buscando estimar o progresso geral da obra pela disposição dos andaimes. Os pontos do andaime foram detectados, extraídos e reconstruídos de forma satisfatória, com precisão dos resultados chegando a mais de 70% Xu et al. (2018). Esse trabalho mostra uma nova perspectiva de acompanhamento do progresso de uma obra utilizando uma ferramenta de aprendizado de máquina.

Vetrivel et al. (2015) buscou utilizar as nuvens de ponto em um contexto diferente dos outros estudos analisados até o momento, para avaliação dos danos em edifícios após um evento de desastre, pois alega que as nuvens fornecem características essenciais para uma análise detalhada nesse sentido. O interessante é que, nesse caso, os autores também utilizaram o RF e um algoritmo SVM (*Support Vector Machine*) para detectar as regiões danificadas, e o RF se mostrou mais útil ao identificar 95% dessas regiões (Vetrivel et al., 2015). Este estudo e o anterior mostram que o RF apresentou um bom desempenho para classificação de nuvens de pontos que foram utilizadas com objetivos diferentes, o que reforça a versatilidade do algoritmo.

Também foi gerado um mapa de correlação entre autores pelo VOSViewer, porém como a quantidade de autores não necessariamente relacionados era alta, optou-se por filtrar a maior rede de correlação existente dentre os 185 autores mapeados, resultando em 11 autores listados na Figura 9. Foram gerados 2 clusters que representam 2 artigos conectados por um dos autores, que serão tratados nesta seção.

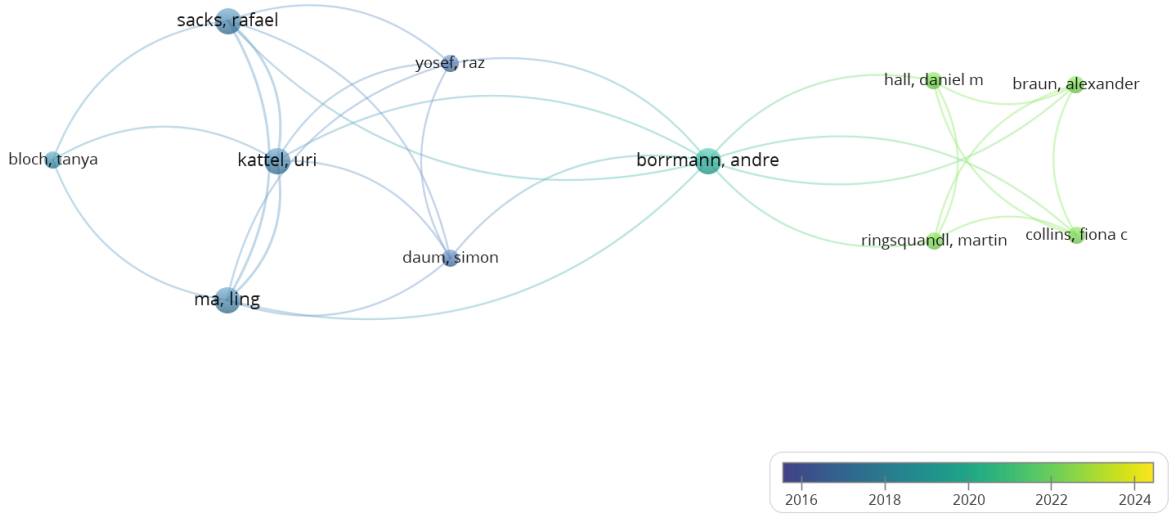

Figura 9: Maior rede de correlação entre autores.

Fonte: A autora.

Sacks et al. (2017) trata do software “SeeBIM” (*Semantic Enrichment Engine for* BIM), de enriquecimento BIM, que incorpora informações a modelos BIM por meio de um modelo de construção existente, bancos de dados ou conjuntos de regras. Nesse caso, as nuvens de pontos foram utilizadas apenas para servir como base de uma modelagem manual de uma estrutura complexa a ser utilizada no software. O autor menciona que os resultados de modelos BIM obtidos automaticamente por meio de nuvens de pontos não são semanticamente ricos e que normalmente há deficiência de informações sobre a identificação dos objetos e outros dados alfanuméricos (Sacks et al., 2017).

Collins et al. (2022) trazem um algoritmo geométrico que auxilia na classificação de formas de nuvens de pontos para correção semântica de modelos BIM, pois alegam que nos modelos BIM há frequentemente elementos mal classificados. Complementando o que foi observado no estudo feito por Sacks et al. (2017), Collins et al. (2022) acreditam que a identificação automatizada de elementos geométricos mal classificados e elementos aproximados/incertos deve estabelecer a base para a recuperação automática do modelo e reduzir a perda de informações ao longo do processo BIM.

Também foi elaborado um mapa de correlação entre as referências utilizadas nos trabalhos, que pode ser visto na Figura 10.

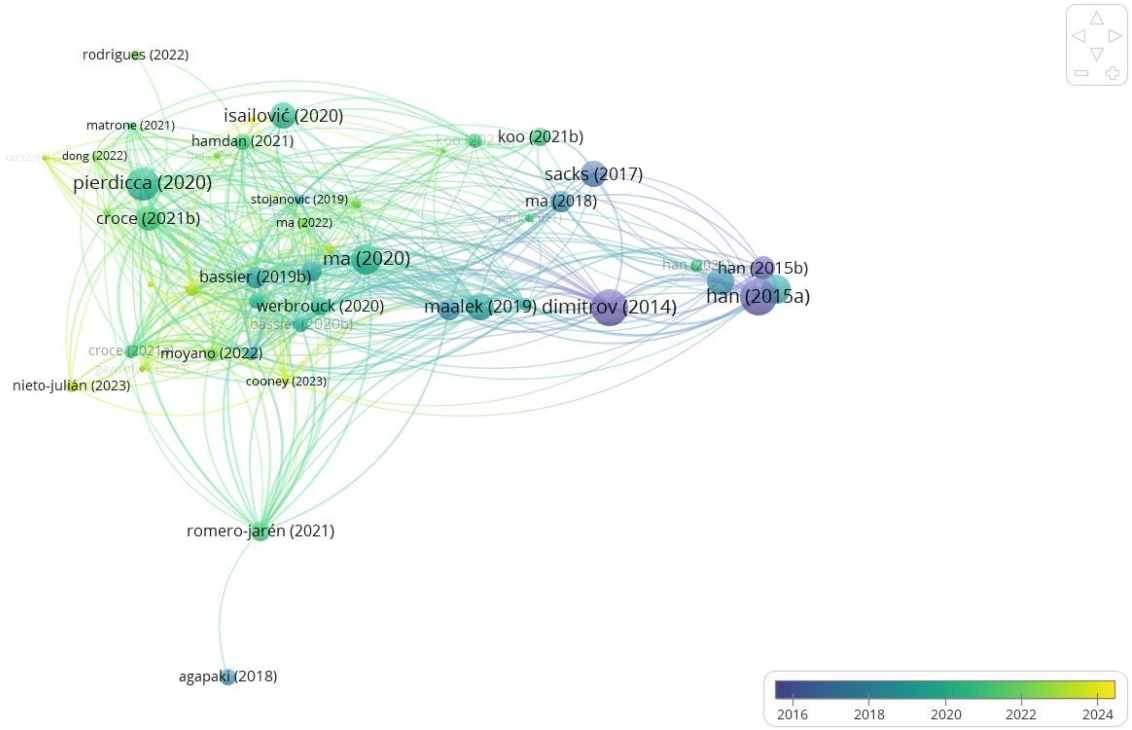

Figura 10: Rede de correlação entre referências utilizadas.

Fonte: A autora.

Romero-Jarén e Arranz (2021) trazem um método para segmentar, classificar e modelar automaticamente nuvens de pontos adquiridas por escaneamento a laser, capaz de gerar superfícies 3D precisas de elementos de construção de ambientes internos. Os autores Romero-Jarén e Arranz (2021) ressaltam que as superfícies criadas podem ser exportadas para um modelo BIM, porém a conversão para um modelo IFC não fez parte do escopo do trabalho.

Agapaki et al. (2018), que no mapa aparece em um cluster isolado, se relaciona ao estudo anterior por abordar o custo da modelagem de objetos industriais (IO) a partir de nuvens de pontos. Foi observado que as horas gastas com a modelagem de objetos cilíndricos, como tubulações, foram reduzidas em 64% (Agapaki et al., 2018).

Rodrigues et al. (2022) que também aparece um pouco deslocado dos demais trabalhos, abordou o *Deep Learning* para classificação do estado de degradação de edifícios, possuindo forte relação com Bahreini e Hammad (2024) citado anteriormente, apesar de não possuírem uma ligação forte por meio de suas referências. Os dois trabalhos foram capazes de exportar patologias reconhecidas por meio do *Deep Learning* para modelos BIM.

Os trabalhos mais recentes que aparecem no mapa tendem a focar em H-BIM, com investimento em tecnologias para identificação e classificação do patrimônio histórico (Cooney et al., 2023; Croce et al., 2023; Nieto-Julián et al., 2023). Croce et al. (2023) também utiliza do algoritmo RF na etapa de segmentação de nuvens de pontos, em que ele foi treinado a partir de dados de covariância e outras características presentes no *dataset* utilizado, obtendo como resultado do classificador uma nuvem de pontos que pode ser diferenciada de acordo com a distinção dos elementos arquitetônicos que a compõe.

Com a elaboração da análise bibliométrica, foi possível observar os mais diversos campos de aplicação de nuvens de pontos dentro do contexto BIM e de que forma a classificação das nuvens de pontos têm sido usadas para facilitar processos. Além disso, é observada a necessidade de desenvolvimento de estudos focados no processo de extração de informação semântica de nuvens de pontos, tendo em vista que vários trabalhos relatam o desafio no processo *Scan-to-BIM* de forma automática.

3 METODOLOGIA

Nesta seção serão descritos os passos metodológicos e ferramentas utilizadas para condução da pesquisa. Para elaboração do estudo, foi definida uma metodologia de abordagem quantitativa, onde foram analisados os resultados obtidos a partir do aprendizado de máquina para avaliar a eficiência e o desempenho da utilização do algoritmo escolhido. O fluxograma da pesquisa pode ser observado na Figura 11.

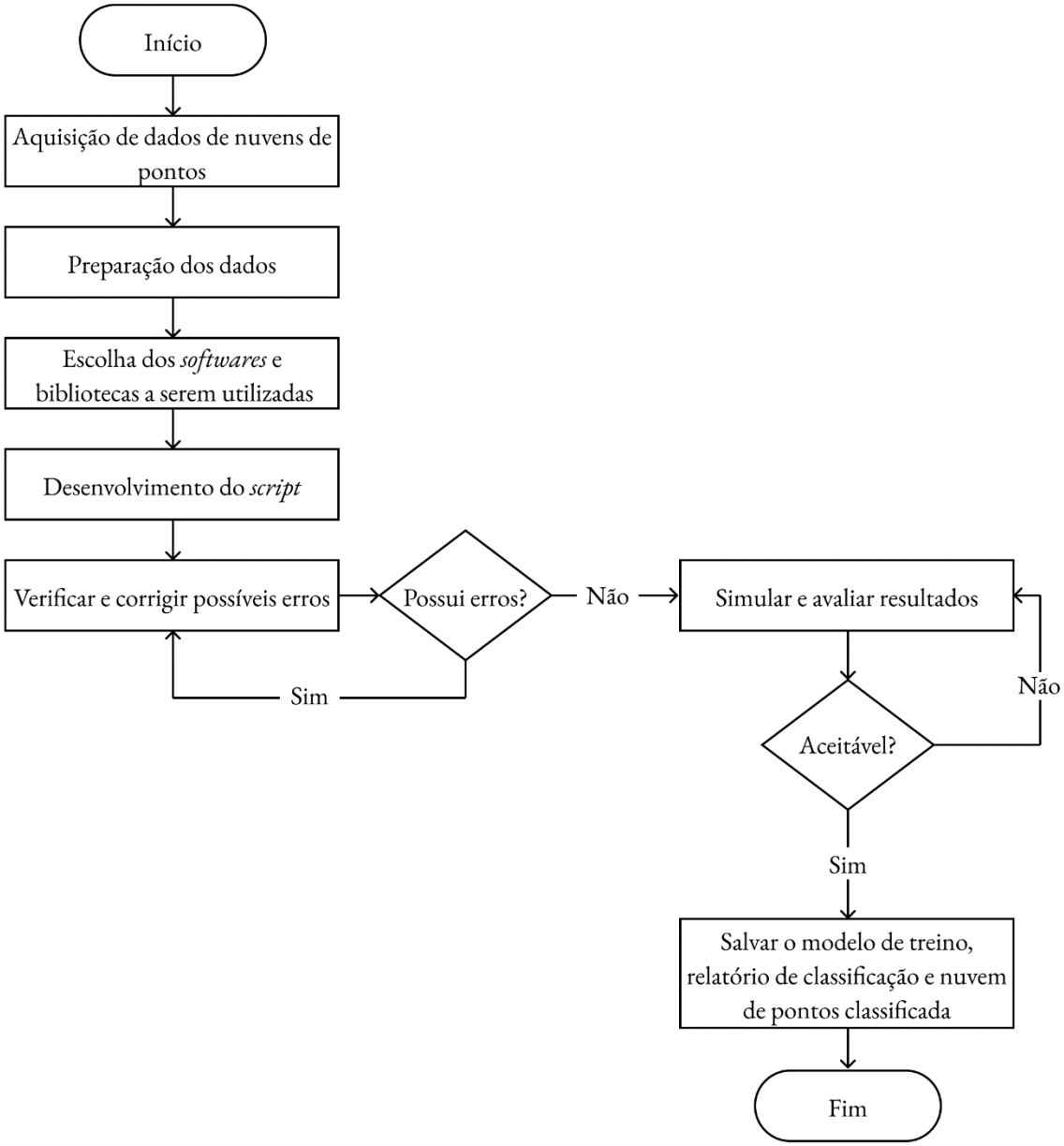

Figura 11: Fluxograma da pesquisa.

Fonte: A autora.

3.1. AQUISIÇÃO DOS DADOS

Para aquisição dos dados necessários para o treinamento do modelo, foi utilizado o GeoSampa, uma plataforma da cidade de São Paulo que permite acesso a mapas oficiais de forma gratuita, com dados georreferenciados da cidade. Nessa plataforma estão disponíveis dados de *LiDAR* aéreo de todo o município, separados por quadrículas com códigos diferentes, com nomenclatura estilo MDS_color_XXXX-XXX.

Para desenvolvimento do algoritmo, foram escolhidas as nuvens de pontos do GeoSampa da opção MDS 2017, que possibilita o *download* de arquivos *.laz* e *.laz(color)*, além de permitir visualizar os metadados do levantamento e visualizar as nuvens de forma online. No caso dessa pesquisa, foram utilizadas as nuvens em formato *.laz(color)*, que possuem informações adicionais de cor para cada um dos pontos.

Dada a diversidade de morfologias das nuvens de pontos de toda a cidade de São Paulo, foram escolhidas nuvens de pontos que, visualmente, abrangessem a maior quantidade de cenários possível, com presença equilibrada das classificações que compõe o objetivo desse trabalho. Para isso, foram escolhidas nuvens que possuíssem construções variadas, com diferentes cores de telhados; áreas com vegetação densa e áreas com pouca vegetação; e terreno no geral, especialmente rodovias e estradas.

3.2. PREPARAÇÃO DOS DADOS

As nuvens obtidas do GeoSampa possuem uma classificação nativa composta por 13 valores que representam diferentes classificações. Optou-se por trabalhar com as nuvens de pontos em formato *.xyz* dada a facilidade de manipular os dados como arquivos de texto, portanto os arquivos em formato *.laz(color)* foram convertidos para *.xyz* por meio do software de processamento de nuvens de pontos, o *CloudCompare*.

As nuvens nativas do *GeoSampa* possuem uma estrutura de texto semelhante a uma planilha quando no formato *.xyz*, onde a primeira linha é referente ao cabeçalho da nuvem de pontos e cada uma das colunas é uma informação diferente, provenientes do equipamento utilizado no escaneamento por *LiDAR*. As colunas do arquivo são compostas pelo conteúdo que pode ser visto no Quadro 2.

Quadro 2: Morfologia das nuvens de pontos do GeoSampa.

Conteúdo	Coluna
Coordenadas de cada ponto	X
	Y
	Z
Informações sobre as cores de cada ponto	R
	G
	B
Origem do ponto	PointSourceId
Informações definidas pelo usuário do equipamento	UserData
Ângulo de varredura do sensor	ScanAngleRank
Sinaliza se o ponto está próximo à borda da linha do voo	EdgeOfFlightLine
Direção da varredura (ida ou volta)	ScanDirectionFlag
Informa quantas vezes o laser atingiu o objeto	NumberOfReturns
	ReturnNumber
Tempo de captura do ponto	GpsTime
Intensidade do retorno do laser	Intensity
Classificação do tipo de ponto (terreno, vegetação, construção etc).	*Classification*

Fonte: A autora.

Para o estudo desenvolvido e dada a necessidade de redução do processamento computacional, foram mantidas apenas as colunas referentes às coordenadas (X, Y, Z), as cores dos pontos (R, G, B) e a coluna *Classification*, que possui a informação da classificação do tipo de objeto presente na nuvem de pontos.

No Quadro 3 estão descritas as classificações das nuvens possíveis presentes da coluna *Classification* e o que cada uma delas representa no *GeoSampa*. Para o desenvolvimento do algoritmo, foram foco desse estudo apenas as camadas referentes ao terreno, à vegetação e às construções.

Quadro 3: Classificação nativa do GeoSampa.

Número	Classe	Conteúdo
0	Never classified	Nunca classificado
1	Unclassified	Pontos não classificados
2	*Ground*	Terreno
3	Low *vegetation*	Vegetação baixa
4	Medium *vegetation*	Vegetação média
5	High *vegetation*	Vegetação alta
6	Building	Construções
7	Low point (noise)	-
8	Key-point	-
9	Water	Corpos d'água
10	Overlap	-
19	Other features (19)	Rede de energia e outros
20	Other features (20)	Terreno e outros

Fonte: A autora.

3.3. ESPECIFICAÇÕES DOS EQUIPAMENTOS E SOFTWARES UTILIZADOS

Para o desenvolvimento da pesquisa foi utilizado principalmente um notebook com as seguintes configurações:

-Processador: 11th Gen Intel® Core™ i7-11800H @ 2.30 GHz;

-RAM Instalada: 16,0 GB;

-Tipo de Sistema: Sistema Operacional de 64 bits, processador baseado em x64.

Como apoio, também foi utilizado um C*luster* do Centro de Informática da Universidade Federal de Pernambuco, o Apuana, com as seguintes especificações:

- Nome do modelo: Intel® Xeon® Gold 5318Y CPU @ 2.10GHz;
- Família da CPU: 6;
- Núcleos por soquete: 24;
- Threads por núcleo: 2;
- GPU: Nvidia RTX 3090 24 GB e A100 80 GB;
- Arquitetura: x86_64 (64 bits);
- Tamanho máximo de memória (de acordo com o tipo de memória) - 6 TB;
- Tipos de memória - DDR4-2933;
- Velocidade máxima de memória - 2933 MHz.

Para desenvolvimento do *script*, foi utilizado o *software* Visual Studio Code, utilizando linguagem Python, com as seguintes bibliotecas:

- *pickle*: para salvar o modelo treinado;

- *time*: para contabilizar o tempo de execução do código;

- *pandas*: para manipular a base de dados;

- *scikit-learn*: biblioteca de aprendizado de máquina, que possui diversos algoritmos, como o Random Forest;

- *matplotlib*: para criação de gráficos.

Dentro da biblioteca scikit-learn, foram importados os seguintes módulos:

- *RandomForestClassifier*: algoritmo de aprendizado de máquina baseado em árvores de decisão;

- *train_test_split*: função que divide os dados em conjuntos de treinamento e teste, para avaliar o desempenho do modelo;

- *classification_report*: gera métricas de avaliação para os modelos de classificação, fornecendo dados como precisão, recall e F1-score;

- *StandardScaler:* faz a normalização dos dados.

Além disso, também foi utilizado o software *CloudCompare,* que permite processar nuvens de pontos, fazer conversão do tipo de arquivo, filtrar classificações das nuvens de pontos e diversas outras funções. No caso do estudo, o *CloudCompare* foi utilizado tanto para fazer a conversão de arquivos *.laz* para *.xyz* como para visualizar e comparar visualmente os resultados obtidos pelo script.

4 *SCRIPT* DE CLASSIFICAÇÃO DE NUVENS DE PONTOS

Nesta seção são apresentados os pseudocódigos que representam o *script* de classificação de nuvens elaborado, independente da sintaxe da linguagem de programação utilizada, para um melhor entendimento do algoritmo desenvolvido.

4.1. ESTRUTURA DO SCRIPT

A estrutura do *script* foi desenvolvida a partir de 8 etapas principais, que estão descritas nessa seção. No início do algoritmo, são carregados os dados a serem utilizados no *script*, isso inclui tanto os dados do conjunto de treinamento pré-classificados que serão utilizados para construir o modelo de aprendizado, quanto o conjunto de dados originalmente não classificado a ser classificado pelo algoritmo.

Em uma segunda etapa, há o mapeamento das classes, que associa os valores numéricos originais referentes à coluna "*Classification*" aos rótulos que serão utilizados, que são definidos pelo usuário de acordo com a necessidade de aprendizado. No caso deste estudo, foram rotuladas apenas as colunas referentes ao terreno, vegetação e construções e, todas os valores que correspondesse a outros rótulos, seriam tratados como uma camada unificada de "outros".

Na terceira etapa do algoritmo, o conjunto de dados importado na primeira etapa é dividido em dois subconjuntos, um para teste e um para treino. Nesse caso, foi definido que 20% dos dados seriam utilizados para teste, enquanto 80% seria utilizado para treinamento do modelo. Já na quarta etapa o conjunto de dados de treino e teste é normalizado, para garantir que haja equilíbrio entre as variáveis de aprendizado do modelo.

Na quinta etapa é treinado e criado o modelo de aprendizado a partir do conjunto de dados utilizado, por meio da biblioteca do Random Forest existente na linguagem Python, onde o usuário define a quantidade de estimadores, que representam a quantidade de árvores de decisão que serão criadas no modelo. No caso deste estudo, foram utilizadas 100 árvores, que é o valor *default* da biblioteca, porém, dada a baixa quantidade de características a serem avaliadas no processo de aprendizado (X, Y, Z, R, G, B), acredita-se que o algoritmo também teria um bom desempenho com uma quantidade menor de árvores. Utilizando uma quantidade maior de árvores, o processamento computacional tende a ser maior, mas isso não gera nenhum tipo de prejuízo no resultado a ser obtido.

Na sexta e na sétima etapa, o modelo criado faz previsões no conjunto de dados de teste para avaliar as métricas de aprendizado e, em seguida, gera um relatório de aprendizado com

esses valores, respectivamente. Por fim, a nuvem de pontos originalmente não classificada é classificada a partir do modelo gerado e é exportada com as classes previstas pelo algoritmo. A representação da estrutura do algoritmo pode ser vista a seguir:

1) Carregamento dos dados que serão utilizados no script:
    a. Importa os arquivos de treinamento (*dataset*) e o arquivo da nuvem de pontos originalmente não classificada em estruturas de dados adequadas.
2) Mapeamento das classes:
    a. Cria um dicionário de mapeamento para associar os valores numéricos da coluna "*Classificatio*n" aos rótulos desejados (por exemplo, 2 equivale a "*ground*", 6 equivale a "*buildings*", 5 equivale a "*vegetation*" e os demais números serão definidos como "*others*");
    b. Aplica esse mapeamento à coluna "*Classificatio*n" do conjunto de treinamento.
3) Divisão dos dados:
    a. Divide os dados de treinamento em conjuntos de treinamento e teste usando a função *train_test_split*;
    b. São definidas as porcentagens de pontos que serão utilizadas para teste e treino (exemplo: 20% dos dados carregados serão de teste e 80% serão para treino).
4) Normalização dos dados:
    a. Usa a função do *StandardScaler* para normalizar os dados de treinamento e teste.
5) Criação e treinamento do modelo:
    a. Cria um modelo *RandomForestClassifier* com determinado número de estimadores que representam o número de árvores de decisão que o modelo cria;
    b. Treina o modelo nos dados normalizados de treinamento.
6) Previsões:
    a. Faz previsões no conjunto de teste normalizado.
7) Avaliação o modelo:
    a. Imprime um relatório de classificação com métricas como precisão, recall e *F1-score*;
8) Exportação de arquivos com resultados:

a. Salva os dados não classificados agora classificados em um novo arquivo .xyz.

4.2. PSEUDOCÓDIGO DO SCRIPT DESENVOLVIDO

Neste tópico é apresentado o algoritmo de classificação de nuvens de pontos mais detalhadamente por meio de um pseudocódigo, isto é, uma forma genérica de representar o algoritmo sem que haja a sintaxe de uma linguagem de programação. Na Etapa 1, o algoritmo importa as bibliotecas mencionadas anteriormente e os dados de treinamento. Na Etapa 2, há um mapeamento das classes existentes na coluna "*Classification*", em que são designadas as classes desejadas para cada um dos valores na coluna. Caso o valor da classe de determinado ponto seja diferente dos determinados, ele recebe "*others*". Um detalhamento maior da Etapa 2 é feita no Item 4.3

Na Etapa 3, os dados são divididos em conjuntos de treino e teste e, na Etapa 4, esses dados são normalizados. Na Etapa 6 são feitas previsões no conjunto de dados e, na Etapa 7, as previsões realizadas são avaliadas e é gerado um relatório com as métricas de avaliação. A partir do aprendizado realizado, na Etapa 8 são feitas previsões no conjunto de dados originalmente não classificados. Na Etapa 9, as classes determinadas na Etapa 2 agora irão receber valores numéricos escolhidos pelo usuário, para diferenciar as classes. O conjunto de dados incialmente não classificado agora é salvo como um conjunto de dados classificado na Etapa 10. O Algoritmo 2 pode ser visto a seguir.

Algoritmo 2: Treinamento e classificação de nuvens

Step 1: Import necessary libraries and load input data (training data and unclassified point cloud)

Step 2: Map 'Classification' column to desired values

for each row in data frame:

if 'Classification' is null, **return** 'Classification' ← 'others'

else, map 'Classification' to desired values

Step 3: Split data into training and test sets (x_train; x_test; y_train; y_test)

Step 4: Normalize training and test data

x_test_scaled, x_train_scaled ← Normalization applied to x_train and x_test

Step 5: Random Forest implementation to create and train the model

model ← Create RandomForestClassifier with n_estimators = 100

Train model with x_train_scaled and y_train

Step 6: Make predictions on normalized test set

y_pred ← model.predict(x_test_scaled)

Step 7: Calculate and print the classification report

print classification_report(y_test, y_pred)

if the results are satisfactory, then save trained model

Step 8: Normalize unclassified data and make the predictions

dataframe_unclassified_scaled ← Normalize dataframe_unclassified

predictions ← model.predict(dataframe_unclassified_scaled)

Step 9: Map predictions back to corresponding numbers and is added as a new column to unclassified data

new_column of dataframef_unclassifed ← predictions_mapped (Map predictions to corresponding numbers)

Step 10: Save unclassified now classified data

Save dataframe_unclassified to a new .xyz file

4.3. MAPEAMENTO DAS COLUNAS

A etapa “2) Mapear as Classes” do pseudocódigo mencionado se refere ao mapeamento das colunas apresentadas no Quadro 3. O script mapeia os valores numéricos da coluna “*Classificatio*n” dos dados de treino e os traduz para valores textuais, como “*ground*”, “*buildings*” e “*vegetation*”. Isso permite que mais de uma camada seja utilizada como uma única classe, exemplo: os valores 3, 4, 5 da coluna *Classificatio*n (que originalmente se referem à “low *vegetation*”, “medium *vegetation*” e “high *vegetation*”, respectivamente) podem ser transformados em uma classe unificada de “*vegetation*”.

Posteriormente no script esse dicionário criado é revertido novamente para valores numéricos, tendo em vista que o *CloudCompare* lê os arquivos com valores numéricos, e o usuário pode escolher números de sua preferência para visualizar como as novas classificações. Para um melhor entendimento, foi elaborado um esquema de como funciona esse “tradutor” de classes para valores numéricos, que pode ser observado na Figura 12.

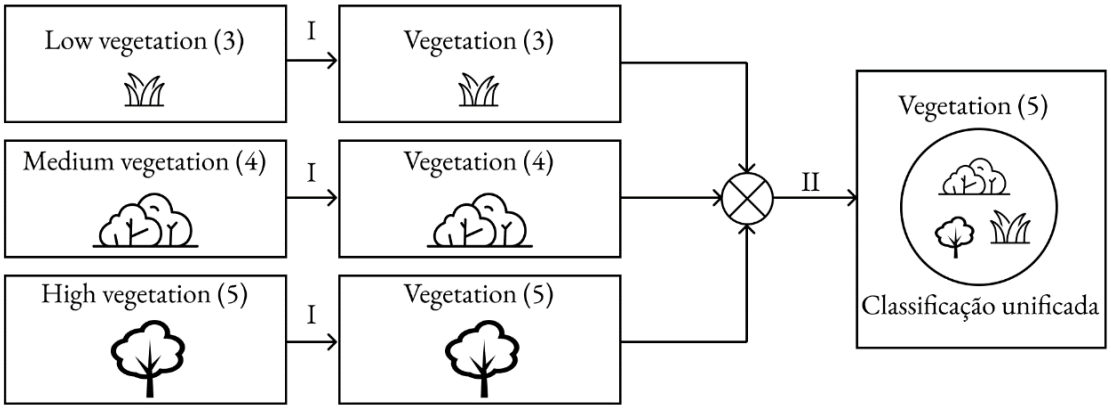

Figura 12: Ilustração do processo de mapeamento das colunas.

Fonte: A autora.

Onde:

I. Os valores na coluna *Classificatio*n referentes a (3), (4) e (5) recebem a nova classificação "*vegetation*";

II. Todos os valores que contêm a classificação "*vegetation*" são substituídos por um novo valor numérico da escolha do usuário. Exemplo: (5).

5 ANÁLISE DE RESULTADOS

Os testes iniciais no script foram realizados com um *dataset* de apenas uma nuvem de pontos, para avaliar como se comportaria o algoritmo com menos dados e, a partir daí, ir escalando a quantidade de nuvens no *dataset*. As nuvens de pontos utilizadas têm, em média, 300 Mb em formato .xyz, já com a preparação dos dados aplicada, com exclusão das colunas que não serão utilizadas pelo algoritmo. Para cada um dos resultados, foi gerado um relatório de avaliação do aprendizado, exportado um modelo de aprendizado em formato .pkl e contabilizado o tempo que o script demorou para gerar os resultados.

5.1. INFLUÊNCIA DA QUALIDADE DOS DADOS DE ENTRADA NO RESULTADOS

Para um conjunto de dados com 10.638.403 pontos, sendo 80% dos dados destinados para treino e 20% para teste, foi gerado um relatório de treinamento que pode ser observado na Tabela 1. Para o mapeamento das colunas, foi determinado que 2: '*ground*', 5: '*vegetation*', 6: '*buildings*' e qualquer outro valor presente deveria ser classificado como ‘*others*’.

Tabela 1: Relatório de aprendizado com apenas uma nuvem no *dataset*.

	Precision	Recall	F1-score
Buildings	0,99	0,99	0,99
Ground	0,25	0,01	0,03
Others	0,96	0,99	0,97
Vegetation	0,98	0,96	0,97
Accuracy			0,98
Macro Avg	0,79	0,74	0,74
Weighted Avg	0,97	0,98	0,97

Fonte: A autora.

No Quadro 4 estão os códigos das nuvens do GeoSampa que foram utilizadas para treinamento e teste, bem como a nuvem que foi utilizada para classificação. Nesse primeiro teste, buscou-se utilizar nuvens que possuíssem equilíbrio entre as 3 classes principais a serem definidas.

Quadro 4: Nuvens utilizadas no Teste 1.

Nuvens utilizadas para treinamento e teste	MDS_color_4311-244
Nuvem a ser classificada	MDS_color_3326-433

Fonte: A autora.

Na Figura 13 está indicada a nuvem MDS_color_4311-244 em formato RGB (à esquerda) e sua classificação original do GeoSampa (à direita). Neste caso, as construções estão indicadas como verde claro; as vegetações em verde escuro; e o terreno como vermelho.

Figura 13: *Dataset* de apenas uma nuvem e sua classificação nativa do GeoSampa (terreno em vermelho; construções em verde claro; e vegetação em verde escuro).

Fonte: A autora.

Já na Figura 14, está a nuvem MDS_color_3326-433 em RGB (à esquerda) e classificada pelo script (à direita). A coluna "*Classificatio*n" da MDS_color_3326-433 retornou apenas as classes "*buildings*", "*vegetation*" e "*others*", nenhum dos pontos foi classificado como "*ground*".

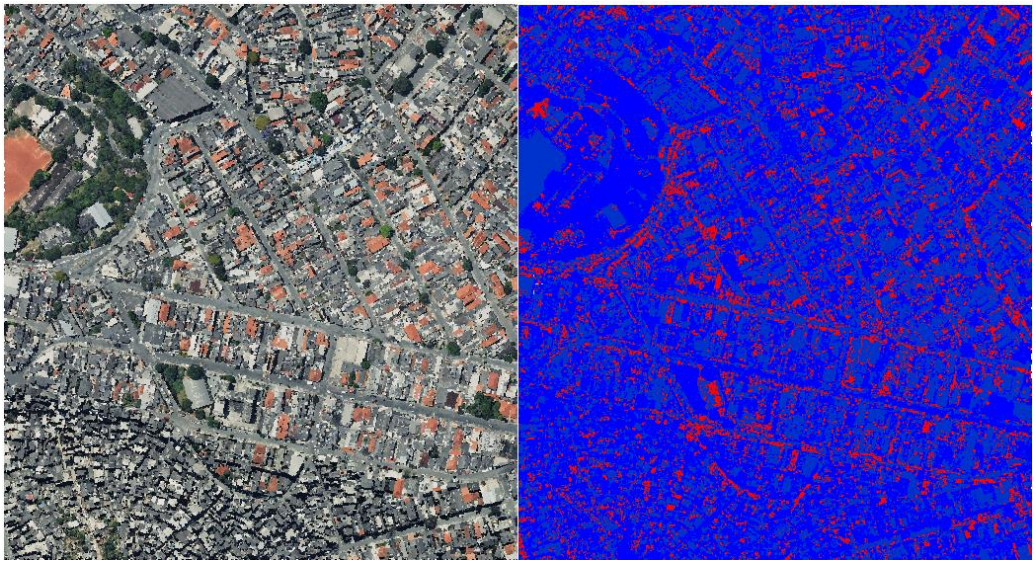

Figura 14: Classificação inicial da nuvem MDS_color_3326-433 (vegetação em azul mais claro; construção em azul escuro; e "outros" em vermelho).

Fonte: A autora.

Na Figura 15 está indicada a visualização de cada uma das classes de forma separada, para um melhor entendimento dos resultados obtidos, sendo ‘*vegetation*’, ‘*buildings*’ e ‘*others*’, da esquerda para a direita, respectivamente.

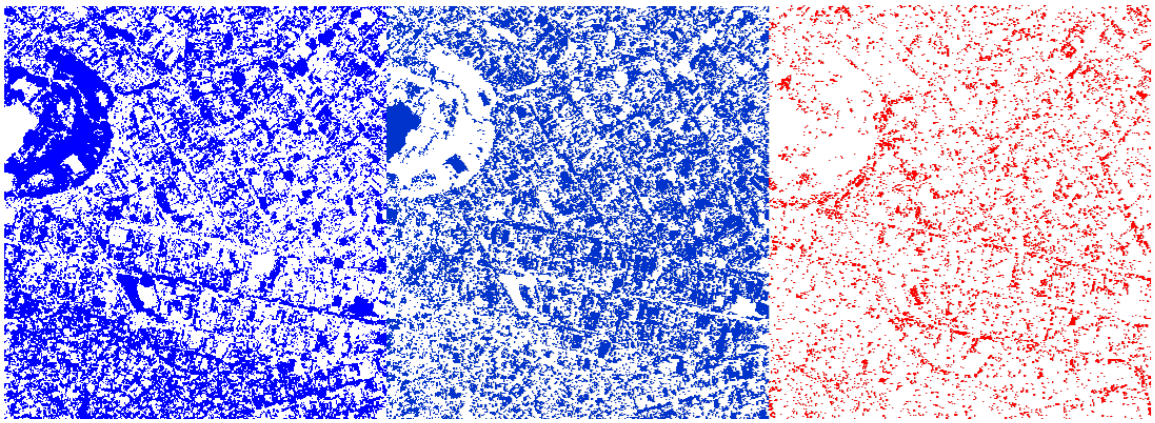

Figura 15: Visualização de cada uma das classes geradas na MDS_color_3326-433 (vegetação, construção e outros, respectivamente).

Fonte: A autora.

Dado que o aprendizado da camada “*ground*” no relatório de classificação foi baixo, foi observado que a camada de classificação original do GeoSampa referente ao terreno é bastante ruidosa, ou seja, seus pontos são dispersos e pouco densos. Por esse motivo, foi realizada uma mudança apenas na fase de mapeamento das colunas no script, para avaliar os efeitos que ignorar a classificação “*ground*” geraria; o *dataset* e a nuvem utilizada para classificação foram mantidos, conforme Quadro 4.

Nesse caso, para o mapeamento de colunas, foi determinado que 5: '*vegetation*', 6: '*buildings*' e qualquer outro valor presente deveria ser classificado como ‘*others*’, inclusive o ‘*ground*’. Os resultados desse teste podem ser observados na Figura 16. As diferenças mais significativas foram percebidas na classificação do ‘*others*’, que identificou menos pontos.

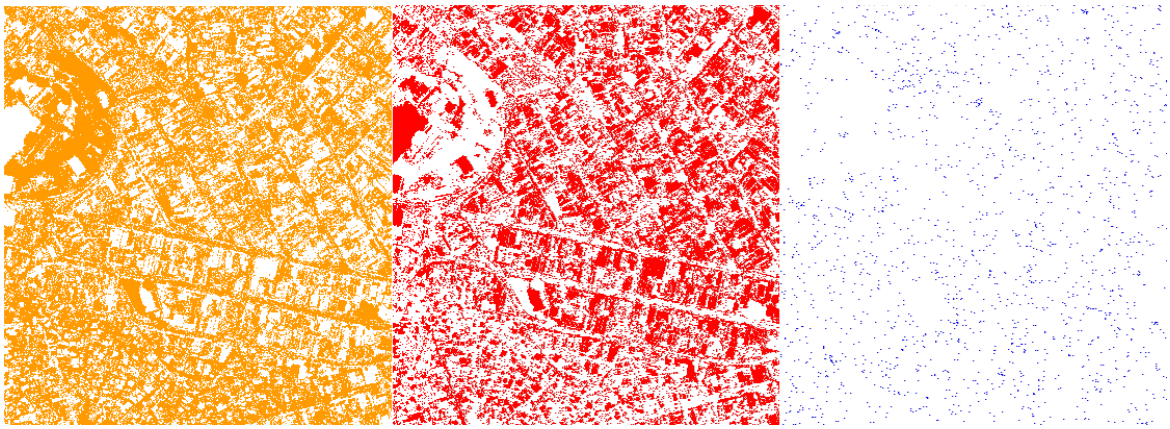

Figura 16: Visualização de cada uma das classes geradas na MDS_color_3326-433, sem considerar a classificação ‘*ground*’ (vegetação, construção e outros, respectivamente)..

Fonte: A autora.

Tabela 2: Relatório de aprendizado desconsiderando o '*ground*'.

	Precision	Recall	F1-score
Buildings	0,99	0,99	0,99
Others	0,99	0,99	0,99
Vegetation	0,98	0,96	0,97
Accuracy			0,99
Macro Avg	0,99	0,98	0,98
Weighted Avg	0,99	0,99	0,99

Fonte: A autora.

Apesar dos resultados do relatório indicarem uma maior precisão no aprendizado, percebe-se pela Figura 16 que incluir a classificação do '*ground*' com as classificações do '*others*' tornou a classificação do '*ground*' pior, visualmente falando. Além disso, a nuvem referente ao '*vegetation*' passou a incluir alguns pontos que representam ruas, que anteriormente estavam na nuvem do '*ground*'.

5.2. INFLUÊNCIA DO TAMANHO DO *DATASET*

De forma semelhante ao que foi feito no Item 5.1, foram analisados os efeitos que aumentar o *dataset* teriam na classificação final da nuvem, pois espera-se que quanto mais dados, mais eficiente o algoritmo se tornará. Inicialmente foram utilizadas nuvens de pontos de 5 localidades diferentes do GeoSampa, totalizando 19.398.396 pontos. o algoritmo foi alimentado com 5 nuvens de pontos com morfologias diferentes, buscando representar o maior número de configurações possíveis. As nuvens escolhidas no *dataset* mencionado estão especificadas no Quadro 5.

As 5 nuvens escolhidas para montar o *dataset* inicial foram focadas em representar: espaço urbano denso, com muitas casas e prédios; bairros residenciais com predominância de casas; regiões com massa d'água; e áreas de morro, com relevo acentuado.

Quadro 5: Nuvens utilizadas no *dataset*.

Código das nuvens no GeoSampa
MDS_color_3322-433
MDS_color_2324-452
MDS_color_4311-244
MDS_color_3336-224

Fonte: A autora.

A morfologia das nuvens pode ser observada na Figura 17, onde está a classificação de cada uma delas, sendo vegetação em amarelo, construções em vermelho e terreno em azul. A partir da Figura 17 é possível perceber a questão do ruído na classe azul referente ao '*ground*'.

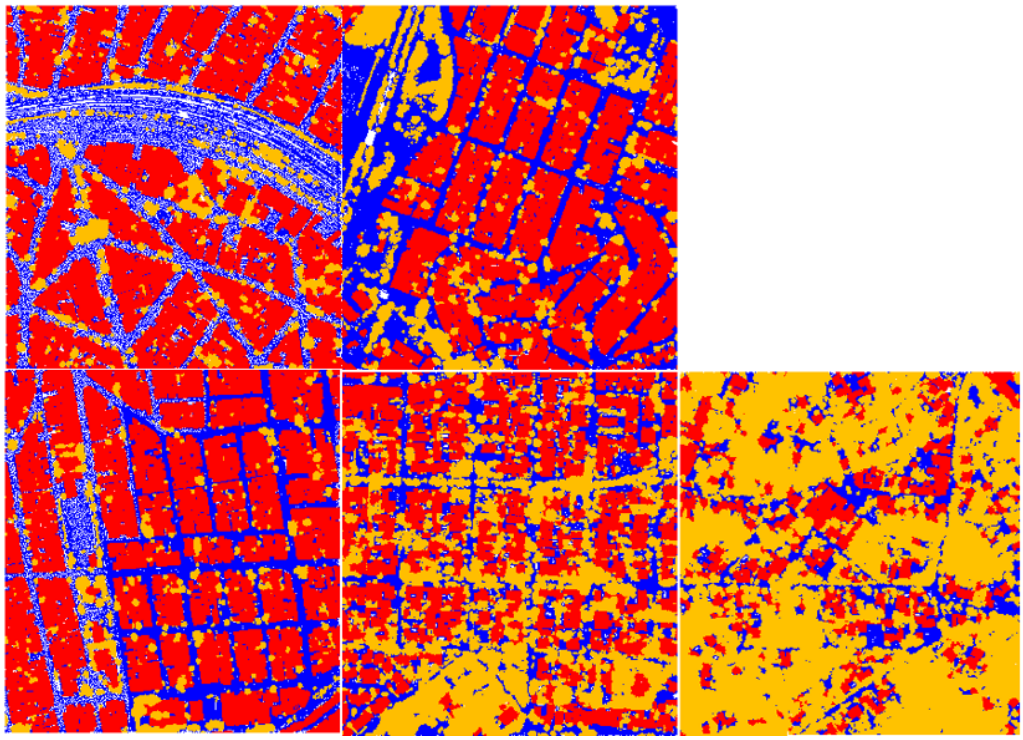

Figura 17: *Dataset* com 5 quadrantes (terreno em azul; construções em vermelho; e vegetação em amarelo).

Fonte: A autora.

O mesmo *dataset* classificado apresentado na Figura 17 pode ser observado na Figura 18 em RGB, ou seja, com as cores reais obtidas no levantamento realizado.

Figura 18: *Dataset* com 5 quadrantes em RGB.

Fonte: A autora.

A partir dos dados apresentados na Figura 17, foram realizados os mesmos testes para comparação. Inicialmente foi utilizado o mesmo mapeamento de colunas que diferenciava as classificações "*buildings*", "*vegetation*", "*ground*" e "others", foi obtido o relatório de aprendizado apresentado na Tabela 3.

Tabela 3: Relatório de aprendizado com um *dataset* de 5 nuvens.

	Precision	Recall	F1-score
Buildings	0,98	0,99	0,98
Ground	0,99	0,98	0,98
Others	0,98	0,96	0,97
Vegetation	0,98	0,97	0,97
Accuracy			0,98
Macro Avg	0,98	0,98	0,98
Weighted Avg	0,98	0,98	0,98

Fonte: A autora.

Como resultado, na Figura 19 é possível observar a nuvem original em RGB (à esquerda) e a nuvem classificada à direita, onde o algoritmo definiu apenas as classes "*vegetation*" (em amarelo) e "*buildings*" (em vermelho), não sendo nenhum dos pontos classificados como "*ground*" e nem "*others*".

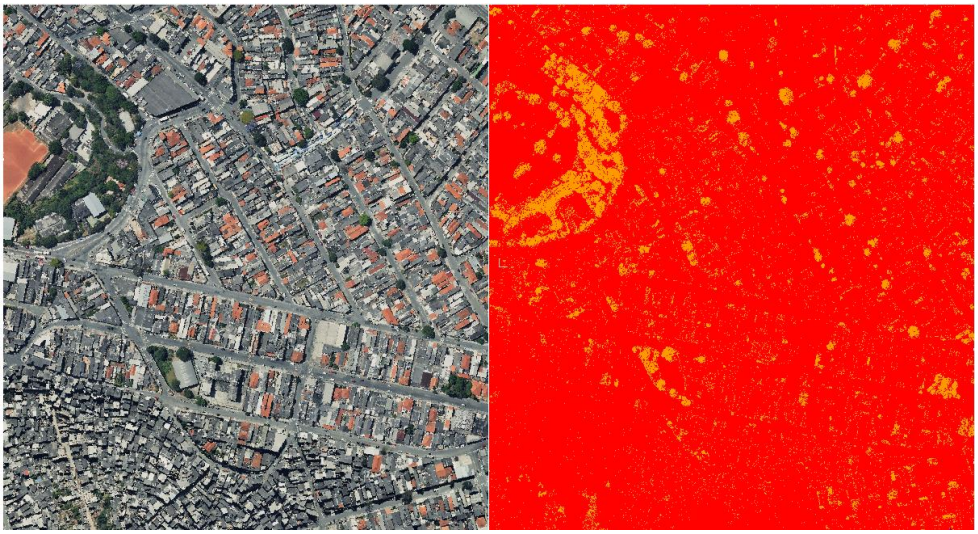

Figura 19: Nuvem original e nuvem classificada (vegetação em amarelo e construções em vermelho).

Fonte: A autora.

Por fim, o mesmo dataset apresentado na Figura 17 foi utilizado em um teste com um novo mapeamento das colunas, com a inclusão de mais um quadrante de pontos para aumentar o conjunto de dados, que pode ser visto na Figura 20. A intenção do teste era observar como se comportaria o aprendizado se a camada original do *ground* fosse unida com as camadas 19 e 20, referentes a "other features" (Quadro 3). Isso porque, visualmente, as camadas *others features* pareciam conter partes do terreno e eram mais densas que a camada *ground*, o que

levantou a hipótese de que o algoritmo pudesse classificar melhor o terreno caso houvesse essa alteração no mapeamento das classes.

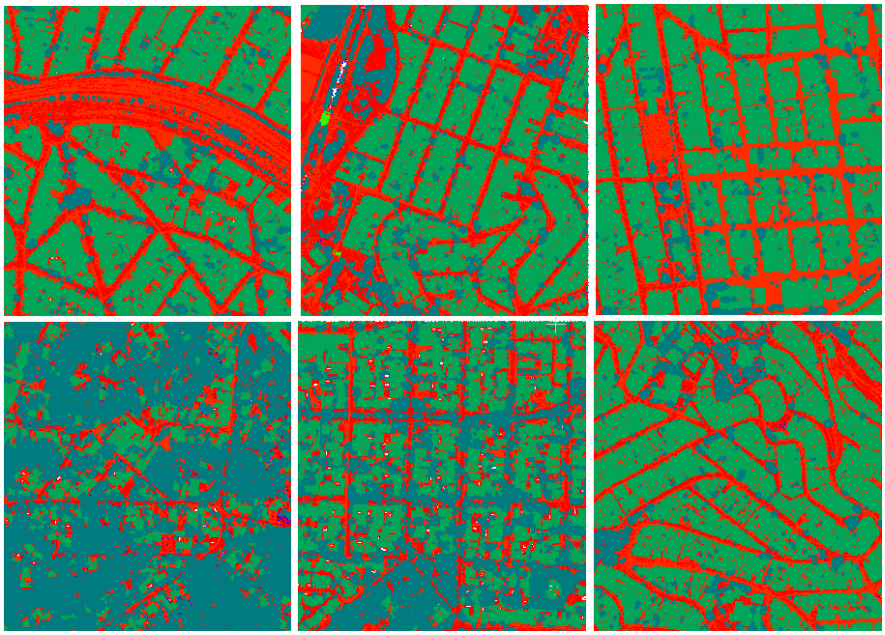

Figura 20: *Dataset* com 6 quadrantes (terreno em vermelho; construções em verde claro; e vegetação em verde escuro).

Fonte: A autora.

O mesmo *dataset* classificado apresentado na Figura 20 pode ser observado na Figura 21 em RGB, ou seja, com as cores reais obtidas no levantamento realizado.

Figura 21: *Dataset* com 6 quadrantes em RGB.

Fonte: A autora.

O resultado obtido a partir dessa mudança pode ser visto na Figura 22, em que a nuvem foi classificada com as 3 classificações desejadas, *ground* (em azul), *vegetation* (em amarelo) e *buildings* (em vermelho). Na parte inferior esquerda da imagem é possível observar uma mancha vermelha na classificação e, nessa região, há uma zona de topografia mais baixa na nuvem de pontos. Não foi identificado o motivo pelo qual a variação da topografia fez o algoritmo identificar essa região como uma área de construções.

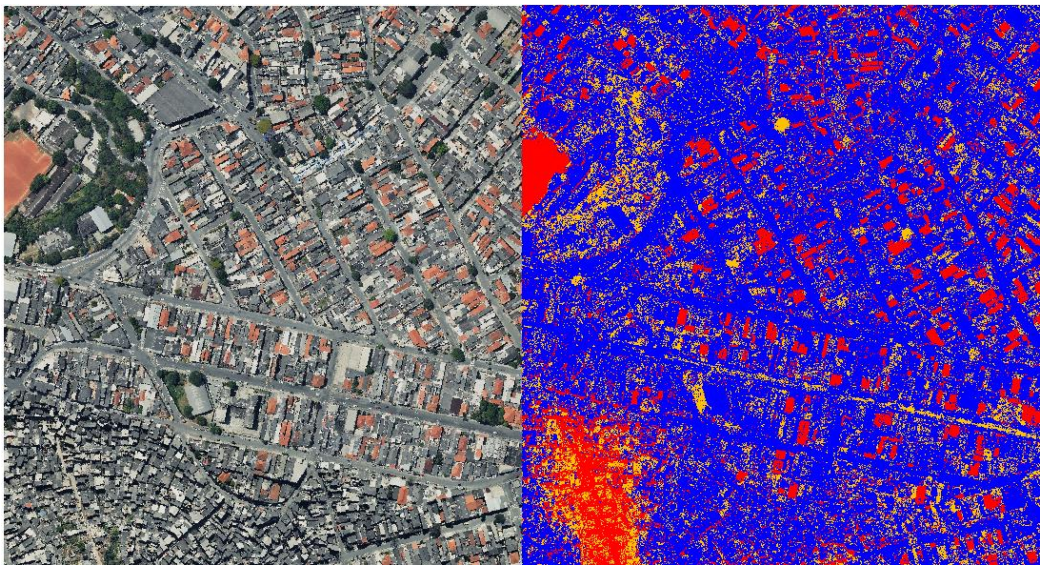

Figura 22: Resultado a partir da união do *ground* com *others* features (terreno em azul; vegetação em amarelo; construções em vermelho).

Fonte: A autora.

Na Figura 23 estão separadas as classes geradas para uma melhor observação. Apesar do ruído existente nas nuvens, é possível observar que a junção da classe *ground* às duas mencionadas anteriormente (19 e 20) gerou um melhor resultado que o aprendizado da classe *ground* de forma individual. Isso mostra que a qualidade nos dados de entrada interfere de forma significativa no aprendizado, pois, no resultado anterior, nenhum dos pontos foi classificado como terreno.

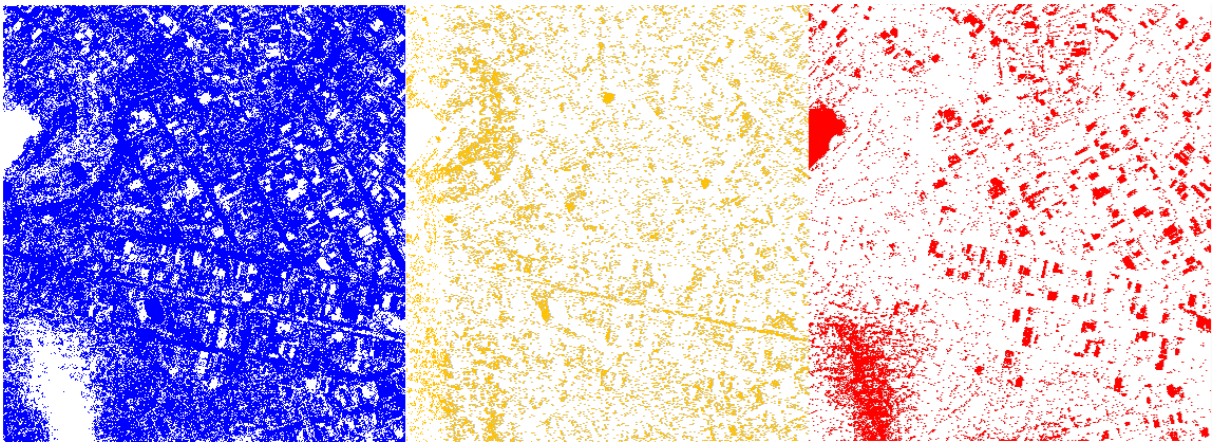

Figura 23: Classes divididas na nuvem classificada (terreno em azul; vegetação em amarelo; construções em vermelho).

Fonte: A autora.

A partir desse resultado obtido, foi gerada uma matriz de confusão, que é comumente utilizada para avaliar modelos de classificação em aprendizados de máquina supervisionado. Cada uma das linhas da matriz representa uma as instâncias de uma classe real, enquanto cada uma das colunas representa as instâncias de uma classe prevista. Nesse caso específico, a primeira linha representa *buildings*; a segunda representa *ground*; e a terceira representa *vegetation*.

A diagonal principal representa as previsões que foram feitas corretamente a partir do modelo de aprendizado. Fora da diagonal principal, estão os erros de previsão existentes, que se referem aos falsos negativos e aos falsos positivos. No caso da Figura 24, apenas há valores significativos na diagonal principal, que é o melhor cenário de aprendizado. Como a matriz gerada não está normalizada (Figura 24), ela apresenta os valores correspondentes ao número de pontos utilizados como teste durante o aprendizado (20% do conjunto total de pontos no *dataset*, isto é, ao somar todos os pontos que aparecem dentro da matriz, o valor será correspondente a 20% do total de pontos no *dataset* com 6 quadrantes.

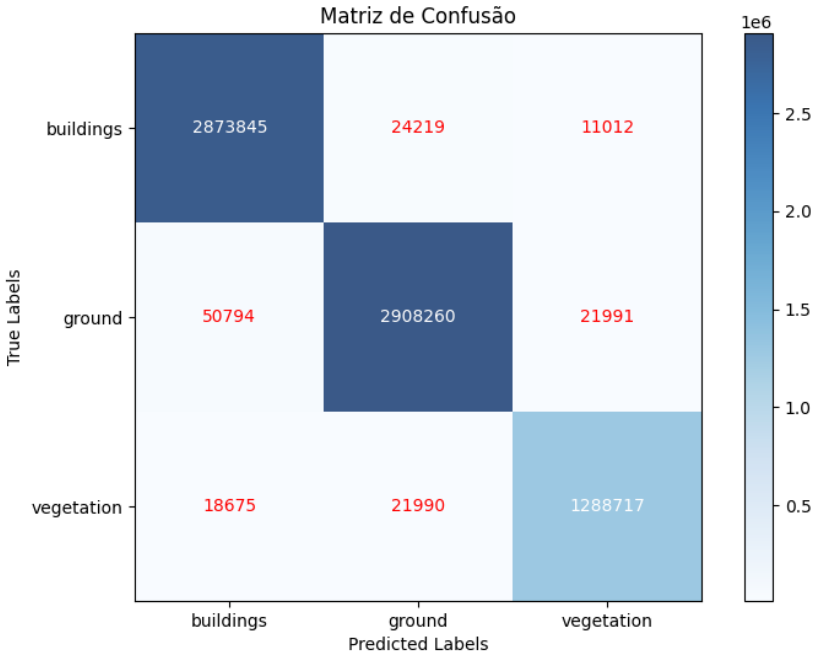

Figura 24: Matriz de confusão gerada.

Fonte: A autora.

Essa seria uma boa forma de avaliar os resultados se a distribuição de pontos de cada classe fosse uniforme, porém pode-se observar que a classe *vegetation* ficou um pouco mais clara do que as demais na escala de cores dessa matriz. Esse resultado passa a impressão de que o aprendizado da classe *vegetation* foi pior que a do *ground* e a do *buildings*, porém isso só significa que a quantidade de pontos de *vegetation* foi proporcionalmente menor na amostra.

Para resolver esse problema, a matriz de confusão foi normalizada, ou seja, os valores foram ajustados para que demonstrassem as porcentagens dos valores referentes às classes e não seus valores absolutos. Para isso, cada valor na matriz é dividido pelo total de pontos verdadeiros de suas respectivas classes.

Isso faz com que os valores passem a variar de 0 a 1 e, quanto mais próximo de 1 o valor na diagonal principal, isso significa que uma alta proporção de previsões para a classe foram realizadas de maneira correta. Por consequência, é desejado que os valores fora da diagonal principal sejam próximos a 0, pois isso indica que uma baixa proporção de previsões foi erroneamente atribuída a outras classes. O resultado da matriz normalizada pode ser visto na Figura 25.

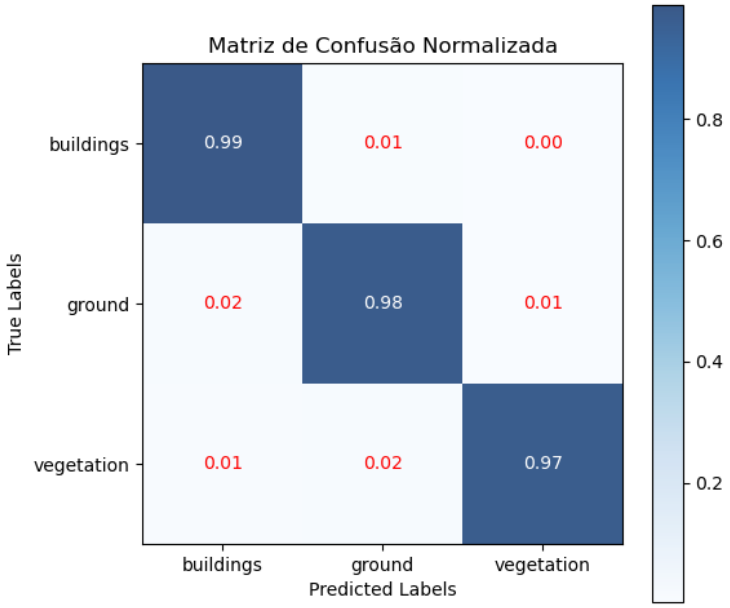

Figura 25: Matriz de confusão gerada.

Fonte: A autora.

É possível concluir que o desempenho do modelo foi satisfatório, com todas as 3 classes apresentando valores de previsão corretas próximos ao ideal, variando entre 97% a 99% de acerto. Outra questão a ser discutida é como a qualidade dos dados de entrada no algoritmo prejudicam a classificação, tendo em vista que em nenhum dos relatórios de aprendizado a camada "*ground*" teve boas métricas de forma individual, apenas quando foi unida às outras camadas. Um exemplo pode ser visto na Figura 26.

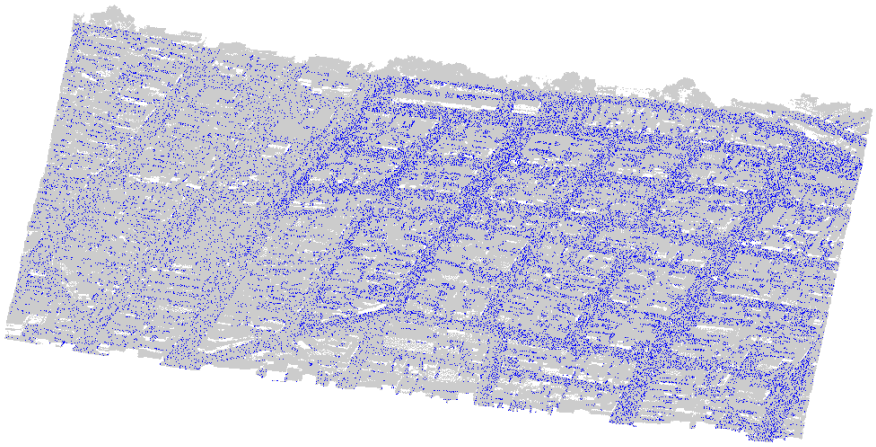

Figura 26: Baixa qualidade na classe "*ground*" de uma das nuvens do GeoSampa.

Fonte: A autora.

Acredita-se que as métricas baixas ligadas ao *ground* tenham ocorrido devido à baixa densidade dos pontos, pois avaliando cada camada original individualmente no *CloudCompare*,

a do “*ground*” apresentava o pior desempenho, conforme observado na Figura 26. Supõe-se que com dados de entrada com uma qualidade melhor devem tornar o aprendizado mais acurado, como ocorreu com as classes *buildings* e *vegetation*.

5.3. CLASSIFICAÇÃO DE NUVENS OBTIDAS POR FOTOGRAMETRIA

O script também foi testado em nuvens de pontos obtidas por fotogrametria, pois antes só havia sido testado em nuvens obtidas por *LiDAR*. Na Figura 28 pode-se observar a nuvem obtida por fotogrametria classificada pelo algoritmo. As partes em verde escuro referem-se à vegetação, o verde mais claro refere-se às construções e o vermelho refere-se a “outros”. Alguns pontos dispersos foram classificados como terreno (em azul), porém não é possível observá-los na imagem. Isso reforça que a classificação da camada “*ground*” foi prejudicada pela baixa qualidade dos dados de entrada.

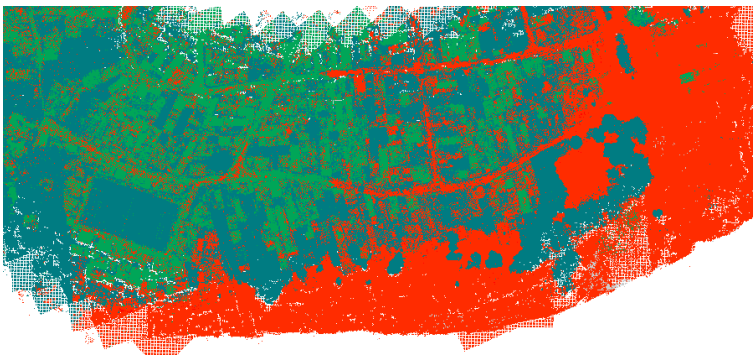

Figura 27: Nuvem de drone classificada (vegetação em verde escuro; construções em verde claro; terreno em vermelho).

Fonte: A autora.

Uma particularidade ser observada nesse resultado que é o padrão dos telhados no Brasil, onde há uma variedade de materiais utilizados, como telhas de barro, fibrocimento, telhas metálicas, PVC, entre outros. Acredita-se que isso possa confundir o aprendizado de algumas camadas, pois às vezes o telhado pode se confundir com o terreno por ter cores parecidas. O mesmo ocorreu na Figura 28, onde uma quadra de futebol com grama verde foi classificada como vegetação por ser verde.

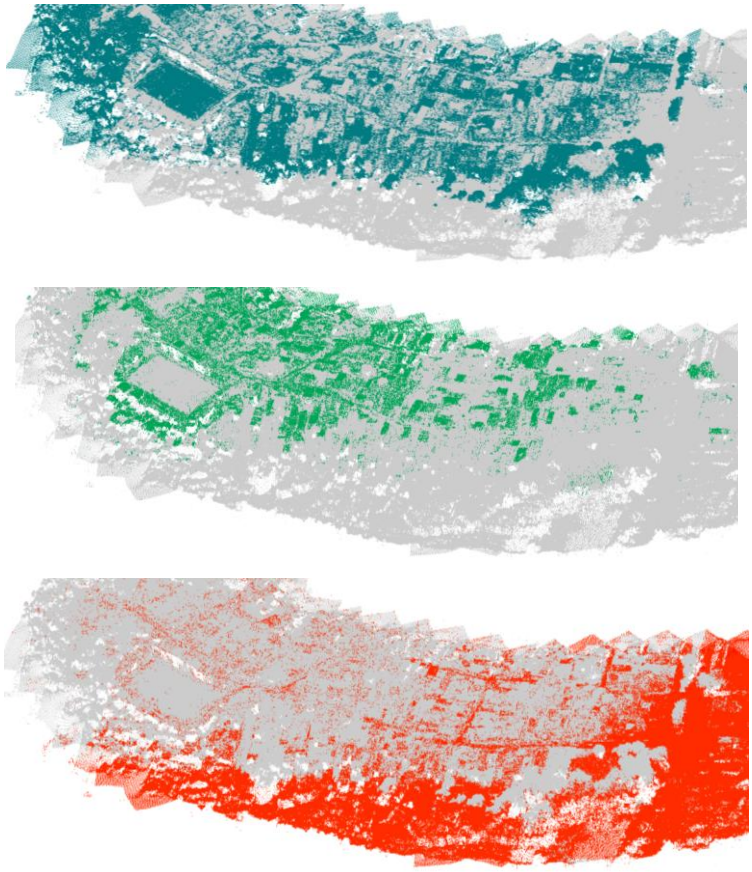

Figura 28: Nuvem obtida por voo de drone classificada.

Fonte: A autora.

Na Tabela 4 está indicado o relatório de aprendizado da classificação da nuvem de pontos obtida por fotogrametria, onde novamente o *ground* teve o pior desempenho.

Tabela 4: Relatório de aprendizado aplicado na nuvem obtida por fotogrametria.

	Precision	Recall	F1-score
Buildings	0,97	0,99	0,98
Ground	0,57	0,27	0,37
Others	0,85	0,83	0,84
Vegetation	0,96	0,98	0,97

Fonte: A autora.

5.4. TRATAMENTO NOS RESULTADOS

Para simular o comportamento desejado que o algoritmo teria, foi realizado um pós-tratamento em uma das nuvens classificadas apresentada durante os resultados para analisar os potenciais oportunidades que o algoritmo pode trazer para a modelagem BIM.

Foi utilizada no *CloudCompare* uma das nuvens obtidas a partir da classificação da vegetação apresentada nos resultados. No *CloudCompare*, há uma função chamada SOR

(*Statistical Outlier Remover*), que permite ajustar os parâmetros para realizar uma “limpeza” na nuvem de pontos. Os parâmetros escolhidos podem ser observados na Figura 29.

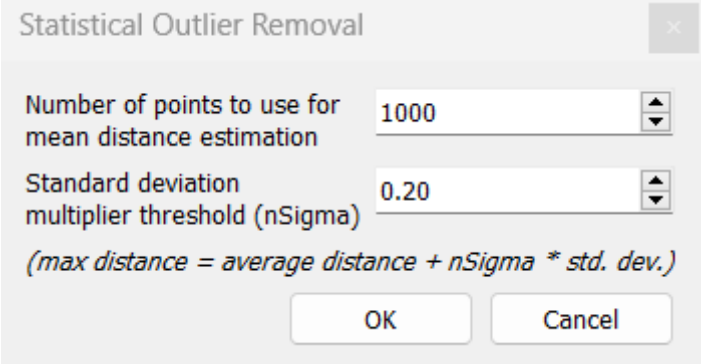

Figura 29: Função Statistical Outlier Remover no *CloudCompare*.

Fonte: A autora.

Como resultado da função e dos parâmetros escolhidos, o *CloudCompare* removeu os outliers e manteve apenas os pontos mais “aglomerados”, que realmente indicavam a presença de vegetação, ignorando os ruídos presentes na classificação pelo aprendizado. A nuvem obtida pode ser vista na Figura 30.

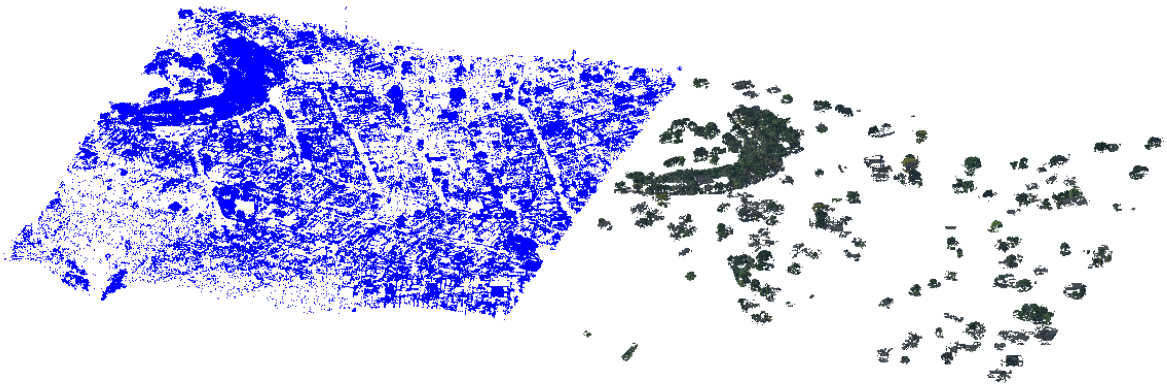

Figura 30: Nuvem de pontos da vegetação antes e após tratamento no *CloudCompare*.

Fonte: A autora.

Esse resultado traz a possibilidade de utilizar as nuvens separadas por classes dentro de *softwares* BIM, caso essa nuvem final fosse transformada em uma família dentro do Revit, por exemplo. Em uma etapa inicial, o uso dessas nuvens seria apenas para facilitar a modelagem, tendo em vista que o escopo deste estudo se preocupou apenas em classificar as nuvens de pontos. Porém, posteriormente, o algoritmo desenvolvido poderia ser utilizado junto a algoritmos de segmentação para que cada um dos elementos fosse separado de forma individual.

Isso é particularmente útil em um cenário onde se deseja filtrar apenas determinados elementos de um levantamento realizado por drone, por exemplo.

As nuvens divididas por classes podem ser posteriormente trabalhadas para serem transformadas em superfícies e então em objetos BIM. Isso permitiria que não houvesse necessidade de modelar objetos que não fossem o foco do projeto e que não requisitassem informação semântica, como, por exemplo, a vegetação. A classificação das nuvens também pode ser útil para desenvolvimento de estudos focados no processo de extração de informação semântica de nuvens de pontos, que se apresentou como um obstáculo existente na literatura estudada.

6 CONCLUSÕES E PERSPECTIVAS FUTURAS

6.1. CONCLUSÕES

Neste trabalho foi apresentado um script desenvolvido a partir do algoritmo de classificação Random Forest, capaz de classificar nuvens de pontos oriundas de *LiDAR* e de fotogrametria em três classes: construção, vegetação e terreno. Foi visto que o aprendizado para a classificação do terreno não foi tão satisfatório como as demais camadas, mas isso pode ser justificado pela qualidade dos dados de entrada utilizados, conforme discutido anteriormente.

Como contribuição principal, o *script* desenvolvido pode ser utilizado para aprendizado de qualquer *dataset* de nuvens de pontos, desde que o formato da nuvem tenha as informações das coordenadas, cores e classificação de cada ponto presente na nuvem. Da mesma forma, o *script* é capaz de classificar qualquer nuvem de pontos, também desde que ela tenha informações de coordenadas e cores. Isso é particularmente útil para separação de classes específicas em nuvens de pontos de grandes áreas, como nuvens obtidas por meio de levantamento aéreo via drone.

Um ponto interessante a ser observado é que, em um dos resultados, apesar de algumas construções terem sido classificadas corretamente, apenas foram identificadas aquelas que possuíam telhados alaranjados. Isso pode indicar que o *dataset* utilizado não foi suficiente para que o algoritmo também aprendesse outras variedades de telhados. Outra interpretação desse mesmo resultado recai sobre os diferentes padrões de construção existentes no Brasil, pois há a presença de diferentes materiais com diferentes cores, como telhas de barro, de fibrocimento, de metal e outras, dificultando o reconhecimento de diferentes tipos de telha.

Nos casos em que os telhados são acinzentados, a aplicação do algoritmo pode confundi-los com a pavimentação, que é majoritariamente cinza. Outro fator que pode confundir o aprendizado é a presença de sombras no momento do escaneamento das localidades, pois a presença de construções verticais gera nuvens sobre algumas áreas da nuvem. No entanto, no *dataset* utilizado acredita-se que não houve problemas de interpretação causadas por sombras e sim pela ruidez nos dados de entrada.

Neste trabalho foram utilizadas apenas as informações de cor e coordenadas nas nuvens visando tornar o algoritmo mais “universal”, tendo em vista que a morfologia de nuvens de pontos de *LiDAR* e de fotogrametria são diferentes. Porém, acredita-se que a utilização dos demais parâmetros provenientes do escaneamento a *laser* também possam melhorar o

aprendizado, tendo em vista que eles tornariam mais fácil reconhecer certos padrões existentes nas camadas.

6.2. PERSPECTIVAS FUTURAS

O trabalho também avaliou a influência do tamanho do *dataset* utilizado no aprendizado do algoritmo, com testes feitos com diferentes quantidades de pontos. Foi observado que o aprendizado com uma menor quantidade de pontos tem boas taxas, porém a classificação de determinadas camadas fica melhor definida com uma maior quantidade de pontos no *dataset*, como foi o caso da vegetação. Dessa forma, acredita-se que quanto mais representativo o *dataset*, com nuvens densas, melhor o aprendizado e, consequentemente, melhor a classificação das nuvens.

Como perspectivas futuras a partir deste estudo, pretende-se aprimorar o algoritmo e tornar os dados de entrada mais refinados. Com a classificação mais definida, poderiam ser utilizados algoritmos de clusterização de nuvens de pontos para transformar cada um dos objetos presentes na nuvem em superfícies, os chamados *mesh*. A partir disso, poderiam ser transformados em famílias Revit, por exemplo. Não havendo a necessidade de modelar esses objetos que não precisassem de parametrização. Isso tornaria o projeto mais fiel ao “real”, não havendo a necessidade de buscar por famílias de forma online, e economizaria tempo de modelagem.

Também será estudado o reconhecimento de objetos individuais de forma automática em nuvens de pontos com a junção de algoritmos de segmentação e classificação, pois neste estudo apenas foi utilizado um algoritmo de classificação. Porém, acrescentar a segmentação ao processo facilita o reconhecimento de objetos das nuvens, além da possibilidade de transformar esses objetos em elementos BIM parametrizados, que ainda se apresenta como um desafio em toda a literatura analisada. A intenção é tornar cada vez mais prática a modelagem BIM com auxílio de nuvens de pontos.

Para trabalhos futuros, também se propõe que o algoritmo seja adaptado para apenas nuvens de pontos provenientes de *LiDAR* e com uso dos demais parâmetros, para avaliar o impacto que isso teria nos resultados obtidos. Também serão testadas outras técnicas de aprendizado de máquina, como SVM, *Deep Learning* e outras que apareceram na etapa de análise bibliométrica deste estudo. A intenção é analisar as diferenças nos resultados obtidos a partir do Random Forest, que foi utilizado nessa pesquisa, e das demais técnicas, e de que forma essas técnicas se comportam no algoritmo desenvolvido.

REFERÊNCIAS

Abreu, N., Pinto, A., Matos, A., & Pires, M. (2023). Procedural Point Cloud Modelling in Scan-to-BIM and Scan-vs-BIM Applications: A Review. In *ISPRS International Journal of Geo-Information* (Vol. 12, Issue 7). Multidisciplinary Digital Publishing Institute (MDPI). https://doi.org/10.3390/ijgi12070260

Agapaki, E., Miatt, G., & Brilakis, I. (2018). Prioritizing object types for modelling existing industrial facilities. *Automation in Construction*, *96*, 211–223. https://doi.org/10.1016/j.autcon.2018.09.011

Albano, R. (2019). Investigation on Roof Segmentation for 3D Building Reconstruction from Aerial LIDAR Point Clouds. *Applied Sciences 2019, Vol. 9, Page 4674*, *9*(21), 4674. https://doi.org/10.3390/APP9214674

Amorim, A. L. de. (2015). DISCUTINDO CITY INFORMATION MODELING (CIM) E CONCEITOS CORRELATOS. *Gestão & Tecnologia de Projetos*, *10*(2), 87. https://doi.org/10.11606/gtp.v10i2.103163

Anil, E. B., Tang, P., Akinci, B., & Huber, D. (2011). *Assessment of Quality of As-is Building Information Models Generated from Point Clouds Using Deviation Analysis*.

Ayodele, T. O. (2010). Types of Machine Learning Algorithms. *New Advances in Machine Learning*. https://doi.org/10.5772/9385

Azhar, S. (2011). Building information modeling (BIM): Trends, benefits, risks, and challenges for the AEC industry. *Leadership and Management in Engineering*, *11*(3), 241–252. https://doi.org/10.1061/(ASCE)LM.1943-5630.0000127

Bahreini, F., & Hammad, A. (2024). Dynamic graph CNN based semantic segmentation of concrete defects and as-inspected modeling. *Automation in Construction*, *159*. https://doi.org/10.1016/j.autcon.2024.105282

Bassier, M., & Vergauwen, M. (2020). Unsupervised reconstruction of Building Information Modeling wall objects from point cloud data. *Automation in Construction*, *120*, 103338. https://doi.org/10.1016/J.AUTCON.2020.103338

Bello, S. A., Yu, S., Wang, C., Adam, J. M., & Li, J. (2020). Review: Deep Learning on 3D Point Clouds. *Remote Sensing 2020, Vol. 12, Page 1729*, *12*(11), 1729. https://doi.org/10.3390/RS12111729

Bensalah, M., Elouadi, A., & Mharzi, H. (2019). Overview: the opportunity of BIM in railway. *Smart and Sustainable Built Environment*, *8*(2), 103–116. https://doi.org/10.1108/SASBE-11-2017-0060

Bosché, F., Ahmed, M., Turkan, Y., Haas, C. T., & Haas, R. (2015). The value of integrating Scan-to-BIM and Scan-vs-BIM techniques for construction monitoring using laser scanning and BIM: The case of cylindrical MEP components. *Automation in Construction*, *49*, 201–213. https://doi.org/10.1016/J.AUTCON.2014.05.014

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

Caelen, O. (2017). A Bayesian interpretation of the confusion matrix. *Annals of Mathematics and Artificial Intelligence*, *81*(3–4), 429–450. https://doi.org/10.1007/S10472-017-9564-8/METRICS

Collins, F. C., Ringsquandl, M., Braun, A., Hall, D. M., & Borrmann, A. (2022). Shape encoding for semantic healing of design models and knowledge transfer to scan-to-BIM. *Proceedings of the Institution of Civil Engineers: Smart Infrastructure and Construction*, *175*(4), 160–180. https://doi.org/10.1680/jsmic.21.00032

Cooney, J. P., Oloke, D., & Gyoh, L. (2023). A novel heritage BIM (HBIM) framework development for heritage buildings refurbishment based on an investigative study of microorganisms. *Journal of Engineering, Design and Technology*, *21*(4), 1046–1082. https://doi.org/10.1108/JEDT-07-2021-0370

Croce, V., Caroti, G., Piemonte, A., De Luca, L., & Véron, P. (2023). H-BIM and Artificial Intelligence: Classification of Architectural Heritage for Semi-Automatic Scan-to-BIM Reconstruction. *Sensors*, *23*(5). https://doi.org/10.3390/s23052497

Decuyper, M., Stockhoff, M., Vandenberghe, S., -, al, & Ying, X. (2019). An Overview of Overfitting and its Solutions. *Journal of Physics: Conference Series*, *1168*(2), 022022. https://doi.org/10.1088/1742-6596/1168/2/022022

Diab, A., Kashef, R., & Shaker, A. (2022). Deep Learning for LiDAR Point Cloud Classification in Remote Sensing. *Sensors 2022, Vol. 22, Page 7868*, *22*(20), 7868. https://doi.org/10.3390/S22207868

Dimitrov, A., & Golparvar-Fard, M. (2014). Vision-based material recognition for automated monitoring of construction progress and generating building information modeling from unordered site image collections. *Advanced Engineering Informatics*, *28*(1), 37–49. https://doi.org/10.1016/j.aei.2013.11.002

Ebrahimi, M., Hojat Jalali, H., & Sabatino, S. (2023). Probabilistic condition assessment of reinforced concrete sanitary sewer pipelines using LiDAR inspection data. *Automation in Construction*, *150*, 104857. https://doi.org/10.1016/J.AUTCON.2023.104857

Escudero, P. A. (2023). Scan-to-HBIM: automated transformation of point clouds into 3D BIM models for the digitization and preservation of historic buildings. *Vitruvio*, *8*(2), 52–63. https://doi.org/10.4995/vitruvio-ijats.2023.20413

Foody, G. M. (2023). Challenges in the real world use of classification accuracy metrics: From recall and precision to the Matthews correlation coefficient. *PLOS ONE*, *18*(10), e0291908. https://doi.org/10.1371/JOURNAL.PONE.0291908

Hammad, A. W. A., Da Costa, B. B. F., Soares, C. A. P., Haddad, A. N., Hammad, A. W. A. ;, Da Costa, B. B. F. ;, Soares, C. A. P. ;, & Haddad, A. N. (2021). The Use of Unmanned Aerial Vehicles for Dynamic Site Layout Planning in Large-Scale Construction Projects. *Buildings 2021, Vol. 11, Page 602*, *11*(12), 602. https://doi.org/10.3390/BUILDINGS11120602

Han, K. K., & Golparvar-Fard, M. (2015). Appearance-based material classification for monitoring of operation-level construction progress using 4D BIM and site photologs. *Automation in Construction*, *53*, 44–57. https://doi.org/10.1016/j.autcon.2015.02.007

Han, S., Jiang, Y., Huang, Y., Wang, M., Bai, Y., & Spool-White, A. (2023). Scan2Drawing: Use of Deep Learning for As-Built Model Landscape Architecture.

Journal of Construction Engineering and Management, *149*(5). https://doi.org/10.1061/JCEMD4.COENG-13077

Heron, M. J., Hanson, V. L., & Ricketts, I. (2013). Open source and accessibility: advantages and limitations. *Journal of Interaction Science 2013 1:1*, *1*(1), 1–10. https://doi.org/10.1186/2194-0827-1-2

Heydarian, M., Doyle, T. E., & Samavi, R. (2022). MLCM: Multi-Label Confusion Matrix. *IEEE Access*, *10*, 19083–19095. https://doi.org/10.1109/ACCESS.2022.3151048

Hiasa, S., & Kawamoto, H. (2023). TECHNICAL REPORT ON BIM/CIM APPLICATION TO A HIGHWAY PROJECT: ADVANTAGES AND CHALLENGES OF LASER TOPOGRAPHIC SURVEY. *Journal of Japan Society of Civil Engineers*, *11*(1), E0011. https://doi.org/10.2208/JOURNALOFJSCE.F3-E0011

Hossin, M., & Sulaiman, M. (2015). A Review on Evaluation Metrics for Data Classification Evaluations. *International Journal of Data Mining & Knowledge Management Process*, *5*(2), 01–11. https://doi.org/10.5121/IJDKP.2015.5201

Igor, W., Oliveira, A., Leão Da Silva, F. W., & Melo Barros, L. (2023). Reconstrução de um projeto comercial com a utilização da tecnologia scanner para BIM. *The Journal of Engineering and Exact Sciences*, *9*(3), 15605–01e. https://doi.org/10.18540/jcecvl9iss3pp15605-01e

Khan, M. Y., Qayoom, A., Nizami, M. S., Siddiqui, M. S., Wasi, S., & Raazi, S. M. K. U. R. (2021). Automated Prediction of Good Dictionary EXamples (GDEX): A Comprehensive Experiment with Distant Supervision, Machine Learning, and Word Embedding-Based Deep Learning Techniques. *Complexity*, *2021*. https://doi.org/10.1155/2021/2553199

Khomsin, Pratomo, D. G., Anjasmara, I. M., & Ahmad, F. (2019). ANALYSIS of the VOLUME COMPARATION of 3'S (TS, GNSS and TLS). *E3S Web of Conferences*, *94*. https://doi.org/10.1051/E3SCONF/20199401014

Kim, M., & Lee, D. (2023). Automated two-dimensional geometric model reconstruction from point cloud data for construction quality inspection and maintenance. *Automation in Construction*, *154*. https://doi.org/10.1016/J.AUTCON.2023.105024

Kingsford, C., & Salzberg, S. L. (2008). What are decision trees? *Nature Biotechnology 2008 26:9*, *26*(9), 1011–1013. https://doi.org/10.1038/nbt0908-1011

Lee, J. H., Park, J. J., & Yoon, H. (2020). Automatic Bridge Design Parameter Extraction for Scan-to-BIM. *Applied Sciences 2020, Vol. 10, Page 7346*, *10*(20), 7346. https://doi.org/10.3390/APP10207346

Lee, J., Jo, H., & Oh, J. (2023). Application of Drone LiDAR Survey for Evaluation of a Long-Term Consolidation Settlement of Large Land Reclamation. *Applied Sciences (Switzerland)*, *13*(14), 8277. https://doi.org/10.3390/app13148277

Lee, J., Lee, Y., Park, S., & Hong, C. (2023). Implementing a Digital Twin of an Underground Utility Tunnel for Geospatial Feature Extraction Using a Multimodal Image Sensor. *Applied Sciences (Switzerland)*, *13*(16). https://doi.org/10.3390/app13169137

Lopes, T. R. (2019). CIM e suas potencialidades parqa gestão da manutenção Curitiba. *Arq.Urb*, *25*, 123–140.

Luleci, F., Chi, J., Cruz-Neira, C., Reiners, D., & Catbas, F. N. (2024). Fusing infrastructure health monitoring data in point cloud. *Automation in Construction*, *165*, 105546. https://doi.org/10.1016/J.AUTCON.2024.105546

Ma, Y., Zheng, Y., Wang, S., Wong, Y. D., & Easa, S. M. (2022). Point cloud-based optimization of roadside LiDAR placement at constructed highways. *Automation in Construction*, *144*, 104629. https://doi.org/10.1016/J.AUTCON.2022.104629

Mahmoud, M., Chen, W., Yang, Y., & Li, Y. (2024). Automated BIM generation for large-scale indoor complex environments based on deep learning. *Automation in Construction*, *162*, 105376. https://doi.org/10.1016/j.autcon.2024.105376

Mihić, M., Sigmund, Z., Završki, I., & Butković, L. L. (2023). An Analysis of Potential Uses, Limitations and Barriers to Implementation of 3D Scan Data for Construction Management-Related Use—Are the Industry and the Technical Solutions Mature Enough for Adoption? *Buildings*, *13*(5). https://doi.org/10.3390/BUILDINGS13051184

Moyano, J., Nieto-Julián, J. E., Lenin, L. M., & Bruno, S. (2022). Operability of Point Cloud Data in an Architectural Heritage Information Model. *International Journal of Architectural Heritage*, *16*(10), 1588–1607. https://doi.org/10.1080/15583058.2021.1900951

Nieto-Julián, J. E., Farratell, J., Bouzas Cavada, M., & Moyano, J. (2023). Collaborative Workflow in an HBIM Project for the Restoration and Conservation of Cultural Heritage. *International Journal of Architectural Heritage*, *17*(11), 1813–1832. https://doi.org/10.1080/15583058.2022.2073294

O'Donnell, J., Truong-Hong, L., Boyle, N., Corry, E., Cao, J., & Laefer, D. F. (2019). LiDAR point-cloud mapping of building façades for building energy performance simulation. *Automation in Construction*, *107*, 102905. https://doi.org/10.1016/J.AUTCON.2019.102905

Ojeda, J. M. P., Huatangari, L. Q., Calderón, B. A. C., Tineo, J. L. P., Panca, C. Z. A., & Pino, M. E. M. (2024). Estimation of the Physical Progress of Work Using UAV and BIM in Construction Projects. *Civil Engineering Journal (Iran)*, *10*(2), 362–383. https://doi.org/10.28991/CEJ-2024-010-02-02

Parmar, A., Katariya, R., & Patel, V. (2019). A Review on Random Forest: An Ensemble Classifier. *Lecture Notes on Data Engineering and Communications Technologies*, *26*, 758–763. https://doi.org/10.1007/978-3-030-03146-6_86

Pérez, J. A., Gonçalves, G. R., & Galván, J. M. (2022). Comparative analysis of the land survey using UAS and classical topography in road layout projects[Análisis comparativo del levantamiento del terreno mediante UAS y topografía clásica en proyectos de trazado de carreteras]. *Informes de La Construccion*, *74*(565), e431. https://doi.org/10.3989/ic.86273

Puri, N., & Turkan, Y. (2020). Bridge construction progress monitoring using lidar and 4D design models. *Automation in Construction*, *109*, 102961. https://doi.org/10.1016/J.AUTCON.2019.102961

Rodrigues, F., Cotella, V., Rodrigues, H., Rocha, E., Freitas, F., & Matos, R. (2022). Application of Deep Learning Approach for the Classification of Buildings' Degradation State in a BIM Methodology. *Applied Sciences (Switzerland)*, *12*(15). https://doi.org/10.3390/app12157403

Rodriguez-Galiano, V., Sanchez-Castillo, M., Chica-Olmo, M., & Chica-Rivas, M. (2015). Machine learning predictive models for mineral prospectivity: An evaluation of neural networks, random forest, regression trees and support vector machines. *Ore Geology Reviews*, *71*, 804–818. https://doi.org/10.1016/J.OREGEOREV.2015.01.001

Romero-Jarén, R., & Arranz, J. J. (2021). Automatic segmentation and classification of BIM elements from point clouds. *Automation in Construction*, *124*. https://doi.org/10.1016/j.autcon.2021.103576

Sacks, R., Eastman, C., Lee, G., & Teicholz, P. (2018). *Bim Handbook: A Guide to Building Information Modeling for Owners, Designers, Engineers, Contractors, and Facility Managers* (3rd ed.).

Sacks, R., Ma, L., Yosef, R., Borrmann, A., Daum, S., & Kattel, U. (2017). Semantic Enrichment for Building Information Modeling: Procedure for Compiling Inference Rules and Operators for Complex Geometry. *Journal of Computing in Civil Engineering*, *31*(6). https://doi.org/10.1061/(ASCE)CP.1943-5487.0000705

Santos, R. L., Nobre, A., Carvalho, D. E. B. de, & Neto, M. P. (2022). Fotogrametria ao alcance de todos: a modelagem 3d com celulares. *Anais Do Seminário Do Programa de Pós-Graduação Em Desenho, Cultura e Interatividade* , *1*(1). https://doi.org/10.13102/DCI.V1I1.9636

Shirowzhan, S., Sepasgozar, S. M. E., Li, H., Trinder, J., & Tang, P. (2019). Comparative analysis of machine learning and point-based algorithms for detecting 3D changes in buildings over time using bi-temporal lidar data. *Automation in Construction*, *105*, 102841. https://doi.org/10.1016/J.AUTCON.2019.102841

Sothe, C., De Almeida, C. M., Schimalski, M. B., La Rosa, L. E. C., Castro, J. D. B., Feitosa, R. Q., Dalponte, M., Lima, C. L., Liesenberg, V., Miyoshi, G. T., & Tommaselli, A. M. G. (2020). Comparative performance of convolutional neural network, weighted and conventional support vector machine and random forest for classifying tree species using hyperspectral and photogrammetric data. *GIScience & Remote Sensing*, *57*(3), 369–394. https://doi.org/10.1080/15481603.2020.1712102

Succar, B. (2009). Building information modelling framework: A research and delivery foundation for industry stakeholders. *Automation in Construction*, *18*(3), 357–375. https://doi.org/10.1016/j.autcon.2008.10.003

Tang, P., Huber, D., Akinci, B., Lipman, R., & Lytle, A. (2010). Automatic reconstruction of as-built building information models from laser-scanned point clouds: A review of related techniques. In *Automation in Construction* (Vol. 19, Issue 7, pp. 829–843). Elsevier B.V. https://doi.org/10.1016/j.autcon.2010.06.007

Vetrivel, A., Gerke, M., Kerle, N., & Vosselman, G. (2015). Identification of damage in buildings based on gaps in 3D point clouds from very high resolution oblique airborne images. *ISPRS Journal of Photogrammetry and Remote Sensing*, *105*, 61–78. https://doi.org/10.1016/j.isprsjprs.2015.03.016

Volk, R., Stengel, J., & Schultmann, F. (2014). Building Information Modeling (BIM) for existing buildings — Literature review and future needs. *Automation in Construction*, *38*, 109–127. https://doi.org/10.1016/J.AUTCON.2013.10.023

Wang, B., Wang, Q., Cheng, J. C. P., & Yin, C. (2022). Object verification based on deep learning point feature comparison for scan-to-BIM. *Automation in Construction*, *142*, 104515. https://doi.org/10.1016/j.autcon.2022.104515

Wang, J., Sun, W., Shou, W., Wang, X., Wu, C., Chong, H. Y., Liu, Y., & Sun, C. (2015). Integrating BIM and LiDAR for Real-Time Construction Quality Control. *Journal of Intelligent and Robotic Systems: Theory and Applications*, *79*(3–4), 417–432. https://doi.org/10.1007/S10846-014-0116-8/METRICS

Wang, Q., Guo, J., & Kim, M. K. (2019). An Application Oriented Scan-to-BIM Framework. *Remote Sensing 2019, Vol. 11, Page 365*, *11*(3), 365. https://doi.org/10.3390/RS11030365

Wang, Q., & Kim, M. K. (2019). Applications of 3D point cloud data in the construction industry: A fifteen-year review from 2004 to 2018. *Advanced Engineering Informatics*, *39*, 306–319. https://doi.org/10.1016/J.AEI.2019.02.007

Wang, T., & Chen, H. M. (2023). Integration of building information modeling and project management in construction project life cycle. *Automation in Construction*, *150*, 104832. https://doi.org/10.1016/J.AUTCON.2023.104832

Xu, Y., Tuttas, S., Hoegner, L., & Stilla, U. (2018). Reconstruction of scaffolds from a photogrammetric point cloud of construction sites using a novel 3D local feature descriptor. *Automation in Construction*, *85*, 76–95. https://doi.org/10.1016/j.autcon.2017.09.014

Xu, Y., Zhou, Y., Sekula, P., & Ding, L. (2021). Machine learning in construction: From shallow to deep learning. *Developments in the Built Environment*, *6*, 100045. https://doi.org/10.1016/J.DIBE.2021.100045

Yin, H., Lin, Z., & Yeoh, J. K. W. (2023). Semantic localization on BIM-generated maps using a 3D LiDAR sensor. *Automation in Construction*, *146*, 104641. https://doi.org/10.1016/J.AUTCON.2022.104641

Zhang, Y., & Rabbat, M. (2018). A Graph-CNN for 3D Point Cloud Classification. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, *2018-April*, 6279–6283. https://doi.org/10.1109/ICASSP.2018.8462291