Automation in Construction 181 (2026) 106644

Contents lists available at ScienceDirect

Automation in Construction

journal homepage: www.elsevier.com/locate/autcon

Automating clash relevance filtering in BIM-based multidisciplinary coordination using machine learning

Rayane Ailem, Conrad Boton *

Research Laboratory on Information Technology in Construction, Department of Construction Engineering, École de Technologie Supérieure, 1100 Notre-Dame Street West, Montreal, Canada

ARTICLE INFO

Keywords:
Building information modeling
Machine learning
3D coordination
Clash detection
Artificial intelligence

ABSTRACT

In a context where Machine Learning (ML) is reshaping the construction industry and where normative frameworks such as ISO 19650 govern BIM data management, this paper aims to automate the filtering of true and false clashes in 3D models coordination process, using machine learning (ML). A metadata extraction plug-in is developed to gather the necessary data for training ML models. Tests are conducted on BIM models to evaluate the plug-in's ability to identify and classify clashes, followed by a reimplementation of the solution within an existing BIM software environment. Validation, carried out through both technical testing and feedback from industry professionals, demonstrates the plug-in's functionality and its ability to replicate the decision-making process of a BIM coordinator in clash filtering. Intended for construction professionals this paper highlights the potential of AI to enhance BIM quality control while complying with regulatory standards and meeting the practical needs of the industry.

1. Introduction

The construction industry is undergoing an unprecedented digital transformation driven by the growing adoption of Building Information Modeling (BIM) [1], which centralizes data management and facilitates coordination throughout the project lifecycle [2]. Clash detection is a critical component of 3D multidisciplinary coordination, as it enables project teams to identify and resolve spatial conflicts between building components early in the design phase. By ensuring that architectural, structural, and MEP systems are compatible within a single coordinated model, clash detection supports constructability reviews, enhances collaboration among disciplines, and contributes directly to overall project quality and efficiency [3,4].

While automated clash detection tools such as Navisworks can identify thousands of conflicts, the sheer volume—often unprioritized and including a high proportion of false positives (up to 60 %)—creates a significant burden for coordination teams [5–7]. Manual classification and prioritization of these clashes is time-consuming, prone to error, and often relies on subjective judgment, which may compromise the effectiveness of model reviews [4,7].

Recent research has explored hybrid approaches that combine geometric and semantic data extraction, business rule–based filtering, and user validation interfaces [7–10]. These approaches improve prioritization of critical clashes, reduce cognitive load, and standardize sorting criteria. However, most solutions remain limited to laboratory experiments or require extensive manual intervention, lacking direct integration into operational BIM platforms [9,11]. The integration of Artificial Intelligence (AI) and Machine Learning (ML) into clash management offers a promising path forward, allowing models to learn from previously classified conflicts and predict the relevance of new ones based on geometric and contextual attributes [12–15].

This study focuses on leveraging ML to automatically classify and filter clashes within Navisworks, targeting both interdisciplinary and intradisciplinary conflicts in public building projects. Unlike previous work, the proposed solution is fully integrated into a professional BIM environment, capable of extracting relevant data, processing it automatically, and presenting actionable results to end-users [10,11]. The objectives are to reduce the time required for clash review, improve detection accuracy, and ensure compliance with international standards such as ISO 19650 [16]. By aligning technological innovation with practical industry requirements, this work contributes to both the scientific literature on AI-assisted BIM and the professionalization of digital coordination practices, representing a significant step toward intelligent, reliable, and sustainable automation of quality control in

* Corresponding author.
E-mail addresses: rayane.ailem.1@ens.etsmtl.ca (R. Ailem), conrad.boton@etsmtl.ca (C. Boton).

https://doi.org/10.1016/j.autcon.2025.106644
Received 26 May 2025; Received in revised form 29 October 2025; Accepted 30 October 2025
Available online 4 November 2025

construction projects.

The remainder of this article is organized as follows: Section 2 reviews the literature, including the digital transformation in the construction industry, the control quality assessment of BIM models and the need for automation and intelligent clash sorting. Section 3 details the research approach while Section 4 presents the main results, in particular the issues and contextualization and the development of the artifact. Section 5 presents the evaluation of the artifact through tests conducted in Navisworks, evaluation by practitioners and criteria-based evaluation. Section 6 discusses the key findings including a comparative analysis with previous work and the formulation of recommendations. Section 7 concludes the paper.

2. Literature review

Multidisciplinary coordination in Building Information Modeling (BIM) projects remains a complex process, primarily due to the large number of irrelevant clashes generated during model integration. Recent research efforts have increasingly focused on automating the clash detection and classification processes to enhance efficiency and decision-making. This section reviews the main advances in this field and positions the present study within the broader context of BIM-based coordination research.

2.1. Need for automation and intelligent clash sorting

The clash detection process is a fundamental component of BIM-based design validation and multidisciplinary coordination [3]. It allows project teams to identify and address spatial conflicts between building components early in the design phase, thus reducing errors during construction [4]. Integrated into BIM workflows, clash detection enhances collaboration among disciplines, supports constructability reviews, and contributes to overall project quality and efficiency [3,4]. As such, it plays a key role in ensuring that the digital model reflects a coordinated and buildable design.

The increasing complexity of BIM projects has led to an overwhelming number of clashes detected automatically by tools such as Navisworks. This abundance of clashes, often unprioritized, creates a burden for coordination teams. Bitaraf et al. [7] highlight that this overload compromises the ability of stakeholders to focus on the truly critical clashes, thus undermining the effectiveness of model reviews. Minor or non-critical clashes take time to analyze, detracting attention from issues with significant technical or functional impacts. This situation forces BIM coordinators to manually filter clashes or rely on subjective prioritization methods, which introduces biases and inconsistencies in problem resolution. In large-scale projects, this approach becomes unsustainable without automated sorting tools. To address these limitations, several studies suggest integrating business rules into the clash filtering process. These rules help evaluate the relevance of a clash based on contextual criteria, such as the nature of the elements involved (structural or technical), the location of the conflict (e.g., circulation or technical zones), or the volume of objects in interaction. For example, a clash between a ventilation duct and a light partition has a different impact compared to a clash between a load-bearing beam and a staircase. Integrating such rules into a filtering engine enables better prioritization of clashes. Harode et al. [8] propose a weighting method based on attributes such as the object's criticality, its function within the project, and the density of nearby interferences. This approach helps make coordination processes more efficient by reducing the cognitive load on users and standardizing sorting criteria.

Given the limitations of fully manual methods and the complexity of total automation, the current trend is moving toward hybrid workflows. These approaches combine automatic extraction of geometric and semantic data, the application of business filtering rules, and a user interface dedicated to validation or adjustment. Bitaraf et al. [7] illustrate this trend with a Navisworks plugin that incorporates a multi-criteria weighting system to classify clashes based on their importance. Their work demonstrates that such an approach significantly reduces the number of irrelevant clashes while improving processing speed. Additionally, Hu and Castro-Lacouture [9] emphasize the need to design systems adaptable to different types of projects, disciplines, and use contexts. This requires configurable filtering systems based on adjustable parameters, such as technical, functional, or regulatory criteria.

The prospects offered by Artificial Intelligence (AI) for the construction industry are promising. One major development is the emergence of intelligent digital twins: virtual representations of infrastructure capable not only of replicating the physical state of structures but also of learning and adapting based on real-time data [12]. The concept of Construction 5.0, advocated by Bassir et al. [13], focuses on the integration of AI, robotics, and advanced automation to enhance not only efficiency but also the sustainability and customization of built assets. The progressive integration of artificial intelligence into the construction industry not only enables the automation of certain design and coordination tasks but also enhances decision-making through predictive analysis and intelligent content generation. In particular, generative AI models, such as large language models (LLMs), are identified as promising tools for supporting professionals in document creation, resource planning, and error detection, while adapting to the specific needs of individual projects [14]. In the longer term, experts foresee AI evolving into fully integrated decision-support systems capable of autonomously proposing optimized scenarios for the design, construction, operation, and decommissioning of infrastructure [12]. While significant challenges remain, the progressive and controlled integration of AI promises a profound transformation of the AEC sector, making projects safer, faster, more cost-effective, and more sustainable.

2.2. Positioning and relevance of the research work

Navisworks is one of the most widely used tools for BIM coordination and interference detection. It offers extensive compatibility with standard formats such as IFC, as well as a powerful geometric clash detection engine. However, its capabilities in semantic analysis and conflict prioritization remain limited. The tool relies solely on geometric criteria and does not offer intelligent ranking based on criticality or business impact. Mehrbod et al. [4] highlight that the lack of direct integration of business rules or logic in the clash detection engine hinders its effectiveness in decision-making. Additionally, the manual export of reports and external management, often via Excel or BCF [17], complicate the utilization of the results.

Integrating a plug-in into the Navisworks environment paves the way for a more dynamic use of data from interference detection. Kazado et al. [10] demonstrated that it is possible to enrich Navisworks with an add-in developed with the .NET API, capable of extracting real-time data, structuring it, and visualizing it in 3D, while remaining non-intrusive to the original model. The principle is based on creating a modular interface composed of several functions: displaying general information, searching for objects by category/type, colorimetric visualization of parameters (such as temperature or CO_2 concentration in the initial example), and most importantly, reading external databases containing classified information (e.g., conflict criticality level). The link between the data and the model objects is ensured by unique identifiers, such as part or element numbers. This architecture allows for the integration of project-specific business rules (such as clearance zones or critical element types) directly into the visualization process. Through dynamic coloring and adaptive transparency of 3D objects, the user can quickly identify priority areas without modifying the model. The system relies on Excel databases that can be queried via OLE DB, ensuring flexibility in adapting to different data sets. This solution offers several advantages: reduced human error risks, the ability to integrate multiple software (Revit, Archicad, Allplan, etc.), non-destructive 3D visualization, and real-time updates via direct data reading. By integrating this logic into a plugin, Kazado et al. [10] show that it is possible to overcome the limitations of Navisworks' native functions and create a business-oriented interface tailored to the real needs of BIM coordinators, which can also include intelligent solutions such as Machine Learning.

Machine Learning (ML), and particularly supervised learning, represents a promising solution for automating the sorting of conflicts based on their relevance. Unlike static rules, ML models can learn from previously classified conflicts (labeled data) and predict the criticality level of new conflicts based on geometric and contextual features. In this project, several supervised models were tested, including Random Forest and Multi-Layer Perceptron (MLP). These models were trained using a dataset of conflicts extracted from several projects. Input variables included attributes such as the type of conflicting elements, their volume, their originating discipline, or their position within the model. The results showed that these models could categorize conflicts into relevant categories (critical, moderate, negligible) with promising accuracy rates. Gupta et al. [15] emphasize that the choice of model strongly depends on the quality of the data and the relevance of the labels used. Indeed, supervised learning relies on labeled data, which implies a prior data preparation phase for training the dataset. These data can be gradually obtained through the accumulation of project data over time. Ultimately, this approach can be integrated into a broader intelligent coordination process, where ML could assist BIM coordinators by helping them focus on the truly priority conflicts.

The project stands out in the literature for its aim to bridge algorithmic research and industrial application in an operational tool that can be directly integrated into business processes. Several previous works have demonstrated the potential of machine learning for classifying BIM conflicts, notably using textual metadata, as seen in the study by Hu and Castro-Lacouture [9], or from images, as in the work by Ahmadpanah [11]. However, these studies typically stop at laboratory experimentation or validation outside of operational contexts, without offering a solution that can be directly used on professional coordination platforms. This project fills that gap by offering an all-in-one solution integrated into Navisworks, capable of extracting relevant data, processing it automatically using supervised learning, and presenting the results in a clear and actionable way for end-users. The importance of this approach is emphasized by Mehrbod [4], who stresses the need for interoperable, modular tools compatible with existing BIM ecosystems to ensure real-world adoption. Moreover, the project is structured in alignment with the ISO 19650 standard, further enhancing its relevance in the context of the progressive structuring of digital practices around common frameworks. By automating conflict management while adhering to the principles of traceability, standardization, and document management promoted by this standard, the project contributes to aligning AI tools with institutional and industrial expectations regarding information governance. Thus, it contributes both to enriching the scientific literature on AI-assisted BIM and to the professionalization of digital coordination practices. Through its integrated approach, normative alignment, and operational focus, the project represents a significant step forward in the intelligent, reliable, and sustainable automation of quality control in construction projects.

3. Research approach

This project adopts a Design Science Research (DSR) approach [18], complemented by action research principles [19] in its experimental aspect, to design, develop, and evaluate a technological tool aimed at optimizing the quality control process within Building Information Modeling (BIM). The DSR methodology provides a structured framework to develop solutions to practical problems, focusing on iterative creation and validation of an artifact [20]. The intended artifact is a software plug-in integrated into a BIM environment, designed to automate conflict classification, thus reducing the time and resources required compared to manual processes, which are often prohibitive. The methodology is organized into five phases: Initial problem analysis and contextualization, research, solution development, and evaluation. Each phase builds on the previous one, in collaboration with industry stakeholders and in compliance with the ISO 19650 standard for BIM data management. The steps of the methodology are presented in Fig. 1.

3.1. Step 1: Initial problem analysis and contextualization

This phase, consistent with the DSR principle of problem relevance, defined the project's context by examining existing BIM quality control processes and establishing preliminary criteria. Current practices for conflict detection and classification were analyzed through workflow mapping and professional interviews that included targeted questions (Table 1). Three experts—a BIM service director, an MEP coordinator, and a generalist BIM coordinator—provided complementary perspectives on the challenges and limitations of prevailing methods.

An initial set of evaluation criteria was then established, aligned with the ISO 19650 standard, to guide the subsequent phases. These criteria include both qualitative metrics (e.g., ease of use) and quantitative metrics (e.g., potential time savings).

3.2. Development of a theoretical framework

The artifact developed in this study is a software plug-in integrated into Navisworks 2025, designed to automate the classification and prioritization of clashes, addressing a critical component of 3D multidisciplinary coordination. The primary goal was to develop an efficient, user-friendly tool compatible with industry workflows [21].

An iterative agile methodology was adopted, combining artifact design, model development, and user-centered prototyping [22,23]:

- Data Extraction: Labeled BIM data, including geometric and semantic metadata, were extracted via software interfaces. These data represented conflicts previously annotated by coordination teams.
- Model Selection and Training: Machine learning models (both supervised and unsupervised) were selected and trained on the extracted dataset. Iterative adjustments were made to optimize predictive performance and ensure relevance to real-world coordination tasks.
- Prototyping: A functional prototype with an intuitive user interface was developed to visualize and filter clashes. The prototype was tested in a controlled environment to validate usability and alignment with professional workflows.

The development environment included Navisworks 2025, Visual Studio Enterprise 2022, Visual Studio Code, and Google Colab Pro for parallel model training. The artifact was developed using C# and Python, and the workstation configuration is detailed in Table 2. This setup ensured compatibility with industrial BIM environments while allowing scalable and reproducible training of machine learning models.

3.3. Evaluation and validation

In accordance with the principles of Design Science Research (DSR), the evaluation phase aimed to assess both the effectiveness and practical utility of the developed artifact [18,24]. Evaluation is a critical component of DSR, as it allows researchers to verify whether the artifact achieves its intended goals and provides meaningful contributions to practice and theory [20].

The Navisworks plug-in was evaluated through tests on real projects,

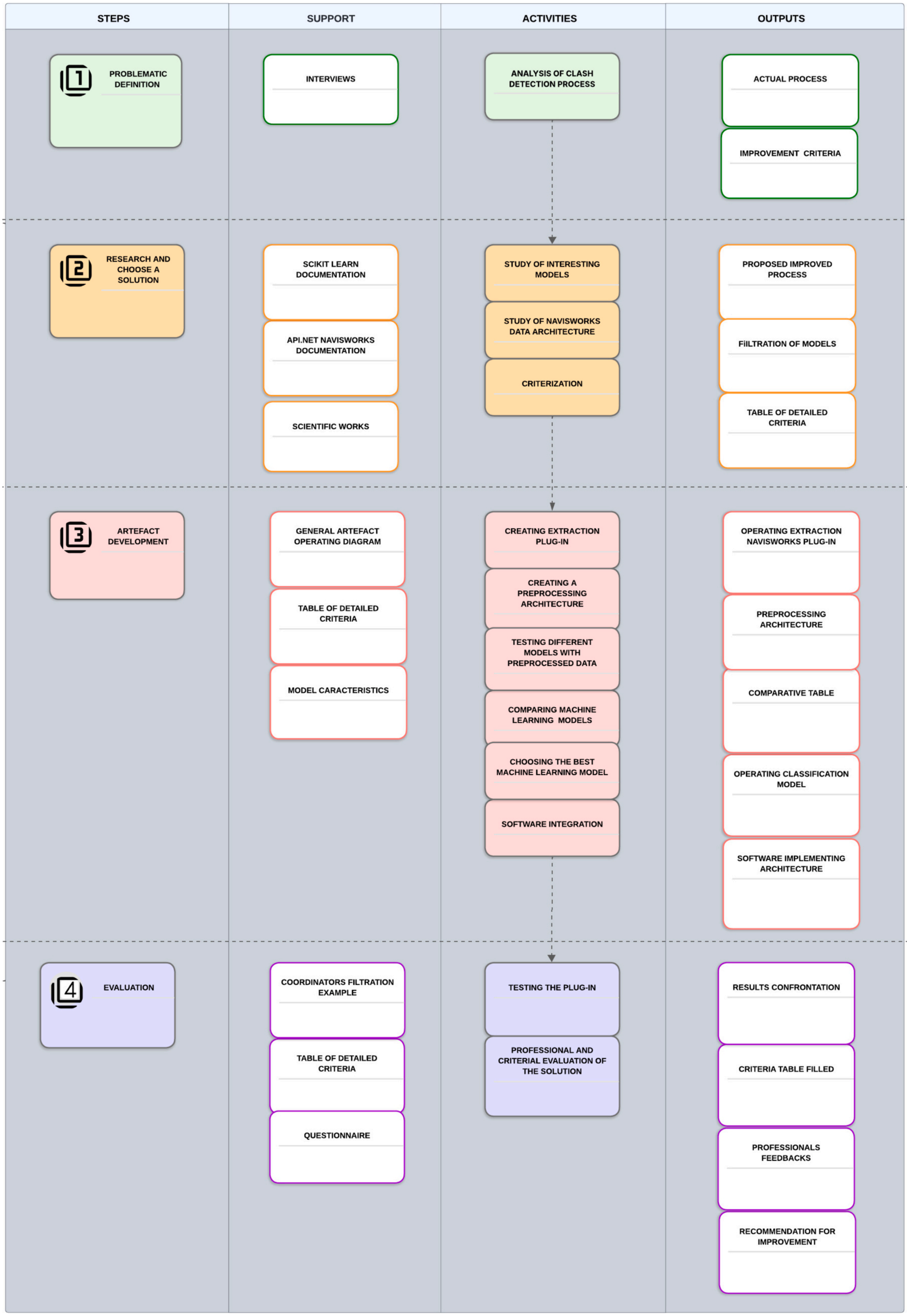

Fig. 1. Synthesis of the steps of the research approach.

Table 1
Questionnaire used during the preliminary meeting.

No.	Question	Purpose / Objective of the Question
1	Can you give me an overall description of the current process for detecting and managing conflicts in your BIM projects?	Understand the general workflow to guide the response.
2	How are Revit models transmitted between different stakeholders (BIM Manager, BIM coordinators)?	Identify the modes of communication and file exchange.
3	What tools do you use to detect clashes and monitor conflicts (e.g.: Navisworks, Newforma Konekt)?	Identify the main software tools used.
4	Once conflicts are detected in Navisworks, how are they analyzed, grouped, or sorted?	Assess the level of automation or manual effort.
5	How do you differentiate a real conflict from a false positive (e.g.: intentional intersection, unmodeled reservation)?	Understand the business logic behind conflict filtering.
6	At what point and in what form are the detection results communicated to the rest of the team?	Identify the dissemination channels and deliverable formats.
7	What happens to a conflict once it is validated as real? How do you monitor it until its resolution?	Understand the conflict lifecycle in the project.
8	What types of projects do you generally handle (schools, hospitals, etc.)?	Get an overview of the project context and volume.
9	Are there recurring technical constraints in conflict management? For example, elements that are non-modifiable like conduits.	Identify frequent technical obstacles.
10	How are priorities between disciplines defined?	Understand the methodological and contractual framework.

Table 2
Workstation device configuration.

Component	Details
Operating System	Windows 11 Pro – Version 24H2
Processor	Intel Core i7-10510U @ 1.80 GHz (Boost up to ~4.9 GHz)
RAM	16 GB
Graphics Card used	AMD Radeon RX 640 (2 GB)

comparing automated and manual coordination against criteria such as efficiency, usability, and workflow integration. A workshop and demonstration confirmed its ability to filter conflicts, prioritize issues with machine learning, and comply with BIM standards [21,23]. Feedback from professionals, collected via a structured questionnaire[1] (Table 3), and combined with quantitative and qualitative measures, provided a robust validation aligned with DSR best practices, ensuring both methodological rigor and practical relevance [20,22]. The artifact was tested iteratively through technical validation within Navisworks, practitioner assessment, and predefined criteria-based evaluation. This multi-method approach aligns with DSR best practices by combining relevance (addressing an actual coordination problem), rigor (using systematic data collection and measurable indicators), and design evaluation (empirical validation of the artifact in real project contexts). The process also emphasized communication and reflection, as professional feedback directly informed the refinement of the plug-in and the formulation of improvement strategies for future iterations.

[1] The questionnaire and interviews were conducted in accordance with the standard research ethics guidelines of École de Technologie Supérieure. Data collected from respondents were used exclusively for research purposes. The identities of the respondents remain confidential in compliance with all applicable ethical regulations of École de Technologie Supérieure.

Table 3
Evaluation questionnaire for professionals.

Questions	Targeted Responses
Role performed in BIM projects	Identification of the professional profile
Level of experience in the BIM environment	Knowledge and familiarity with the BIM ecosystem
Does the plug-in address a real issue in conflict management in Navisworks?	Assessment of the tool's relevance
Is the combination of automatic filtering AI useful?	Evaluation of the added value of the concept
Do you think this tool could be integrated into your current coordination processes?	Potential for integration into real workflow
Based on your experience with existing solutions, how do you rate the difficulty of using the filtering plug-in?	Evaluation of ergonomics and ease of use
Based on your experience, what are the strengths and limitations of this solution?	Collection of qualitative feedback (strengths/weaknesses)
What is your overall assessment of the presented solution?	Synthetic evaluation of overall satisfaction
Select the future development paths	Collection of key priorities for functional improvement
Do you have suggestions for improvements or additional features?	Open suggestions for future development
Would you like to be contacted for a discussion or a test version?	Interest in future engagement/ use

4. Main results

In this section, we present the main results, including the issues and contextualization, the development of the artifact, and its testing.

4.1. Issues and contextualization

The following subsections contextualize the research by examining the current BIM-based coordination process and the challenges it presents in practice. They outline the key issues that motivated the development of the proposed approach and discuss opportunities for improving coordination efficiency and reliability. The section also introduces the quality criteria that guided the design and evaluation of the proposed plug-in.

4.1.1. Process under study

This workflow, involving discipline-specific BIM Managers and BIM Coordinators, begins with an interference detection request initiated by the BIM Manager, who transmits the Revit models to the BIM Coordinator via a common data environment (CDE) or email. A CDE refers to a centralized digital space that enables all stakeholders to collect, manage, validate, and share project data throughout its entire life cycle [25,26]. The coordinator receives these models and exports them in NWC format for federation in Navisworks, where a global model is created. Clash detection tests are then carried out using Navisworks' Clash Detective tool, followed by a manual grouping of clashes by density zones to facilitate their analysis. A crucial step involves separating relevant conflicts (unintentional) from false or irrelevant ones (intentional intersections, such as unexecuted reservations), before exporting the real conflicts to Newforma Konekt for collaborative tracking. An interference report is generated in PDF format and shared via Microsoft Teams, Autodesk Construction Cloud (ACC), or email, before the corrected models are returned to the BIM Manager via the CDE or email, completing the cycle. The analyzed projects concerned public buildings (schools, hospitals), generating an average of 10,000 conflicts during the initial detection. The primary tools identified include Navisworks' Clash Detective for interference detection and Newforma Konekt for conflict tracking. Interviews also highlighted technical constraints, such as the need to maintain the alignment of primary fire alarm conduits and the prioritization of disciplines according to the General BIM Plan and the BIM Execution Plan.

Thus, within the scope of this study, the classic quality control process without the use of machine learning (ML) was carefully analyzed and mapped using a BPMN diagram (Fig. 2).

4.1.2. Issue with the current process and avenues for improvement

The current process requires a significant amount of time and focus to separate true and irrelevant conflicts, as well as to group conflicts by zone for easier management. It was also pointed out that an analysis based solely on images cannot be relevant, as many conflicts require a better contextualized view. This highlights the complexity of managing conflicts without a global perspective, which is typically provided by the BIM coordinators.

This need for a more efficient and contextualized approach led to the creation of the following set of criteria that the solution must meet (Table 4).

A revised clash detection workflow (Fig. 3) is proposed to address limitations of the previous process by integrating automation and centralization. Clashes are now grouped automatically using Autodesk Navisworks' Issue Add-In, while a dedicated plug-in separates relevant and irrelevant clashes by updating their status directly in Navisworks. Issue management can be handled via Newforma Konekt or Autodesk Construction Cloud, with the plug-in also supporting detailed reporting. Although automated clash filtering is planned for a future phase, the updated workflow already reduces manual effort and enhances project efficiency.

4.1.3. Quality criteria for the proposed plug-in

To evaluate the overall performance of the developed plug-in, a set of criteria was defined across several key dimensions (Table 5): data extraction, ease of use, business relevance, prediction accuracy, software integration, scalability, and compliance. These criteria were not limited to a purely technical perspective but also incorporated the normative requirements of the ISO 19650 series, particularly regarding information management, traceability, security, and workflow efficiency. The following table presents these criteria, their rationale, associated indicators, and their potential alignment with ISO principles.

4.2. Development of the artifact

This section will describe each part of the artifact in detail, outlining its development and functionality according to a four-step development process (Fig. 4).

The four steps required to create a functional plug-in can be summarized as follows: data extraction from Navisworks, preprocessing of this data, training of models, integrating the best option selected into a prediction architecture, and finally, integrating the entire solution into Navisworks.

4.2.1. Data extraction

This section describes the initial data extraction process, which served as the foundation for constructing the training conflict dataset used in our machine learning models.

4.2.1.1. Development. Conflict data was retrieved from two public building projects: a long-term care facility and a university, for a total of 36,562 individual conflicts. Initially, the extraction of conflict data from Navisworks was intended to be done by utilizing the conflict report feature of Clash Detective in its HTML Tabular format (Fig. 5), as several studies have done before. However, access to a limited number and types of data without initial customization actions, as well as the need for additional preprocessing to convert data from HTML formats into AI-interpretable formats, led to the creation of a new extraction module that will integrate into the workflow of our final plugin.

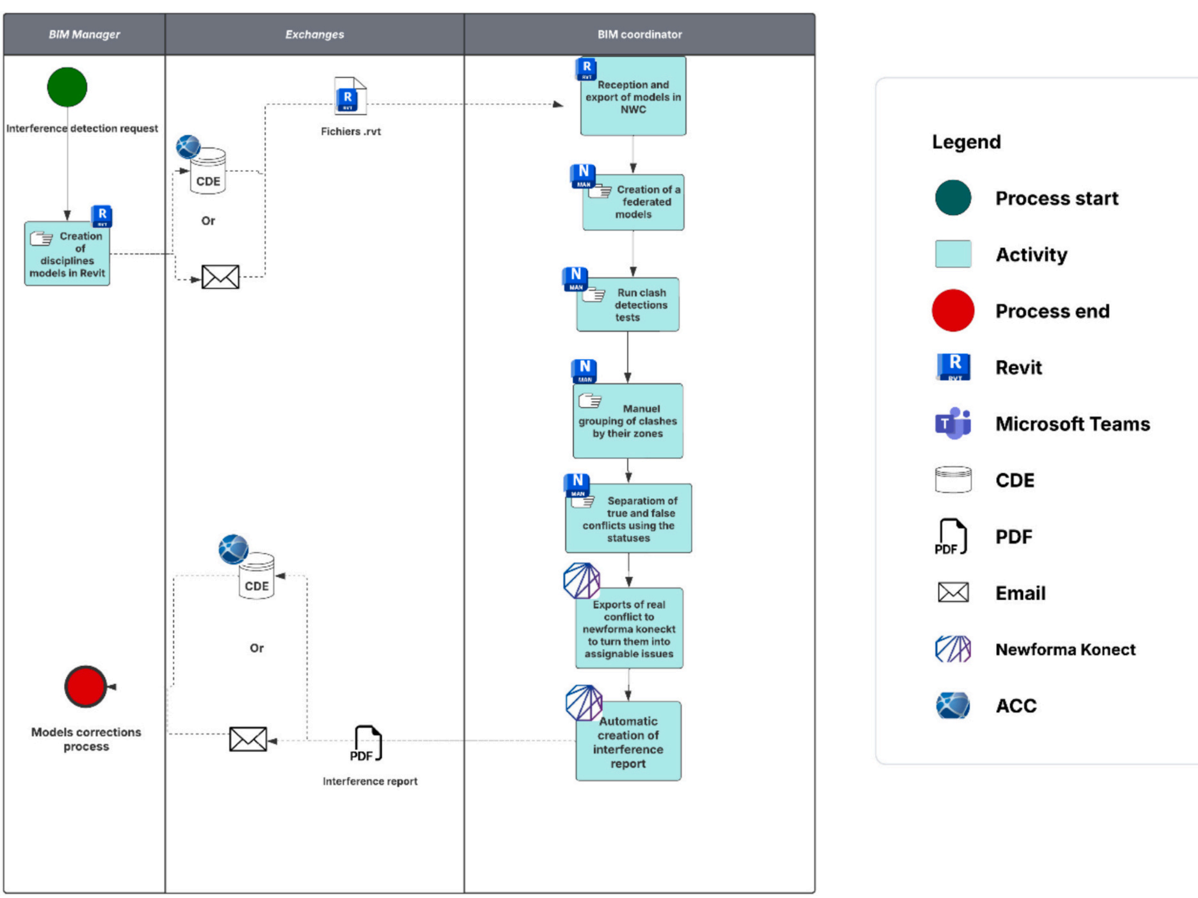

Fig. 2. Current interference detection process.

Table 4
Improvement criteria.

Category	Improvement criterion	Description	Representative metric
Functional efficiency	Management of relevant/irrelevant conflicts	Identify relevant conflicts (unintentional) and irrelevant ones (intentional intersections)	Detection rate equivalent to a BIM coordinator
Technical performance	Integration with existing tools	Seamlessly integrate with Navisworks	Integration with Navisworks
	Reproductibility	Applicability to various public building projects	Capacity to be applied across various projects
Safety and ethics	Data protection	Protect sensitive data in compliance with regulations (e.g., GDPR)	Compliance with data security standards
	Data traceability	Preserve metadata for clear and auditable tracking	Percentage of metadata retained

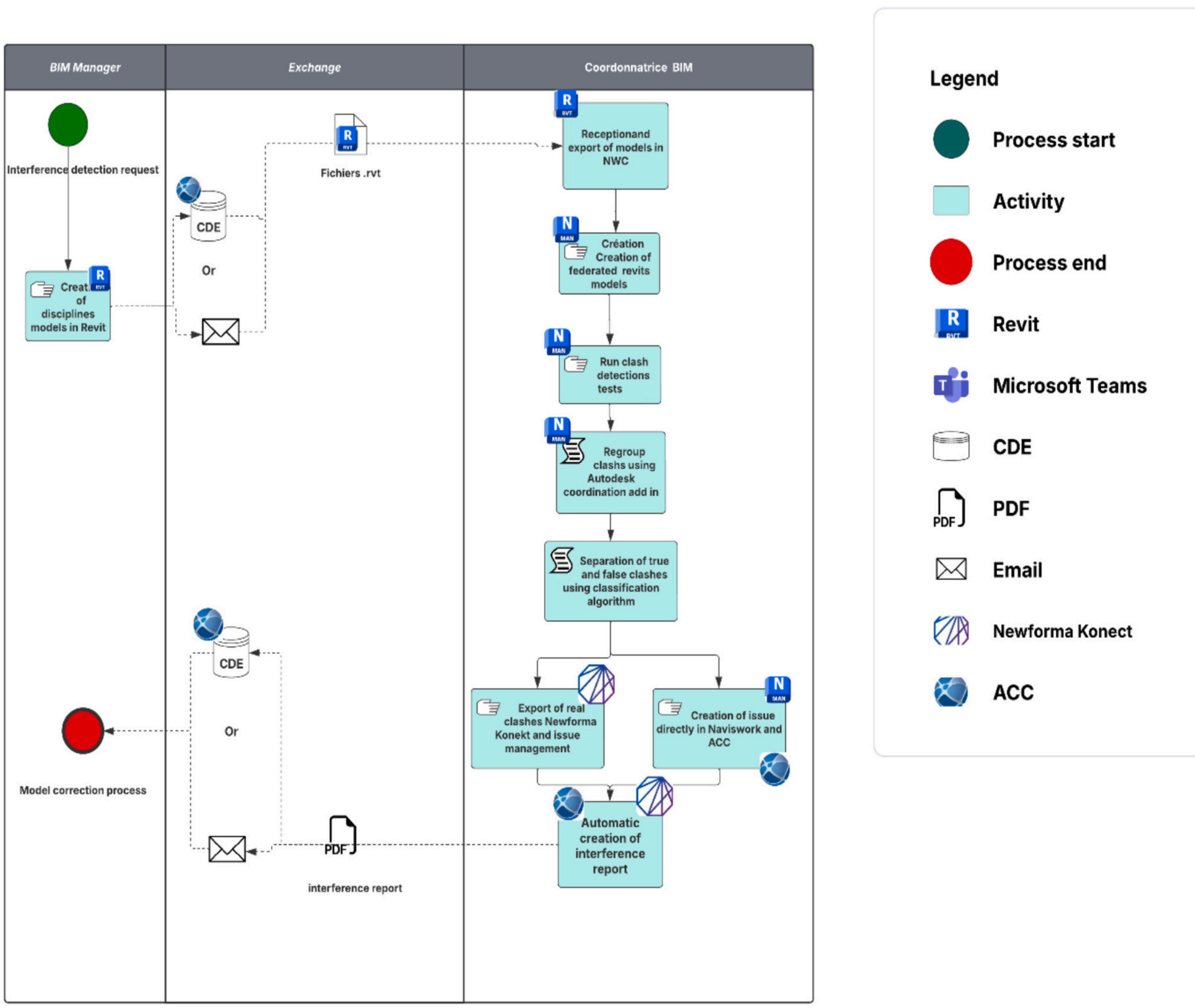

Fig. 3. Proposed enhanced clash detection process.

The extraction module of the plug-in was developed using the Navisworks .NET API, with the NavisLookUp Add-in employed to identify relevant property class paths (e.g., *ModelItem, ClashResult, Geometry*). These classes provided essential attributes such as bounding box coordinates, disciplines, clash types, and geometric values (e.g., distance, overlap volume). Dedicated methods (e.g., *ExtractClashData*) retrieved this information while managing missing properties, such as replacing absent bounding boxes with a default directional vector (1,0,0). To ensure consistency across heterogeneous datasets, all values were standardized via *ToDisplayString* (e.g., degrees, booleans as text) before being structured in JSON format using the Newtonsoft.Json library (Fig. 6).

The plug-in was developed in Visual Studio Enterprise 2022 as a Class Library project (.NET Framework 4.8) compatible with Navisworks 2025. Key references (e.g., *Autodesk.Navisworks.Api.dll, Autodesk.Navisworks.Automation.dll*) were integrated from the Navisworks installation, and *Newtonsoft.Json* was added via NuGet for JSON management. A main class implementing the *FilterPlugin* interface orchestrated extraction functions such as *ExtractClashData*. The compiled DLL was deployed to Navisworks' Plugins folder for testing, and debugging via Visual Studio ensured stable performance in real-world conditions.

4.2.1.2. Operation and utility. The process begins with the extraction of raw conflict information from the Navisworks model, referred to as *RepresentativeResults* in Fig. 6. This includes essential attributes such as distance between elements, conflict status, associated notes, conflict group, test identifier, description, and, when available, location.

Table 5
Detailed criteria.

Category	Criterion	Why It's Important	Indicator / Metric	Target Objective	ISO 19650 Link
Extraction	Extraction Quality	Ensure each clash contains complete data for analysis	% of clashes with complete properties	> 98 %	Data traceability
	Extraction Speed	Must integrate without slowing down business workflows	Extraction time for 10,000 clashes	< 10 s	–
Relevance	Alignment with Business Needs	Automatic sorting must reflect actual site priorities	Gap between automatic and expert sorting	< 10 %	Automated quality control
	Useful Filtering Rate	Effectively eliminate irrelevant false clashes	% of false clashes correctly filtered	> 85 %	Workflow efficiency
ML Accuracy	Classification Accuracy	Avoid misclassifying critical clashes	Accuracy, F1-score	F1 > 0.90	Performance indicator
	Recall	Ensure important clashes are not missed	Recall	Recall >0.85	Performance indicator
Integration	Compatibility with Navisworks	Remain within the BIM working environment	Direct integration (binary)	Yes/No	Common Data Environment (CDE)
	Interoperable Export (ACC, Newforma)	Enable integration into current workflows	Number of compatible output formats	≥ 2 formats	Standards compliance
Evolution	Scalability to New Projects	Ability to reuse the tool in various contexts	Number of projects tested	≥ 2 projects	Dynamic information updating
	Model Update Capability	Ensure continued adaptability to evolving data	Model reloading capability (binary)	Yes/No	Information updating
	Structured and Secure Export	Protect data and enable traceability	Structured and locked format	Yes/No	Data security / traceability

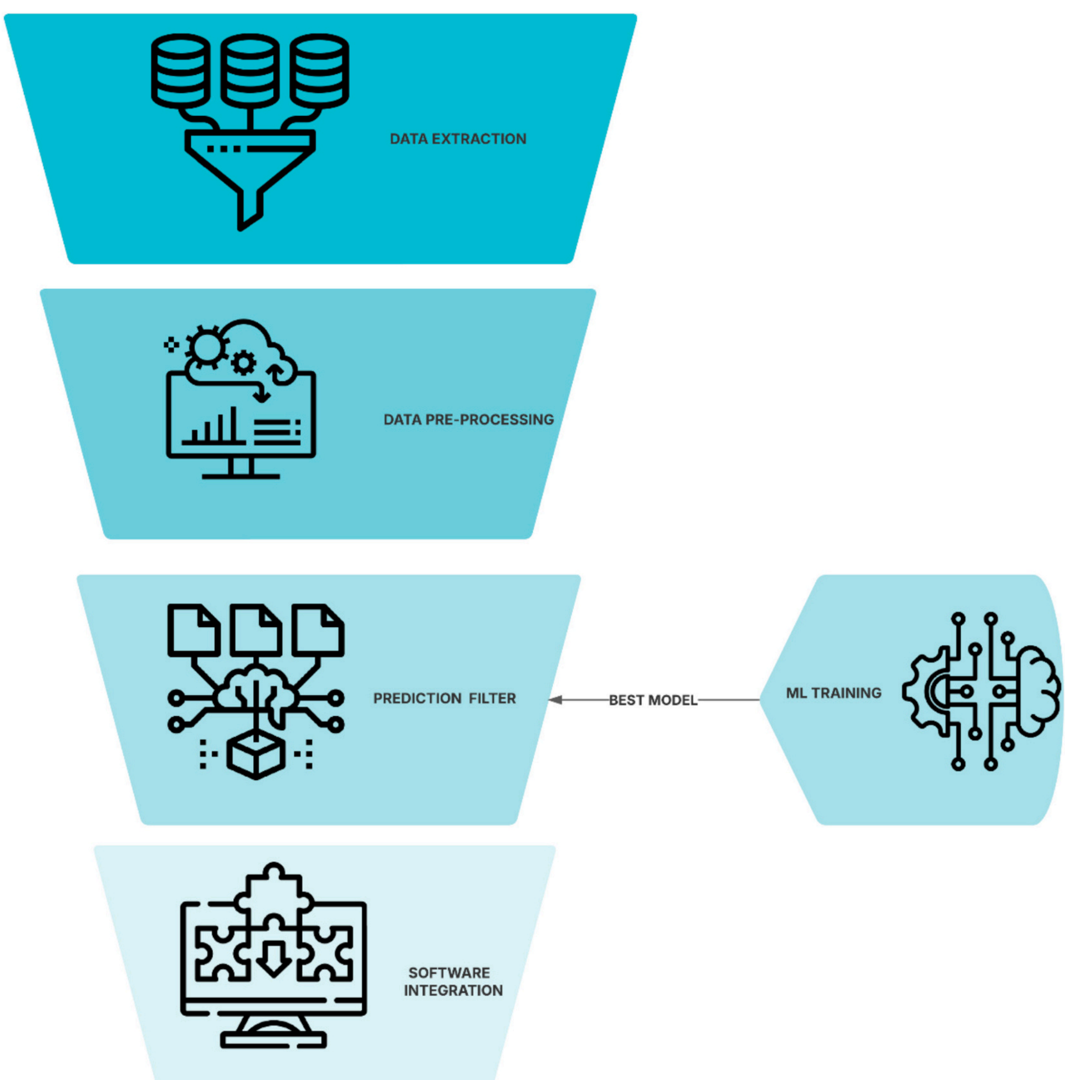

Fig. 4. Global vision of the solution.

AUTODESK NAVISWORKS **Rapport de conflits**

Analyse 1

						Elément 1					Elément 2				
Image	Nom de conflit	Etat	Distance	Emplacement de la grille	Description	ID d'élément	Elément Nom	IFC Type IfcClass	IfcClassificationReference Identification	Matériau Nom	ID d'élément	Elément Nom	IFC Type IfcClass	IfcClassificationReference Identification	Matériau Nom
	REF - IMPLANTATION SITE - IfcReinforcingBarType	Nouveau	-5.568		Dur	*ID d'élément*: 3ksr0xP2T1Xg1NcRz6iCDQ	Solide topographique:TOPOGRAPHIE:19992166	IfcBuildingElementProxyType		Terra	*ID d'élément*: 1$zJmoHN92I9Ll8o_$FXBs	Barre d'armature:ANC.R-1 : Forme Forme d'armature 1:1354479: 1	IfcReinforcingBarType		Rebar - ASTM A615M - Grade 420 - Billet Steel
	REF - IMPLANTATION SITE - IfcWallType	Nouveau	-2.152		Dur	*ID d'élément*: 3ksr0xP2T1Xg1NcRz6iCDQ	Solide topographique:TOPOGRAPHIE:19992166	IfcBuildingElementProxyType		Terra	*ID d'élément*: 1vu7eIcYb1$gfGTo4YiJJW	Mur de base:MUR BÉTON 350 MR10:477797	IfcWallType	B1010210	Concrete - Cast-in-Place Concrete - 30 MPa
	REF - IMPLANTATION SITE - IfcGeographicElementType	Nouveau	-1.856		Dur	*ID d'élément*: 3ksr0xP2T1Xg1NcRz6iCDQ	Solide topographique:TOPOGRAPHIE:19992166	IfcBuildingElementProxyType		Terra	*ID d'élément*: 3Nq37eNBT0he06kIetM3GN	Solide topographique:Solide topographique 1:645116	IfcGeographicElementType		
	REF - IMPLANTATION SITE - IfcColumnType	Nouveau	-1.049		Dur	*ID d'élément*: 3ksr0xP2T1Xg1NcRz6iCDQ	Solide topographique:TOPOGRAPHIE:19992166	IfcBuildingElementProxyType		Terra	*ID d'élément*: 3L3kly9EXCOA$UWQAB_Otn	_Col Bet_CAN_TYP:450_x450:757275	IfcColumnType	B1010240	
	REF - IMPLANTATION SITE - IfcSlabType	Nouveau	-0.818		Dur	*ID d'élément*: 3ksr0xP2T1Xg1NcRz6iCDQ	Solide topographique:TOPOGRAPHIE:19992166	IfcBuildingElementProxyType		Terra	*ID d'élément*: 3FNPrkRI12d9LUCIcbHm4H	Fondations 2:Fondations 2:653213	IfcSlabType		
	REF - IMPLANTATION SITE - IfcFootingType	Nouveau	-0.200		Dur	*ID d'élément*: 3ksr0xP2T1Xg1NcRz6iCDQ	Solide topographique:TOPOGRAPHIE:19992166	IfcBuildingElementProxyType		Terra	*ID d'élément*: 1JBFdKjGXAXeBHBT4xQi7c	Semelle filante:Semelle filante 400x850-SM-2:534679	IfcFootingType		03 31 00 Concrete - Cast-in-Place Concrete
	REF - IMPLANTATION SITE - IfcDoorType	Nouveau	-0.042		Dur	*ID d'élément*: 3ksr0xP2T1Xg1NcRz6iCDQ	Solide topographique:TOPOGRAPHIE:19992166	IfcBuildingElementProxyType		Terra	*ID d'élément*: 3V956uAc9BDvDqPk0XXkxp	Stainless Steel-Bilco-Sand Blast Finish-304	IfcDoorType		
	NIVEAU 2 - IfcWallType	Nouveau	-2.384		Dur	*ID d'élément*: 2r2jcinNv1Tg6y4gS6zdeg	Sol:P1 - DALLE - 280:17037655	IfcSlabType	B1010	00 - Beton structural 254mm	*ID d'élément*: 1vu7eIcYb1$gfGTo4YiJM1	Mur de base:MUR BÉTON 350 MR:477956	IfcWallType	B1010210	Concrete - Cast-in-Place Concrete - 30 MPa
	NIVEAU 2 - IfcSlabType	Nouveau	-2.321		Dur	*ID d'élément*: 2EuosQ$pr2pwKCj9H$t13s	Mur de base:MF1 - MUR BÉTON COULÉ 350mm:21388393	IfcWallType	B2010	00 - Beton structural fini	*ID d'élément*: 0qzkzuoSb2h9rakIReftI1	Sol:DALLE 280:530919	IfcSlabType		03 31 00 Concrete - Cast-in-Place Concrete
	NIVEAU 2 - IfcColumnType	Nouveau	-0.476		Dur	*ID d'élément*: 1kRZ4XJBb2v9MaKPOayKRP	Sol:P3 (SOFFITE) - soffite ventilé - 500:20126307	IfcSlabType	B1010	MEF_00 - Revetement metallique VPS_BLANC	*ID d'élément*: 2mb_Ycu956Cg_Zw7SzrwwO	_Col Bet_CAN_TYP:400x700 CB1:425877	IfcColumnType	B1010240	03 31 00 Concrete - Cast-in-Place Concrete
	NIVEAU 2	Nouveau	-0.236		Dur	*ID d'élément*: 2GfvUSNun59ARv0oc_Mvey	Mur de base:MF1 - MUR BÉTON COULÉ 350mm:21284927	IfcWallType	B2010	00 - Beton structural fini	*ID d'élément*: 2j09Rib99EpvotfBCEAHJ0	Sol:DALLE 175 VAR.:818143	IfcSlabType		03 31 00 Concrete - Cast-in-Place Concrete
	NIVEAU 2 - IfcBeamType	Nouveau	-0.092		Dur	*ID d'élément*: 3Q_OrxxuP6WBCmIfqlDX0H	Mur de base:MX - MUR ENV_VIDE (SOFFITE):21404410	IfcWallType	B2010	00_ Membrane PARE-AIR	*ID d'élément*: 3nJjKHwXjFt8cQkPm1WJVP	_CISC HSS_Carre_Rectangle_CAN:HSS 254x152x6.4:1100930	IfcBeamType	B1010300	05 12 23 Metal - Steel - ASTM A500 - Grade C - Rectangular and Square

Fig. 5. Example of traditional Navisworks HTML conflict report.


```
},
"Conflit186": {
  "Distance": "-0,103",
  "Status": "New",
  "TestName": "S EL",
  "Description": "Dur",
  "OverlapVolume": "0,970 m³",
  "ContactSurface": "1,31 m²",
  "PenetrationDepth": "0,742 m",
  "Center": {
    "X": "1557269,487",
    "Y": "17008739,421",
    "Z": "317,163"
  },
  "AngleBetweenVectors": "40,75°",
  "Comments": "",
  "Properties": {
    "Element 1": {
      "Nom": "Sol",
      "Type": "Sols: Sol: DALLE 280",
      "Matériau": "03 31 00 Concrete - Cast-in-Place Concrete",
      "Fichier source": "11058_S_GEN.rvt",
      "Famille": "Sol",
      "Catégorie": "Sols",
      "Niveau": "LcRevitElement(Level \"NIVEAU 2\", #318953)",
      "Structure": "True"
    },
    "Element 2": {
      "Nom": "Tt_CNR_Coude de conduit EMT_Sans raccords(P)",
      "Type": "Raccords de conduits: Tt_CNR_Coude de conduit EMT_Sans raccords(P): EMT_Sans_Raccords",
      "Matériau": "",
      "Fichier source": "11058_E_EL.rvt",
      "Famille": "Tt_CNR_Coude de conduit EMT_Sans raccords(P)",
      "Catégorie": "Raccords de conduits",
      "Angle": "1,5707963267949",
      "Niveau": "LcRevitElement(Level \"NIVEAU 2\", #3822981)",
      "Structure": "False"
    }
```


Fig. 6. Example of a conflict data extracted in JSON.

Subsequently, information about the two elements involved (*Item1* and *Item2*) is collected, including general categories (e.g., Object, Material) and specific details (e.g., element name, material type), corresponding to *PropertyCategory* and *DataProperty* in the diagram.

Additional geometric information (*GeometryData*) is calculated to characterize the conflicts more comprehensively. This includes overlap volume, contact area, angles between elements, and penetration depth, derived from the bounding boxes of the elements. While these measures provide approximate rather than exact values, they offer meaningful insights into the spatial relationships and severity of conflicts, supporting both visualization and subsequent machine learning tasks.

Finally, all data are consolidated into a structured set (*ExtractPropertyfromClash*), combining raw attributes, calculated geometry, and user-specified element details. This structured output is exported as a JSON report (Fig. 6), which can be pre-processed for training supervised learning models and also serve as input for the final plug-in's conflict status prediction module.

4.2.2. Preprocessing

In this section, we present the data enrichment strategies, and the preprocessing steps applied prior to training.

4.2.2.1. Dataset development and enrichment. Preprocessing begins with loading conflict data from JSON files, including numerical attributes (e.g., distances, overlap volumes) and categorical attributes (e.g., conflict categories, disciplines). Data are structured in a table where each row represents a conflict, with columns for numerical values, categorical identifiers, and binary indicators (e.g., structural element presence). Missing numerical values are imputed using the median, and all numerical features are normalized to a [0,1] range. Categorical variables are encoded numerically to ensure compatibility with machine learning models.

Additional geometric and contextual features are generated from spatial relationships and element properties. Boolean indicators capture whether specific conditions are met, while features derived from geometric properties and coordination rules include *SignificantOverlap, LargeContactSurface*, and *PenetrationDepth*. Combined spatial-contextual features include *IntentionalPenetration*, capturing perpendicular MEP penetrations, *FabricationTolerance*, flagging acceptable clashes during fabrication, *FinishingOverlap*, for minor architectural interferences, and *MinorIntraDiscipline*, isolating clashes within the same discipline. Each element's discipline is automatically determined from its category using predefined rules based on Revit nomenclature extracted via tools such as Dynamo (Fig. 7).

Using these derived disciplines, a *ClashType* is computed to characterize the nature of each clash—for example, a Structural–Architectural or Ventilation–Plumbing clash. Additional binary features such as *MinorStructuralClash*, *MinorMEPPenetration*, or *UnacceptableClash* are defined according to coordination rules to identify minor or critical clashes based on practical criteria. These variables enrich the representation of each clash significantly, providing a mix of geometric, technical, and contextual insights that are essential for classification.

4.2.2.2. Data preparation for training. Following preprocessing (Fig. 8), the dataset was prepared for machine learning integration through two main steps: transformation of categorical variables and construction of the target label. Categorical textual variables (e.g., Category1, Category2, ClashType, Discipline1, Discipline2) were processed using one-hot encoding, converting each unique value into a binary feature (e.g., Discipline1_Structural, Discipline1_Architectural), ensuring that the model interprets categories without introducing artificial ordinal relationships. All numerical, binary, and one-hot encoded features were then concatenated into a single, fully numeric dataset. Normalization parameters and the complete set of encoded feature names were preserved for deployment consistency.

The binary target label (Label) was derived from Navisworks clash statuses ("New," "Active," "Verified," "Approved," "Resolved") and manually refined by two experienced BIM coordinators. Clashes deemed critical (Label = 1) corresponded to unresolved or potentially problematic statuses, while resolved clashes (Label = 0) were considered non-critical. Coordinators applied domain-specific heuristics, including spatial impossibilities, clearance violations, and other coordination errors, to ensure labeling reflected practical project needs. Disagreements were resolved collaboratively, and unrecognized statuses were conservatively labeled as critical to maintain caution.

Finally, the dataset was partitioned into a feature matrix (X), containing all numeric, binary, and encoded features, and a target vector (y), containing the binary labels. The resulting normalized, feature-rich dataset is structured for training supervised models to predict the criticality of clashes in BIM coordination workflows.

4.2.3. Machine learning model

This section presents the methodology followed to identify the most suitable classification model for our task. The approach includes a review of existing solutions explored in related experimental contexts, followed by in-depth evaluations of a selected subset of models chosen for their relevance to the problem domain.

The dataset exhibited a moderate class imbalance between critical (real) and non-relevant clashes. To ensure a robust performance assessment, we employed a 5-fold cross-validation scheme. Model performance was evaluated using standard classification metrics, including accuracy, F1-score, recall, and the Area Under the Receiver Operating Characteristic Curve (AUC-ROC). In addition, threshold tuning was performed to optimize the F1-score, reflecting the practical need to maximize detection of relevant clashes while minimizing false positives in a domain-specific filtering context.

4.2.3.1. Exploratory research. In the context of automatic clash classification in 3D BIM models, an in-depth review of the scientific literature and existing solutions within the technical documentation of various

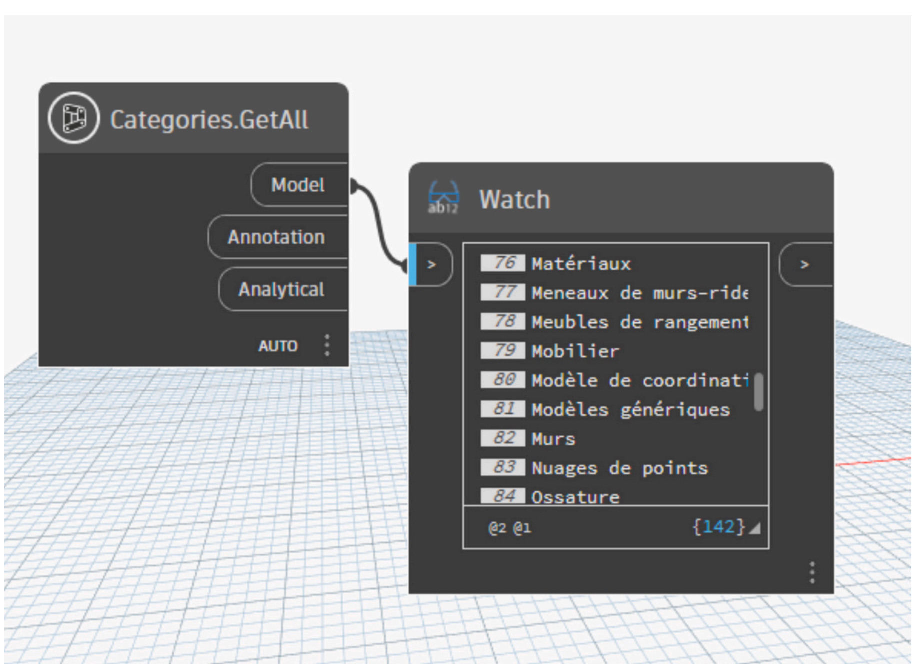

Fig. 7. Example of Revit category list extraction in Dynamo.

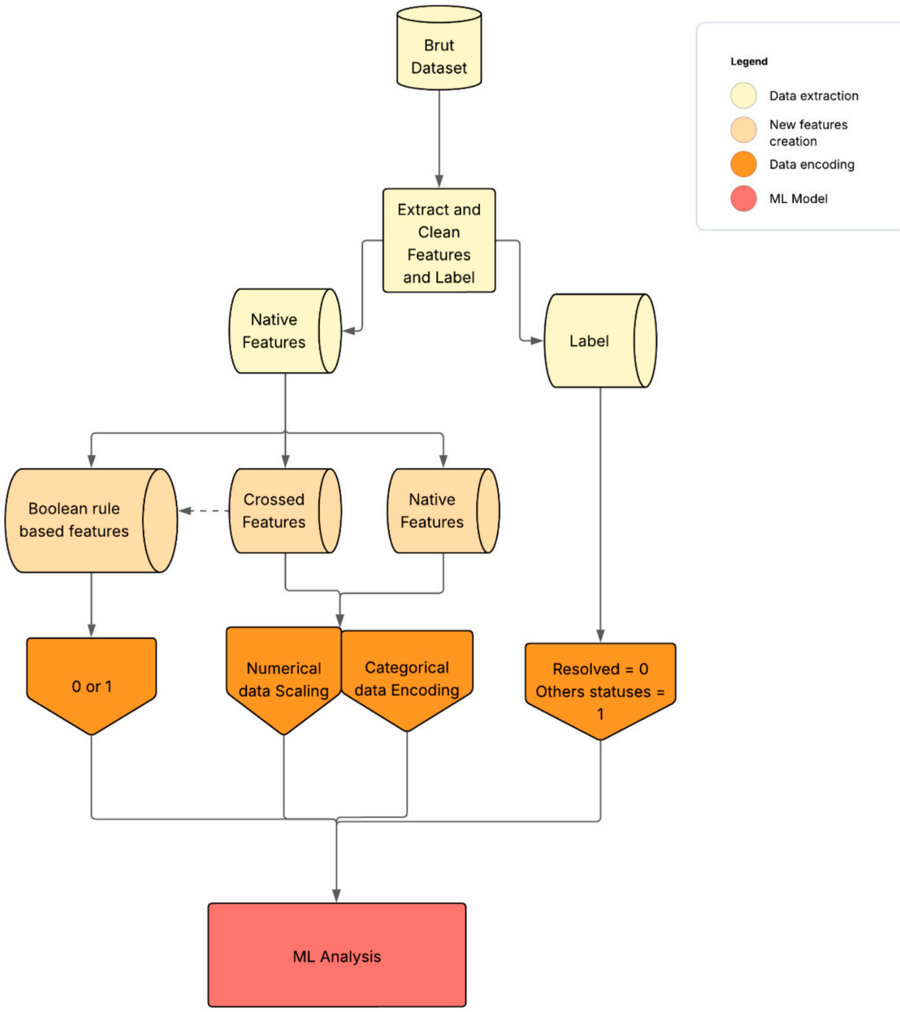

Fig. 8. Simplified diagram of the preprocessing of extracted data.

platforms allowed us to define a strategic, narrowed-down list of models. This selection considers:

- The type of data handled (text, tables, images),
- The complexity of the task (severity classification, grouping by conflict type, etc.),
- The maturity of models within the BIM context (compatibility with Navisworks outputs).

Initially, we focused on scikit-learn models [27], which are known for their ease of implementation and effectiveness when chosen in an appropriate context. The classification models *Random Forest*, *XGBoost*, and *SVM*, which are ensemble models, as well as *SVC*, are initially preferred for their robustness and ease of deployment on data extracted from Navisworks model conflicts. A *K-Means* classification model, although unsupervised, is considered strategic for uncovering patterns that may not be immediately apparent in the datasets. *MLP* (Multilayer Perceptron) offers a progressive step toward deep learning without introducing excessive complexity. *BERT*, a natural language processing model, is selected to handle object names and comments within conflicts (something that traditional models are not equipped to do) and for its accessibility via open-source platforms like Hugging Face. Finally, deep learning architectures could potentially enable the effective combination of images, text, and raw data for advanced contextual classification. The main characteristics of each model are summarized in Table 6.

Several advantages and limitations for each model emerge from the

Table 6
Main characteristic of the selected models.

Model	Type	Main Feature	Key References
Random Forest	Classical ML	Robust, noise-resistant, good for tabular data, interpretable	Hu & Castro-Lacouture [9], scikit-learn [27]
XGBoost	Advanced ML	High performance, handles imbalanced data, fast, accurate	xgboost.ai
SVM	Classical ML	Good on small datasets, efficient for well-separated classes	Ahmadpanah [11]; scikit-learn [27]
MLP	Light Deep Learning	Entry-level DL model for tabular data or text embeddings	Scikit-learn [27]. MLPClassifier
BERT	NLP (Natural Language Processing)	Best for text analysis, captures meaning in names/reports	Hugging Face
K-Means	Unsupervised Clustering	Useful to discover similar clash groups	scikit-learn [27] Harode et al. [8]
Deep Learning (fusion)	Multimodal DL	For text + image + tabular data fusion, an advanced and comprehensive approach	TensorFlow, PyTorch R.adegun [28]

current context (Table 7). The boundary between real and false conflicts is ambiguous, requiring the extraction of additional details beyond the standard *Clash Detective* reports, such as richer data on conflict

Table 7
Strengths and limitations of the selected models.

Algorithm	Strengths	Limitations	Relevance
BERT	Powerful for textual data	heavy fine-tuning	Low (not selected)
Random Forest	Robust, handles noisy data well	Less suited for very complex data	Medium to high
SVM	Effective on small datasets	Sensitive to poorly standardized features	Medium
MLP	Flexible with structured data	Requires more resources than scikit-learn	Medium to high
XGBoost	High performance, fast on medium datasets	Less intuitive to interpret	High
K-means	Useful for unsupervised clustering	Too many clusters with complex data	Low (not selected)
Deep Learning	Powerful on rich datasets	Requires high computational capacity	Low (not selected)

properties or Revit categories. The data is noisy, imbalanced (with more false than real conflicts), and lacks standardization from one conflict to another, complicating processing. Models based on large language models (LLMs), such as *BERT*, have limitations, including data security risks and a significant need for fine-tuning, while deep learning models demand substantial computational resources, which are incompatible with standard environments (CPU). Clustering, although relevant, generates too many clusters due to the complexity and number of features related to the conflicts, making this approach less viable. These constraints lead to the preference for simpler classification algorithms from *scikit-learn* (e.g., Random Forest, SVM, XGBoost, MLP) [27].

For more advanced tests, we will keep the following three models: MLP, Random Forest, XGBoost.

4.2.3.2. Advanced testing. This section presents an in-depth analysis of the key hyperparameters used for three commonly employed classification models in the context of intelligent clash filtering in BIM: the *MLPClassifier* (multi-layer perceptron), the *RandomForestClassifier* (random forest of decision trees), and the *XGBoostClassifier* (extreme gradient boosting). For each of these models, hyperparameters play a crucial role in controlling complexity, learning, regularization, and overall model performance. The following tables detail the most influential parameters, along with concise explanations of their function. Testing will be conducted on each of the three chosen models to determine the optimal hyperparameters, followed by the creation of an ensemble model incorporating the best-performing hyperparameters.

The *MLPClassifier* is a deep learning model based on a fully connected neural network. It is particularly suited for supervised classification problems where the relationships between variables can be complex and non-linear. Hyperparameters here configure the network's structure, learning strategy, regularization against overfitting, and the batch size used during training. The flexibility of this model allows it to be finely tuned to a variety of datasets (Table 8).

The set of hyperparameters for the *MLPClassifier* allows the model to be calibrated based on the complexity of the data and the desired sensitivity to overfitting. Proper configuration is essential to ensure the stability and effectiveness of supervised learning.

The *RandomForestClassifier* relies on the aggregation of a set of decision trees trained on different subsets of the dataset. This robust method reduces the risk of overfitting while offering good performance in terms of accuracy and generalization. The hyperparameters primarily define the depth of the trees, the number of samples required for splits, as well as the sampling methods for bootstrap. Options are also provided to adjust the handling of imbalanced classes (Table 9).

Thanks to its numerous structural and sampling parameters, *Random Forest* allows for precise adaptation to the constraints of the dataset. Its ability to naturally handle variance issues and provide reliable results makes it a highly valued model in the BIM context.

Table 8
Description of the parameters of the MLPClassifier model.

Category	Hyperparameter	Description
Structure	hidden_layer_sizes	Defines the structure of the network by indicating the sizes of the hidden layers.
Structure	activation	Activation function of the neurons in the hidden layers (relu, tanh, etc.).
Learning	learning_rate_init	Initial learning rate for optimization.
Learning	solver	Optimization algorithm (adam, sgd, lbfgs).
Learning	max_iter	Maximum number of iterations during training.
Learning	early_stopping	Stops training if no improvement is observed on a validation set.
Regularization	alpha	L2 regularization coefficient to avoid overfitting.
Sampling	batch_size	Size of mini-batches for stochastic training.
Miscellaneous	random_state	Sets the seed for reproducibility.
Miscellaneous	beta_1 / beta_2	Internal parameters of the adam algorithm for weight updates.

Table 9
Description of the hyperparameters of the RandomForestClassifier model.

Category	Hyperparameter	Description
Structure	max_depth	Maximum depth of the trees in the forest.
	max_features	Maximum number of variables considered for each split.
	max_samples	Proportion of samples used for each tree (if bootstrap).
Learning	n_estimators	Total number of trees to build.
Regularization	min_samples_leaf	Minimum number of samples in a leaf.
	min_samples_split	Minimum number of samples to split a node.
Sampling	bootstrap	Indicates if sampling is done with replacement.
Miscellaneous	random_state	Controls reproducibility.
	criterion	Function used to measure the quality of a split.
	class_weight	Class weights to handle imbalanced classes.

XGBoost is a gradient boosting method that is particularly efficient, designed to optimize both the speed and accuracy of supervised learning. Its hyperparameters cover a wide range of configurations, from tree depth to weight regularization, as well as sample ratio adjustment and class weighting (Table 10). Its performance is often remarkable on structured datasets like those encountered in complex BIM environments.

XGBoost allows for fine-tuning the trade-offs between bias, variance, and complexity. It is particularly recommended for projects where classification performance and handling class imbalances are priorities.

The training of the *Random Forest* (RF), *XGBoost* (GBX), and *Multi-Layer Perceptron* (MLP) models follows a rigorous methodology to ensure

Table 10
Description of the hyperparameters of the XGBoostClassifier model.

Category	Hyperparameter	Description
Structure	max_depth	Maximum depth of the constructed trees.
	colsample_bytree	Fraction of columns (features) used for each tree.
	subsample	Fraction of samples used for each tree.
Learning	learning_rate	Rate of reduction of the weight of each new tree.
	n_estimators	Total number of trees to train.
Regularization	reg_lambda	L2 regularization applied to weights.
	gamma	Loss reduction threshold to allow a split.
	min_child_weight	Minimum weight of a child node to keep it.
Sampling	subsample	Ratio of random samples per tree.
Miscellaneous	random_state	Sets the seed for reproducibility.
	Objective	Objective function to optimize (e.g., "binary: logistic").
	scale_pos_weight	Weight of positive classes to handle imbalances.

optimal performance in conflict classification. After data preprocessing, the data is split into training and test sets, with proportions tailored to each model to maximize efficiency. A hyperparameter search is then conducted using an automated grid search, which tests different combinations to identify the best-performing configuration. This step is crucial for adapting the models to the specifics of the data and avoiding overfitting. Once the optimal hyperparameters are selected, fine-tuning of the classification threshold is performed to optimize the balance between the "False" and "True" class predictions, taking into account the specific needs of the project. This process ensures that each model is trained in a robust and consistent manner, while considering its algorithmic particularities.

Given the imbalance in the dataset with a higher number of false (non-critical) clashes compared to true (critical) ones a rebalancing strategy was required to prevent model bias. The SMOTE (Synthetic Minority Over-sampling Technique) algorithm was applied to artificially generate new samples of the minority class. A 50/50 class distribution was achieved, ensuring equal representation of both classes during training and improving the model's ability to detect critical clashes without being overwhelmed by false positives.

Using the native *GridSearchCV* function from *scikit-learn* [27] in each of the three selected models, the best hyperparameters for each model were determined (Table 11) and subsequently used to create the ensemble model.

To leverage the individual strengths of the three models, an ensemble model (EM) is built by combining the predictions of *RF*, *GBX*, and *MLP*. This approach relies on a voting or probability aggregation strategy, where each model contributes to the final decision based on its performance. The objective is to compensate for the weaknesses of one model with the strengths of others, for example, by exploiting the robustness of *RF* for noisy data, the ability of *GBX* to capture complex relationships, and the flexibility of *MLP* for non-linear patterns. The ensemble model is trained using the already optimized base models, and a new evaluation is performed to verify the performance improvement. This step demonstrates the added value of the ensemble, which consistently outperforms individual models in key metrics, thus providing a more reliable solution for conflict classification in this project.

All models are created through the same Python pipeline (Table 12), set up in a Google Colab Pro environment, designed to automate all the steps required to train a classification model from data extracted from *Navisworks*, for direct integration into a BIM clash filtering plug-in. This is a production-oriented pipeline, with the goal of generating a robust model and inference files usable in an application environment.

Table 11
Best hyperparameters by model.

Catégorie	Modèle	Hyperparamètres	Valeurs
Structure	MLP	hidden_layer_sizes, activation	(256, 128), 'relu'
	Random Forest	max_depth, max_features, max_samples	20, 0.9, 0.9
	XGBoost	max_depth, colsample_bytree, subsample	30, 0.7, 0.9
Learning	MLP	learning_rate_init, solver, max_iter, early_stopping	0.001, 'adam', 300, True
	XGBoost	learning_rate, n_estimators	0.1, 300
Regularization	MLP	alpha	0.0001
	Random Forest	min_samples_leaf, min_samples_split	1, 2
	XGBoost	reg_lambda, gamma, min_child_weight	0.01, 0, 1
Sampling	MLP	batch_size	'auto'
	Random Forest	bootstrap, max_samples	True, 0.9
	XGBoost	subsample	0.9
Ensemble Configuration	Random Forest	n_estimators	300
	XGBoost	n_estimators	300
Miscellaneous	Tous	random_state	42
	MLP	beta_1, beta_2	0.9, 0.999
	Random Forest	criterion, class_weight	'gini', None
	XGBoost	objective, scale_pos_weight	'binary: logistic', 1

Table 12
Description of the model training pipeline.

Function	Role
classify_element_type_with_llm_rules	Classifies elements according to their disciplinary type based on categories and families.
determine_clash_type	Determines the type of conflict based on the two disciplines involved.
select_json_files	Allows the user to select JSON files containing conflicts.
process_clash	Extracts and processes information from a conflict to create a structured entry in the DataFrame.
load_json_data	Loads all conflicts from JSON files and builds the main DataFrame.
export_raw_data	Exports raw data as a dictionary for later use or verification.
preprocess_data	Adds constructed features, encodes categories, normalizes numerical values, and creates labels.
assign_label	Internal function in preprocess_data that converts status to binary label (1 or 0).
train_model	Trains a VotingClassifier (RF, XGB, MLP) and adjusts the optimal threshold with TunedThresholdClassifierCV.
evaluate_model	Evaluates the model by generating classification metrics, ROC/PR curves, and variable importance.
export_results	Saves the model, metadata, configuration files, and variable importance files.
main	Executes data processing, training, evaluation, and export steps by calling all previous functions.

Specifically, this pipeline begins by reading JSON conflict files, applies disciplinary categorization rules, enriches the data with specific features, and then preprocesses them (encoding, normalization, labeling). It then trains a model (*MLP*, *Random Forest*, *XGBoost*, or ensemble model), dynamically adjusts the optimal decision threshold, evaluates performance on the test set, and exports the files required for deployment: model, encoders, thresholds, feature importance, performance curves, and metadata. Each function within this pipeline is responsible for a specific step, ensuring smooth, traceable, and easily reproducible execution.

This training pipeline does not aim to cover the entire exploratory development cycle but focuses on preparing a stable model that is directly usable in an application context. By producing the necessary inference files, it serves as a bridge between the experimental phase (carried out in advance) and the operational integration phase, particularly within the framework of the Navisworks plug-in developed for this project. Its modular structure also allows for future adaptation to other datasets or new models, ensuring its sustainability and portability.

Four approaches were compared: the Multi-Layer Perceptron (MLP), Random Forest, XGBoost, and an ensemble model combining the three previously mentioned. The goal is to identify the model that offers the best trade-off between precision, robustness, and generalization capability. The evaluation relies on several complementary indicators (Table 13): optimized confusion matrices, ROC and Precision-Recall curves, predicted probability distributions, feature importance analysis, as well as a comparative table of global metrics (F1-score, AUC, precision, recall, etc.). Each model is analyzed in detail, highlighting its strengths and weaknesses. These results aim to justify the choice of the most performant model for operational integration into the Navisworks plugin, ensuring reliable and automated classification of relevant conflicts in a multidisciplinary BIM environment.

Table 13
Description of the model evaluation metrics.

Element	Description
Confusion Matrix	Table showing model predictions against actual values (True Positive, False Positive, False Negative, True Negative). Evaluates classification errors.
ROC Curve	Curve illustrating the trade-off between true positive rate and false positive rate at different thresholds. Measures model discrimination ability.
AUC-ROC	Overall performance measure based on the ROC curve. Indicates the quality of class separation.
Precision-Recall Curve	Curve showing the relationship between precision and recall at different thresholds. Evaluates the balance between these two metrics.
AUC (Precision-Recall)	Overall performance measure based on the precision-recall curve. Reflects the quality of positive predictions.
Feature Importance	Graph ranking variables with the most impact on model predictions. Helps identify the most influential features.
Probability Distribution	Graph showing the distribution of predicted probabilities for each class (False/True). Good separation indicates clear distinction between classes.
F1-Score (Optimized Threshold)	Harmonic mean of precision and recall. Evaluates overall performance, especially useful for imbalanced classes.
F1-Score (GridSearchCV)	F1-Score obtained after hyperparameter optimization via GridSearchCV, measured on a validation set or via cross-validation.
Precision	Proportion of correct positive predictions among all positive predictions for a class.
Recall	Proportion of true positives correctly identified relative to all actual true positives.
Support	Number of actual examples in each class. Indicates data size per class.
Accuracy	Overall proportion of correct predictions across all classes.
Macro Avg	Average of metrics (precision, recall, F1-Score) for each class, without weighting by support.
Weighted Avg	Average of metrics (precision, recall, F1-Score), weighted by support of each class.

The confusion matrices obtained after optimizing the classification thresholds (Fig. 9) allow for the comparative evaluation of the four models tested: the ensemble model (ME), XGBoost (GBX), the multilayer perceptron (MLP), and the random forest (RF).

Overall, the ensemble model stands out with the best results, achieving a total of 2299 true negatives and 4699 true positives, compared to only 50 false positives and 265 false negatives. This model effectively minimizes classification errors, demonstrating excellent generalization ability and a good balance between precision and recall. XGBoost also delivers very good performance, with 2268 true negatives and 4732 true positives. It records 81 false positives and 232 false negatives, which remains low. Although slightly less effective than the ensemble model, particularly in terms of false positives, it maintains excellent overall accuracy. The Random Forest model is close to XGBoost in terms of results but falls slightly behind. It records 2249 true negatives and 4637 true positives, with 100 false positives and 327 false negatives, suggesting a slight tendency to under-classify some positive observations. In contrast, the MLP performs significantly worse. With 1011 true negatives and 2293 true positives, it generates a high number of errors: 161 false positives and 192 false negatives. This imbalance highlights a particular weakness in correctly identifying the negative classes, which could be attributed to model instability on this dataset or increased sensitivity to noise.

In summary, the results demonstrate the advantage of an ensemble model combining multiple complementary approaches. The ensemble model benefits from the strengths of each individual classifier to achieve a more robust and reliable classification, surpassing each of them when considered separately. These results confirm the value of the ensemble approach in a context where precision and minimizing classification errors are priorities.

The graphs representing variable importance (Fig. 10) highlight the most influential features in the decision-making process for each of the models used.

The charts presented in Fig. 10 illustrate the variable importance across each model employed, both at the individual level and in terms of *variable families*. The *ensemble model (ME)* exhibits a relatively balanced contribution from multiple descriptor types. At the individual variable level, *Discipline2_Ventilation* emerges as the most influential, followed by specific object categories such as *Category2_Ossature* and *Category2_Modèles génériques*, as well as a key geometric variable, *Distance*. This distribution indicates a heterogeneous use of input information, integrating both contextual and geometric dimensions.

The aggregated analysis by *variable families* corroborates this observation. While variables from *Category2* appear slightly more prominent, numerical descriptors, discipline-related variables, and conflict types (*ClashType*) also contribute substantially to the model's predictions. This diversity of input types highlights the ensemble model's ability to integrate relevant signals from complementary dimensions, thereby mitigating over-reliance on any single information source.

The *XGBoost (XGB)* model demonstrates pronounced sensitivity to the variable *IntentionalPenetration*, which stands out with an individual importance score exceeding 0.07. Other influential variables include categories such as *Category2_Unknown* and *Category2_Ossature*, alongside discipline descriptors like *Discipline2_Ventilation* and *Discipline2_Electrical*. It is worth noting that generic or unspecified categories (e.g., *Unknown*) may represent instances where modeling information is incomplete or absent. While statistically informative, these variables should not be interpreted as having strong domain-specific significance. In the grouped analysis, the predominance of *Category2* indicates the model's focus on a narrow set of strong signals, which may lead to underutilization of other relevant dimensions necessary for a broader understanding of conflicts.

The *Random Forest (RF)* model similarly emphasizes *Category2* variables, in addition to geometric descriptors such as *Distance* and *AngleBetweenVectors*, and variables related to disciplines. The aggregated importance across *variable families* reveals a notable emphasis on object categories, while also incorporating numerical descriptors, suggesting a more diversified approach to decision-making—though still notably influenced by categorical object features.

In the case of the *Multilayer Perceptron (MLP)*, the most influential variables are primarily binary indicators derived from domain-specific rules, including *Structure1*, *SignificantOverlap*, and *IntentionalPenetration*. These variables reflect predefined interference scenarios aligned with recurrent industry logic. The grouped analysis indicates a dominant contribution from such binary rule-based features, underscoring the MLP's capacity to effectively exploit explicitly defined contextual relationships.

In summary, certain variables—including *Category2*, *IntentionalPenetration*, and *Discipline2*—consistently appear among the most influential across models, affirming their central role in predicting conflict criticality. Nevertheless, each model leverages these features differently. *XGBoost* adopts a selective approach focused on a limited number of highly discriminative variables, while *Random Forest* distributes importance more gradually. The *MLP* relies heavily on structured domain-specific rules.

Ultimately, the *ensemble model (ME)* distinguishes itself by distributing variable importance more equitably across categorical, geometric, and contextual dimensions. This integrative strategy capitalizes on the complementary strengths of its constituent models (*XGB*, *RF*, *MLP*), enhancing robustness by reducing dependence on any single dominant signal and improving adaptability to the diverse interference scenarios characteristic of BIM projects.

The probability distributions predicted by each model (Fig. 11) provide an insight into their ability to effectively distinguish between positive and negative classes. In these graphs, probabilities associated with negative instances (the "False" class) are represented in blue, while those of positive instances (the "True" class) are in red.

The Random Forest (RF), XGBoost (GBX), and ensemble model (ME)

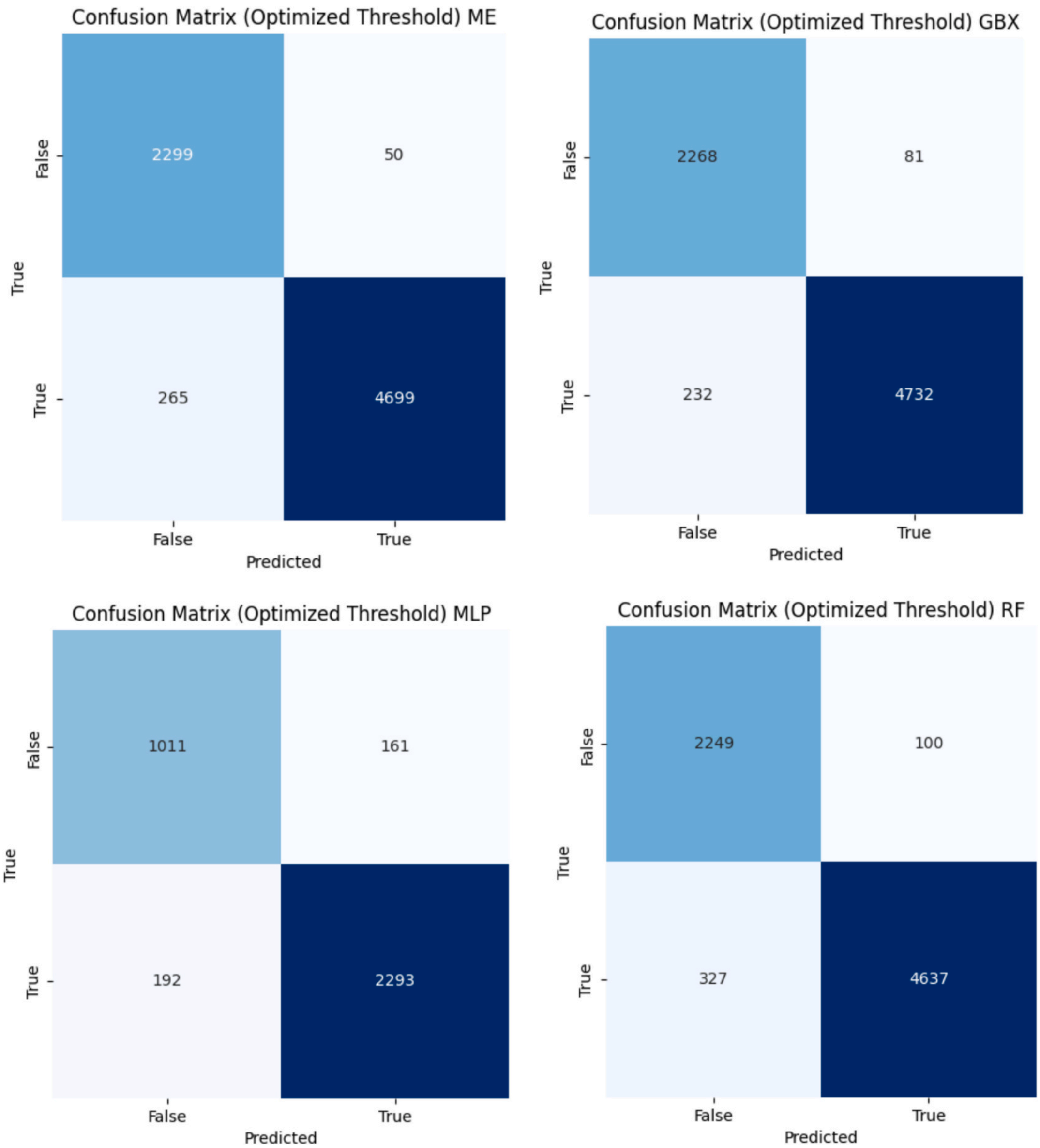

Fig. 9. Confusion matrices of the different training models.

exhibit well-separated distributions, with a strong concentration of probabilities close to 0 for the false class and close to 1 for the true class. This clear separation, combined with a decision threshold set around 0.5 (dashed vertical line), indicates a very good ability to discriminate between the classes. Specifically, the ensemble model stands out with an extremely sharp separation, featuring well-defined peaks and minimal overlap between the two classes, which reflects its robustness in prediction.

In contrast, the MLP model shows a less pronounced distribution. There is more overlap between the classes around the threshold, with a more spread-out and less polarized probability distribution. This suggests that the multilayer perceptron is less confident or less well-calibrated in its predictions, which could lead to greater uncertainty in classifying certain cases. Overall, the analysis of the distributions confirms the previously observed performance: the ensemble model appears to be the most reliable for correctly separating relevant conflicts from irrelevant ones, thanks to its clear and well-structured probabilistic distribution.

The ROC and Precision-Recall curves obtained (Fig. 12) provide insight into the discriminative power of the models, as well as their efficiency in detecting positive instances while minimizing false positives. The ensemble model (ME) stands out with an almost perfect ROC curve, achieving an Area Under the Curve (AUC) of 0.990. This value indicates an excellent ability to separate the positive and negative classes, while simultaneously minimizing false positives and false negatives. The GBX (0.992) and RF (0.987) models also display excellent performance, though slightly behind. In comparison, the MLP model achieves an AUC of 0.962, which is still very satisfactory but indicates slightly less distinct discrimination, particularly in the areas with low false positive rates.

The Precision-Recall (PR) curves further support these observations by emphasizing each model's performance on the positive class—that is, the accurate detection of relevant clashes. The ensemble model (ME) once again achieves the highest area under the curve (AUC), with a value of 0.996, highlighting its ability to maintain high precision even when recall is maximized. XGBoost (GBX) and Random Forest (RF) follow closely, with respective AUCs of 0.996 and 0.994, also demonstrating strong robustness. While MLP remains effective, its PR curve is slightly less regular, with an AUC of 0.981, suggesting a noticeable drop in precision at high recall levels. These findings reinforce the superiority

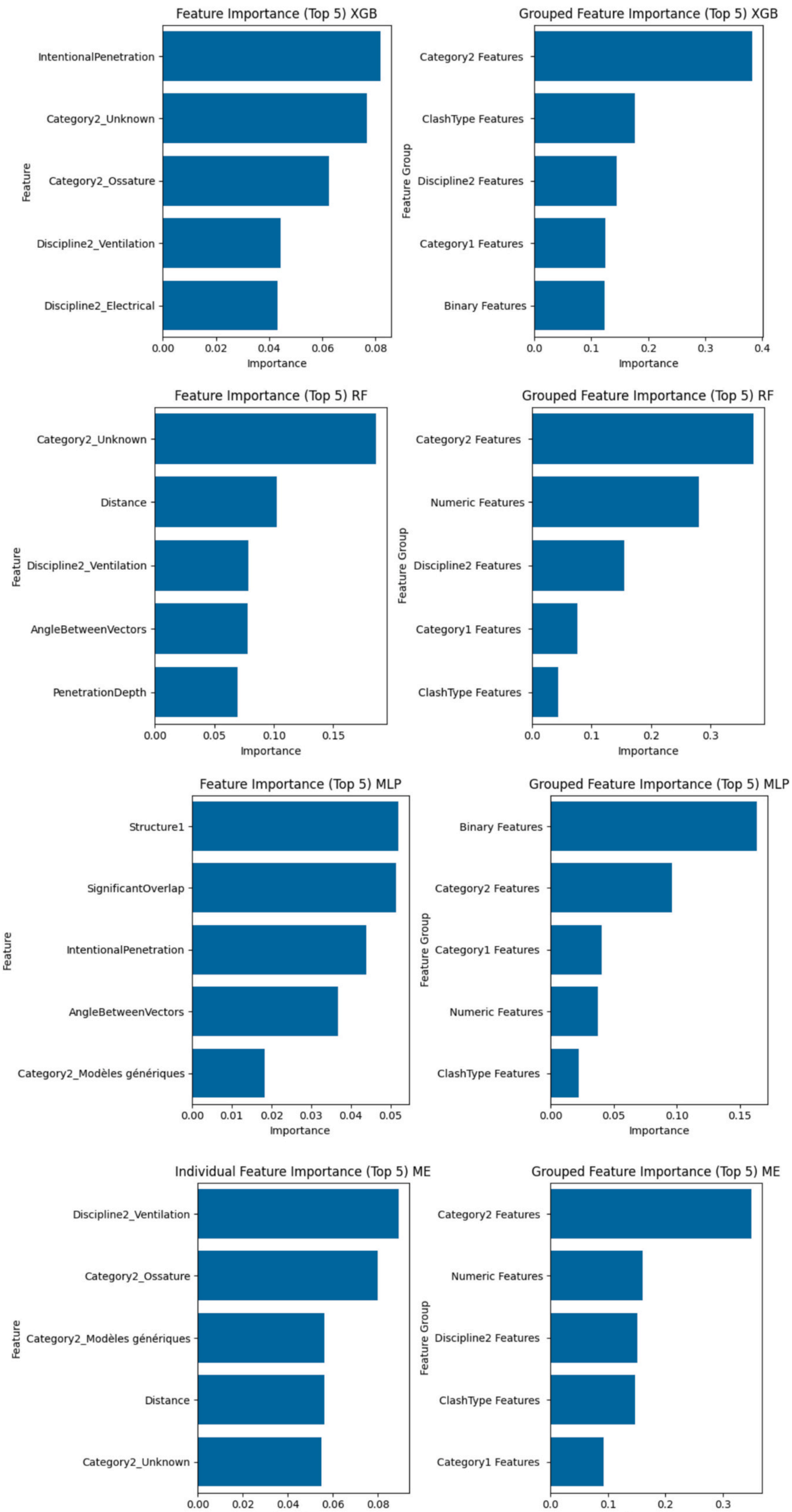

Fig. 10. Variable importance chart by model.

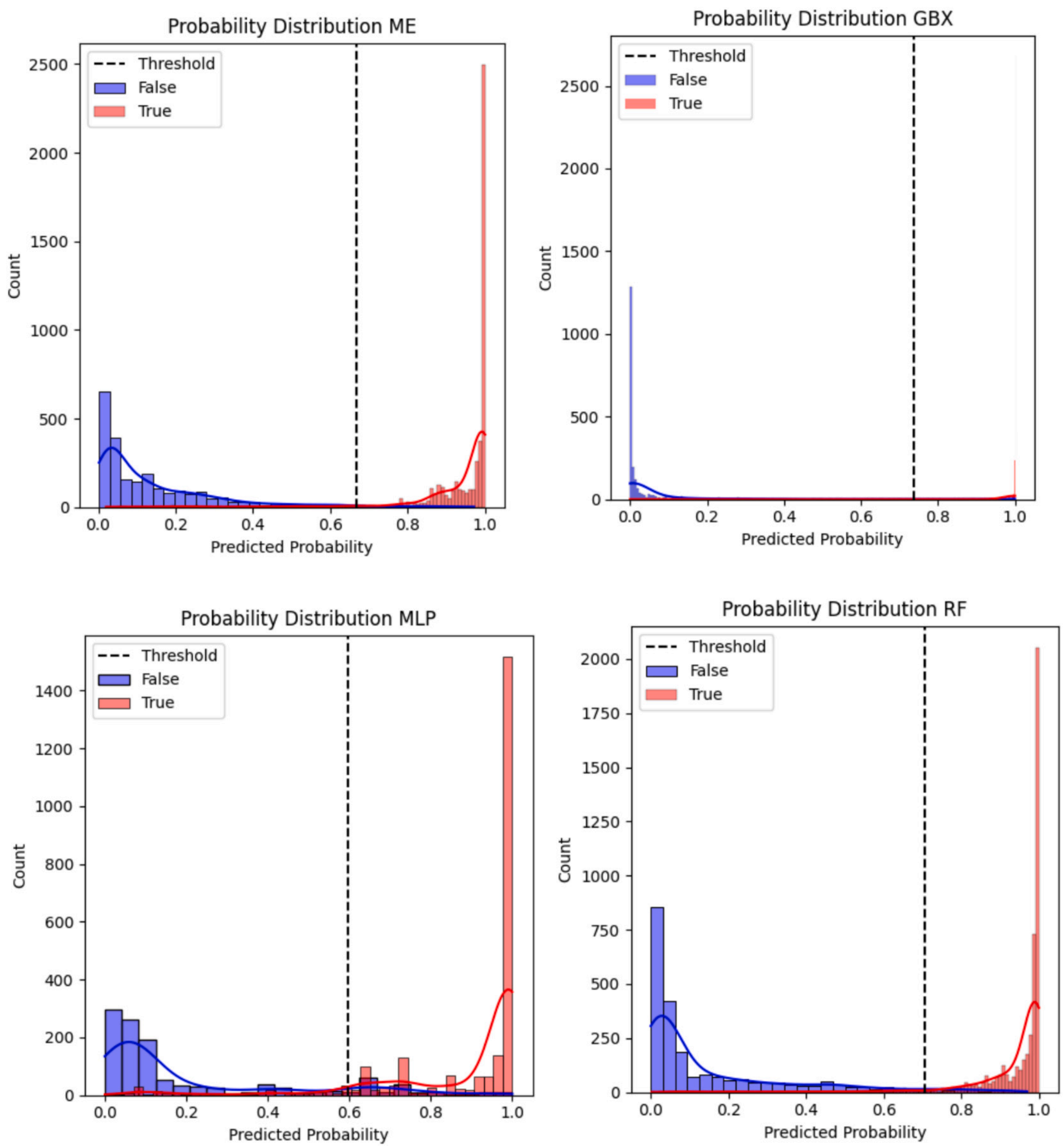

Fig. 11. Probability distributions for the different models.

of the ensemble model, which effectively integrates the strengths of its base learners to achieve an optimal trade-off between sensitivity and precision—a critical requirement for the reliable identification of true clashes.

The performance metrics of the evaluated models are summarized in Table 14. The ensemble model achieves the highest scores across all indicators, including an F1-score of 0.96, precision of 0.96, and AUC-ROC of 0.990, indicating a strong ability to discriminate between relevant and irrelevant conflicts. It also provides the best class-specific accuracy, both for clashes to retain (VC) and clashes to filter (FC), while maintaining a well-balanced recall of 0.91 and overall accuracy of 0.99. The individual models, Random Forest and XGBoost, also demonstrate excellent performance, with F1-scores of 0.945 and 0.956, respectively, and AUC-ROC values exceeding 0.98. MLP, while still performing reasonably well, falls slightly behind across all criteria, most notably with a lower recall of 0.86, potentially indicating challenges in detecting certain relevant clashes. Overall, these results validate the ensemble learning strategy as an effective compromise between robustness and performance, successfully leveraging the complementary strengths of the individual classifiers.

The SHAP summary plots highlight key factors influencing model predictions for clash classification. Geometric features such as *Distance*, *AngleBetweenVectors*, and *PenetrationDepth* consistently show high impact, especially in XGBoost, MLP, and the Ensemble model, confirming their critical role in identifying relevant clashes. Category features like *Isolations des gaines* and *Ossature* are particularly important in the Random Forest model, reflecting the value of typological information. The Ensemble model combines both geometric and disciplinary features, leading to more balanced and reliable predictions. Overall, the analysis confirms that effective clash filtering relies on both spatial metrics and discipline-specific attributes.

The ensemble model has better performance than the three other individual models, with the best F1-score, average precision, and strong recall. This indicates its superior ability to deliver the most accurate predictions without significantly increasing inference time.

Once the models have been trained and evaluated, the results are exported so includes metadata such as feature importance scores, enabling further analysis or integration into automated machine learning pipelines. This export process ensures that the models can be deployed in an operational environment, where they will be used to predict new clashes from raw BIM data. The entire workflow—from training to export—is designed to meet the project's requirements while

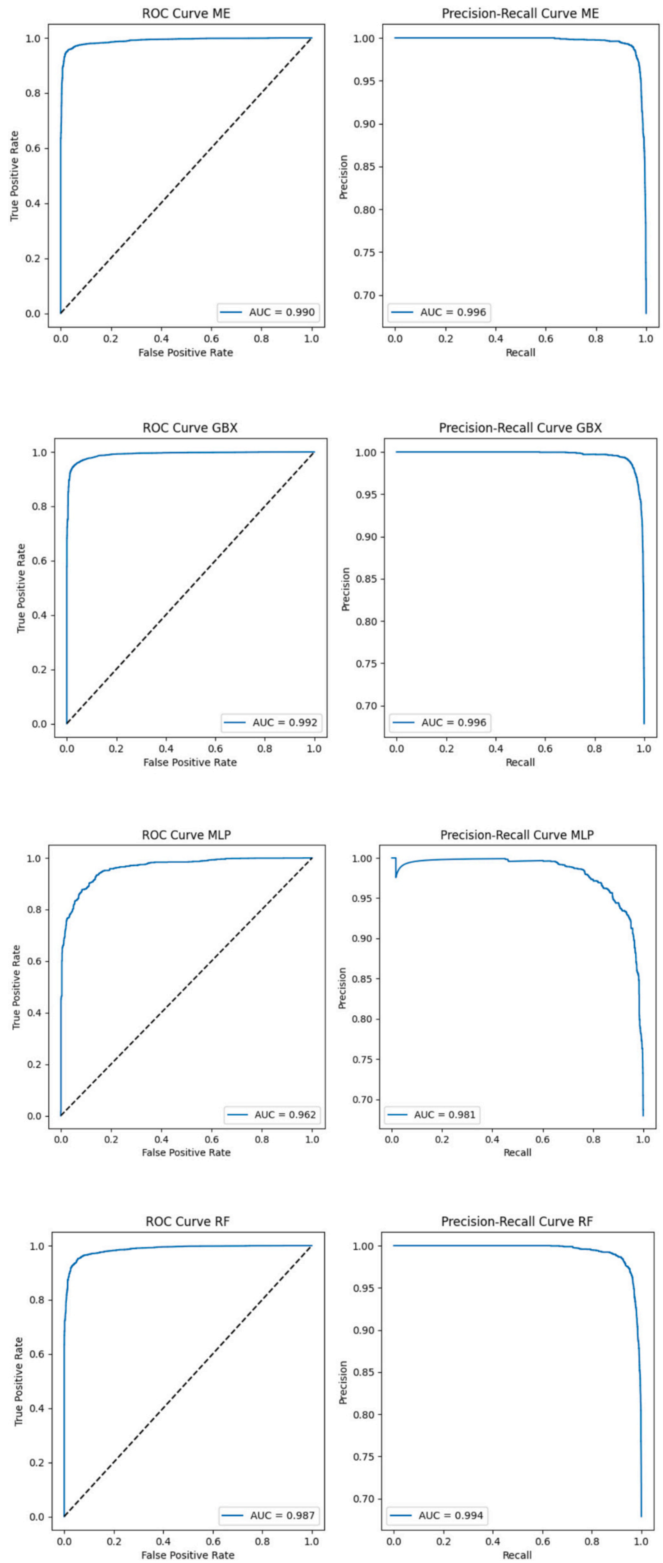

Fig. 12. ROC and Precision-Recall curves for the different models.

offering a scalable and reproducible framework for future implementations.

4.2.4. Prediction model

The Predict module is a lightweight Python architecture integrated into the final plug-in, designed to apply a pre-trained Machine Learning model to new datasets generated from clash detection processes. The user selects a clash file, which is then automatically preprocessed using the same steps applied during model training: encoding of categorical columns, normalization of numerical variables, addition of rule-based features, and identification of the involved disciplines. This preprocessing relies on inference files (such as columnEncoded, scalingParam, and categoryMapping) to ensure full consistency with the saved model (Model.pkl). The predicted output indicates whether a clash is “Resolved” or “Active”.

Once the data has been transformed, it is passed to the model to generate predictions for each conflict. The output is a structured JSON file, where each entry includes the *ClashName* (conflict identifier), the *ClashGroupName* (associated test group), and the predicted *NewStatus* (e.g., “Resolved” or “Active”). This file then serves as an instruction source for the automatic modification of conflict statuses within the plug-in, enabling dynamic and centralized updates of conflict information directly in the BIM user interface.

The prediction pipeline (Table 14) developed in this project is designed to automatically process new conflict files extracted from Navisworks, apply the trained classification model, and save the results in a format compatible with the plug-in. This process follows a sequence of steps: dependency management, loading of inference files (model, normalization parameters, encodings), extraction and preprocessing of new conflict data, model inference, and export of the predictions. Each function within the script is dedicated to a specific task to ensure the robustness, traceability, and reusability of the pipeline across various project contexts. Table 15 summarizes the role of each key function.

This prediction pipeline is designed to integrate seamlessly into the plug-in workflow. It ensures reliable end-to-end execution, from data selection to prediction generation, while automatically handling errors and technical dependencies. Its modular architecture facilitates model updates and deployment on new datasets originating from various BIM projects.

In addition to accuracy metrics, inference times were measured at this point within the Navisworks environment to assess the plug-in’s responsiveness during real-time usage for 11,522 samples. Average inference times per conflict were as follows: Random Forest – 172 **ms**, XGBoost - 92 **ms**, and MLP – 170 **ms**. And Ensemble model regrouping the three of them - 607 ms These values confirm the feasibility of on-the-fly classification without significantly slowing down the coordination workflow.

4.2.5. Final plug-in

The developed plug-in integrates directly within Autodesk Navisworks and aims to enhance the process of detecting and managing conflicts in BIM models by adding a layer of artificial intelligence. This system automates the analysis, classification, and prioritization of detected conflicts within the Clash Detective module, with the goal of facilitating decision-making for professional modelers.

The process begins when the user runs a conflict detection in Navisworks using the Clash Detective tool (Fig. 13). This tool automatically identifies interferences between objects in the model, generating a raw list of conflicts that is often large and difficult to directly utilize. To structure this data, an initial complementary module (add-in) allows for grouping the conflicts according to intelligent criteria: spatial proximity, nature of the involved elements, geometric similarity, etc. This grouping represents the first step in streamlining the information. Once this sorting is done, the user can launch the main plug-in, which orchestrates a complete process of extraction, analysis, and classification of conflicts. Developed in C#, this plug-in automatically extracts the data associated with each conflict (coordinates, object identifiers, element types) and formats them into a standardized JSON file (referred to here as file A). This file serves as the input for an intelligent analysis pipeline.

The extracted metadata is then passed to a Python script that performs rigorous data preprocessing. Using the Pandas and Pandera libraries, the script cleans, encodes, and normalizes the information to

Table 14
Results of the model evaluation.

Model	F1-Score (Optimal Threshold)	AUC-ROC	Avg Precision	Precision FC	Precision VC	Recall	Accuracy
MLP	0.89	0.971	0.88	0.81	0.96	0.91	0.90
Random Forest	0.93	0.978	0.92	0.86	0.97	0.94	0.93
XGBoost	0.95	0.990	0.95	0.91	0.98	0.96	0.95
Ensemble	0.96	0.990	0.96	0.91	0.98	0.96	0.96

Table 15
Description of the prediction architecture pipeline.

Function Name	Role
log_message(message)	Writes a message to a log file for tracking script execution and displays it in the console.
install_package(package, version)	Checks if a dependency is installed and installs it with pip if necessary. Also handles forced reinstallation.
check_and_install_dependencies()	Checks and installs all necessary dependencies for the script to function properly.
classify_element_type_with_llm_rules (…)	Classifies elements of a clash into a discipline (e.g., Structural, Electrical) based on category/family keyword rules.
determine_clash_type(discipline1, discipline2)	Defines a conflict type based on the two involved disciplines (e.g., Structural-Electrical).
process_clash(…)	Extracts and structures key information of a conflict (volume, distance, category, discipline, etc.).
load_inference_files()	Loads the files needed for inference: trained model, normalization parameters, category mapping, encoded columns.
load_new_data(input_path)	Loads the JSON file containing conflicts to predict, extracting each conflict into a pandas DataFrame.
preprocess_new_data(…)	Applies the same preprocessing as during training: feature creation, normalization, categorical encoding, column alignment.
predict(model, X)	Applies the classification model to the preprocessed data and returns the predicted statuses (‘Active’, ‘Resolved’).
save_predictions(…)	Saves the predictions in a JSON file in the expected format, including clash names and their predicted status.
save_error_message(…)	Saves an error message in a JSON file if an exception is raised during execution.
main()	Manages the entire prediction logic: file selection with tkinter, data loading, inference, saving, and error handling.

make it compatible with a Machine Learning model. This model, previously trained on real-world cases, is then called to automatically predict the new status of each conflict.

The predictions are saved in a second JSON file (file B), enriched with the predicted statuses for each individual conflict as well as the groups of similar conflicts. This file is then reintegrated into the Navisworks environment, where the results are displayed directly within the Clash Detective interface (Fig. 14).

Thanks to this seamless integration, the plug-in transforms a purely visual and manual process into a semi-automated workflow, where conflicts are intelligently analyzed, classified, and prioritized. The ultimate goal of this system is to provide professional modelers with a decision-support tool: by prioritizing real conflicts, it facilitates their swift and documented resolution. This system can thus be incorporated into a model review cycle, where identified and classified conflicts are sent to the modeling team for correction, focusing on the most critical issues to address first.

The functions listed in Table 16 correspond to the main steps of the plug-in, ranging from data extraction to status updates, including the invocation of the prediction model. The process involves both C# modules (extraction, interface, updating) and Python modules (analysis, classification).

5. Evaluation

This section presents the evaluation of the proposed plug-in using a combination of technical testing, practitioner feedback, and criteria-based assessment.

5.1. Test in Navisworks

As part of this project, a comparative approach was undertaken to assess the alignment between conflict filtering performed manually by experienced BIM coordinators and automatic filtering carried out using a plug-in we developed.

The goal is to measure the ability of the automatic tool to replicate expert choices, while identifying differences related to grouping logic, implicit business rules, or technical criteria used. The analysis focuses on conflicts detected between various disciplines within the federated BIM project, including architecture (A), structure (S), ventilation (VE), electricity (EL), fire protection (PI), and plumbing (PL). These disciplines were cross-referenced in multiple tests (e.g., A VS S, S VS VE, A VS EL) to assess the relevant types of conflicts from a geometric, technical, or business perspective.

For each test, only the conflicts deemed “true” were retained in both approaches. A detailed analysis was conducted by comparing the volume of conflicts retained, the nature of the elements involved, and the frequency of the cases represented. This comparison allows for the identification of convergences, explainable gaps, and actionable insights for refining the machine learning model used (Table 17).

This comparative analysis aims to highlight the similarities in conflict detection between those identified manually by experienced BIM coordinators and those filtered automatically through a machine learning-based plug-in. The goal is twofold: to validate the ability of the automatic filtering to reproduce business priorities, while also identifying current limitations and potential areas for improvement.

The manually identified conflicts were grouped based on the expertise of the stakeholders, business rules, client expectations, and project issues. In contrast, the automatic filtering relies on an ensemble model trained on manually annotated conflicts, combining Random Forest, XGBoost, and MLP, and incorporating several features of the clash (element type, distance, local density, etc.).

Overall, there is convergence on several critical cases, notably in tests like S VS S or A VS EL, where structural or technical interference conflicts are consistently highlighted. However, discrepancies appear in certain groupings: coordinators, due to their business knowledge, take into account usage logic, clearances, and client contexts. Thus, conflicts like Guardrails vs Rooms are highly prioritized manually but are less detected automatically.

Conversely, automation sometimes highlights frequent but less critical geometric conflicts, according to the coordinators, suggesting room for interpretation. In particular, discrepancies are observed in ventilation conflicts (S VS VE), which are often seen as particularly problematic in practice but are sometimes under-analyzed by the model.

The most common cases of misclassification involve conflicts between secondary technical elements (e.g., flexible ducts vs insulation), which are often considered negligible by coordinators. These errors are

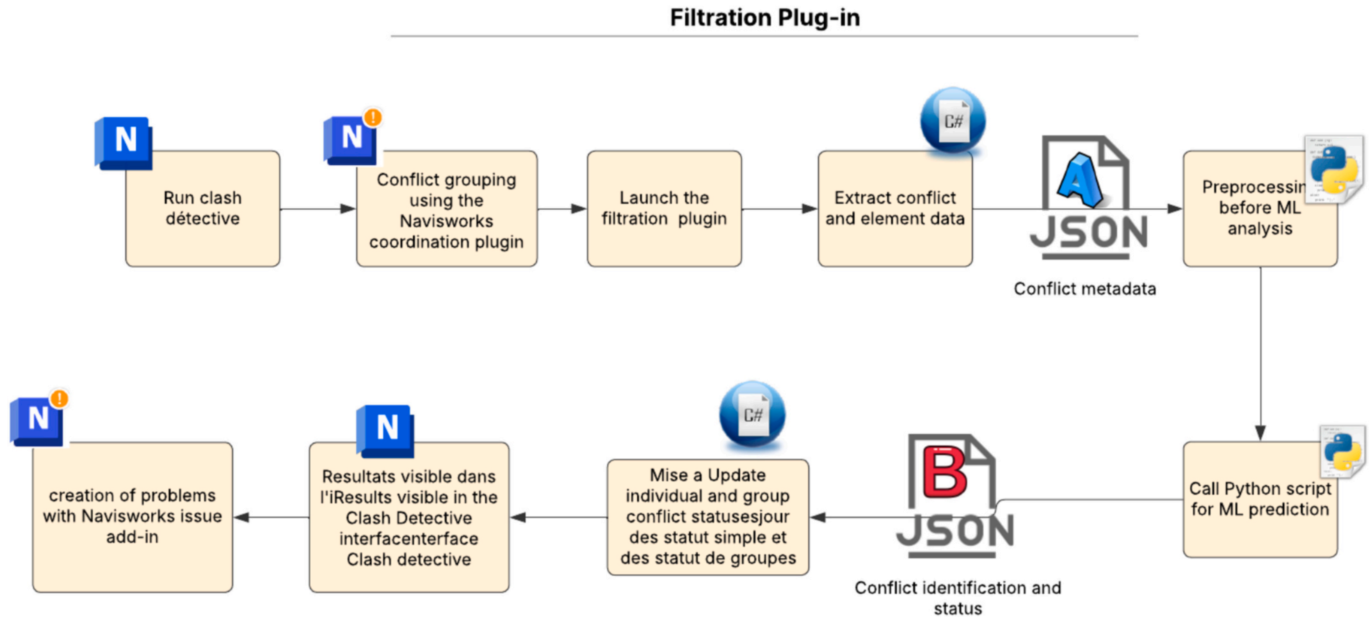

Fig. 13. Simplified diagram of the interference detection process incorporating the filtering plug-in.

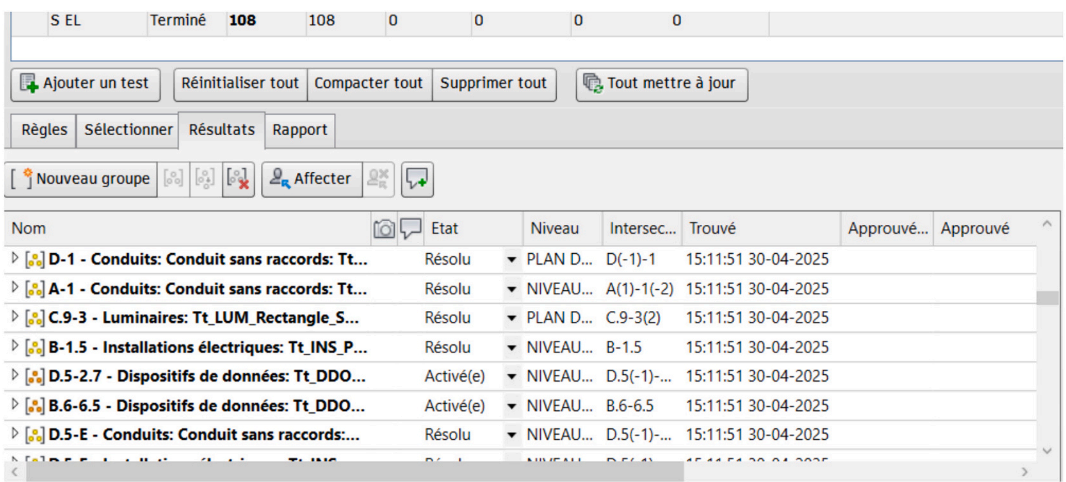

Fig. 14. Example of filtered conflicts (in French).

mostly false positives, reflecting the model's heightened sensitivity to geometric conflicts that have a low business impact.

5.2. *Evaluation by practitioners*

The diagram on Fig. 15 summarizes the quantitative feedback from professionals across four key dimensions of the plug-in: its relevance to field needs, its potential for integration into existing BIM workflows, its usability (inverted here so that higher scores indicate greater ease of use), and the overall appreciation of the solution. The consistency of the ratings reflects a general recognition of the tool's added value, with particular emphasis on its contribution to the intelligent prioritization of conflicts. Slightly lower scores on the usability dimension suggest that further improvements are still needed to enhance user-friendliness.

Table 18 provides a qualitative summary of the feedback given by five professionals who attended the demonstration of the plug-in. Each row outlines the participant's role, their level of BIM experience, and the key strengths and weaknesses noted in their comments. This overview helps contextualize the evaluations according to each participant's position within the coordination workflow. There is a clear interest in the concept of intelligent filtering, alongside concrete observations regarding current limitations, particularly in terms of usability and the reliability of automated processing.

This professional evaluation highlights the strong potential of the conflict filtering plug-in, both in terms of functional value and its ability to integrate into BIM coordination workflows. The feedback gathered validates the relevance of the approach, particularly through the combined use of domain-specific rules and artificial intelligence to better target critical clashes. However, several areas for improvement have been identified, mainly concerning usability, system robustness, and the handling of complex cases. These insights will serve as a foundation for guiding future development, with the goal of delivering a reliable, user-friendly production version that aligns with professional practices.

5.3. *Criteria-based evaluation*

As part of this research, a criteria-based evaluation was conducted to objectively validate the performance and relevance of the developed plug-in. The criteria were defined beforehand within the methodology following the Design Science Research (DSR) approach, and cover the

Table 16
Pipeline of the filtering plug-in architecture.

Function	Role
Execute()	Entry point of the plugin. Launches the overall processing: conflict extraction, property selection, calculations, export, and call to the Python script.
GetClashResults()	Retrieves all conflicts (ClashResult) and conflict groups (ClashResultGroup) present in the model.
ShowFilterDialog()	Opens a user interface to filter conflicts by status, distance, and origin test.
GetAllCategories() / GetAllProperties()	Traverses conflicts to extract all available categories and properties for filtering.
ExtractClashData()	Central analysis function that extracts geometric and semantic properties for each selected conflict.
DisplayResults()	Displays the results in an interactive table (DataGridView) allowing to consult each conflict and its properties.
ExportResults()	Allows the user to export results in JSON or CSV, and to launch automatic prediction.
RunPythonScript()	Calls the predict.py script with the JSON file of conflicts, and retrieves the generated predictions.
UpdateClashStatusesFromJson()	Automatically updates conflict statuses in Navisworks based on predictions.
Geometric functions (CalculateOverlapVolume, CalculateContactSurface, CalculatePenetrationDepth, CalculateAngleBetweenItems)	Calculate advanced geometric metrics from the bounding boxes of conflicting elements.
Interface functions (ShowSelectionDialog, ApplyPreselection)	Manage interactions with the user (selection of categories, properties, filters, etc.).

key aspects of the system: data extraction, domain relevance, model accuracy, integration into the working environment, and scalability.

Each criterion is associated with a measurable indicator and a target objective that enables a factual assessment of the plug-in's effectiveness. The last column of the Table 19 provides concrete evidence of the objectives being met: this includes, for instance, performance measurements for processing time, comparisons with expert annotations, classification reports, and software compatibility tests. These elements demonstrate the plug-in's capability to integrate into existing BIM workflows while efficiently automating conflict filtering.

Table 19 presents the evaluation criteria along with the supporting evidence. Screenshots, execution logs, and excerpts of numerical results further substantiate this analysis in the following sections.

6. Discussion and conclusion

This section discusses the main contributions and implications of the study, situates the proposed approach within the context of previous research, and highlights its practical relevance.

6.1. Comparative analysis with previous work

Several recent studies have explored the potential of machine learning to automate clash filtering in BIM projects. Ahmadpanah [11] introduced an original approach based on image analysis from Navisworks, using the YOLO model to visually identify relevant clashes. This method is relatively easy to implement and relies on easily accessible data, but it does not fully leverage the semantic information embedded in BIM models and remains disconnected from real-world business processes. In contrast, Adegun [28] proposed a more structured approach by combining attribute data extracted from Revit with manual annotations. The resulting system is based on deep learning models capable of delivering accurate predictions in a controlled environment. While this work is notable for its rich datasets and methodological rigor, the tool remains limited to test cases and specific technical disciplines such as MEP. Hu and Castro-Lacouture [9] introduced an interesting framework aiming to formalize the relevance of clashes using predefined business rules. Their method analyzes geometric and topological properties of clashes using models such as Random Forests and SVMs. Although this methodology is logically consistent, it requires substantial data preparation and lacks implementation within standard professional

Table 17
Conflict filtering comparison (Automatic vs Manual).

Tests	Number of true conflicts (manual)	Top conflicts (manual)	Number of true conflicts (automatic)	Top conflicts (automatic)
A VS A	14	Walls vs Walls (7), Walls vs Ceilings (2), Guardrails vs Supports (1)	66	Guardrails vs Rooms (13), Rooms vs Guardrails (7), Storage furniture vs Rooms (6)
A VS S	3	Ceilings vs Load-bearing columns (2), Walls vs Floors (1)	25	Walls vs Floors (6), Walls vs Topographic solid (2), Walls vs Foundations (2)
A VS EL	1	Ceilings vs Electrical installations (1)	32	Rooms vs Electrical installations (6), Ceilings vs Electrical installations (6), Walls vs Lights (6)
S VS S	3	Foundations vs Foundations (2), Walls vs Floors (1)	16	Concrete reinforcement vs Foundations (3), Walls vs Topographic solid (2), Walls vs Concrete reinforcement (2)
S VS PI	1	Floors vs Pipeline (1)	3	Floors vs Pipeline fittings (1), Topographic solid vs Pipeline (1), Load-bearing columns vs Pipeline (1)
S VS VE	1	Topographic solid vs Duct (1)	7	Floors vs Pipeline (6), Topographic solid vs Ventilation outlet (1)
EL VS VE	13	Electrical installations vs HVAC equipment (5), Lights vs HVAC equipment (3), Lights vs Duct insulation (2)	14	Duct vs Lights (3), Ventilation outlet vs Lights (3), Duct insulation vs Lights (2)
PL VS VE	20	Pipeline insulation vs Duct insulation (8), Pipeline vs Duct insulation (4), Pipeline insulation vs Duct (2)	18	HVAC equipment vs Pipeline (6), Duct insulation vs Pipeline fittings (5), Ventilation outlet vs Pipeline insulation (5)
VE VS VE	3	HVAC equipment vs Duct fittings (1), Duct fittings vs Ventilation outlet (1), Pipeline vs Duct insulation (1)	14	Duct insulation vs Flexible duct (5), Duct vs Flexible duct (2), Duct fittings vs Flexible duct (2)

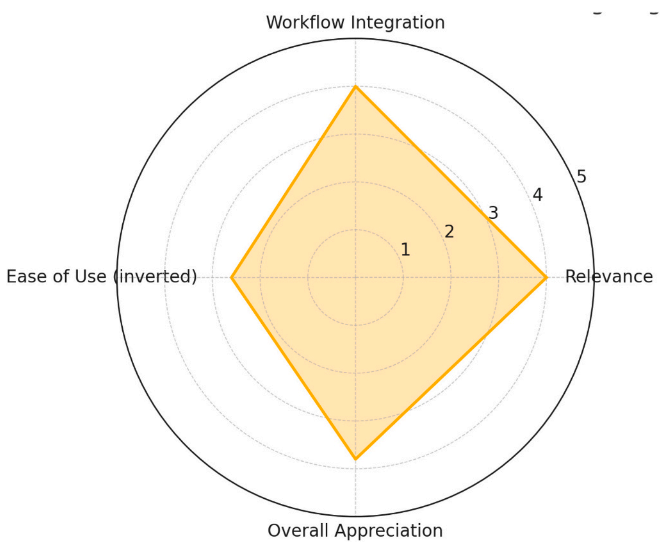

Fig. 15. Quantitative feedback from professionals across four key dimensions of the plug-in.

Table 18
Summary of respondents' profiles and perceptions.

Post	Experience	Overall Feeling
Main BIM Manager	5–10 years	Highlights time optimization and reduction of irrelevant conflicts, but notes a need for comparative tests to validate reliability. Ergonomics could use adjustments.
BIM Integrator	10+ years	Appreciates the relevance of the concept and its integration potential, but points out difficulties during re-import into Navisworks and recommends simplifying manipulations.
BIM Manager	10+ years	Emphasizes the significant contribution of AI to prioritize critical conflicts. Considers the plug-in a good support tool, but notes a need for fine-tuning.
BIM Integrator	10+ years	Recognizes the usefulness of filtering, but warns of a risk of generating artificial errors. The tool still seems to require adjustments for reliable use.
Modeler / BIM Integrator	2–5 years	Appreciates the intelligence brought by machine learning and the overall coherence of the system, although some manipulations still seem unintuitive.

BIM tools. More recently, Harode et al. [8] explored a novel direction by studying the automatic prediction of clash resolution strategies. They used multimodal neural networks combining images, text descriptions, and geometries to assist coordinators in decision-making by directly suggesting suitable resolutions. While promising in terms of decision support, the technical infrastructure is complex and not easily transferable to standard coordination environments.

The project presented in this report stands out from these contributions in several key aspects. It is, to date, the only system offering direct integration within Autodesk Navisworks as a native plug-in. This allows users to launch clash filtering, view results, and export data without leaving their usual working environment. Unlike other approaches that require prior data extraction or transformation, this solution adheres to the digital continuity principles promoted by ISO 19650. Furthermore, the tool is not restricted to a single discipline but has been applied to real projects involving multiple trades, such as architecture, structure, HVAC, and electrical systems, demonstrating broader generalization capabilities. The export of results in structured JSON format, the dynamic reloading of models, and compliance with information security and traceability standards also enhance its operational value.

Our project differs from previous studies by focusing on a custom extraction module that retrieves detailed geometric and textual data directly from Navisworks, unlike previous approaches that relied on basic HTML conflict reports. In particular, we incorporate geometry data (e.g., intersection volume, contact angles) not available in studies like Lin & Huang [29] or Hu & Castro-Lacouture [9]. Furthermore, in contrast to previous work mainly relying on supervised learning, we experimented on a wider range of ML models, including supervised and unsupervised learning. In summary, while previous studies have advanced the field methodologically through annotation strategies, image analysis, or relevance criteria the approach presented here distinguishes itself through software integration, interdisciplinary coverage, adherence to BIM standards, and immediate applicability in real coordination workflows. It represents a significant step forward toward reliable, interoperable, and context-aware automation of quality control in digital construction projects.

6.2. Recommendations and conclusion

The project presented in this paper led to the development of a plug-in integrated into Navisworks, capable of automating clash filtering in BIM models using machine learning. This work addresses a recurrent need in the construction industry: to quickly distinguish true technical conflicts from false positives generated by automated detection

Table 19
Evaluation using predefined detailed criteria.

Category	Criterion	Indicator / Metric	Target Objective	Proof / Verification
Extraction	Extraction Quality	% of conflicts with complete properties	> 98 %	Usable data rate at extraction output of 98 %
Extraction	Extraction Speed	Extraction time for 10,000 conflicts	< 10 s	Extraction time of 5 to 15 s in Navisworks depending on data complexity
Relevance	Alignment with business needs	Gap between automatic sorting and expert sorting	< 10 %	Expert / model comparison table
Relevance	Useful filtration rate	% of false conflicts correctly filtered	> 85 %	91 %
Relevance	True conflict recognition rate	% of true conflicts detected and maintained	> 90 %	98 %
ML Accuracy	Classification accuracy	Accuracy, F1-score	F1 > 0.90	0.95
ML Accuracy	Recall	Recall	Recall >0.85	0.96
Integration	Compatibility with Navisworks	Direct integration (binary)	Yes/No	Yes, plug-in directly included in Navisworks
Integration	Interoperable export (ACC, Newforma)	Number of compatible output formats	≥ 2 formats	Results visible in clash detective and thus opens up possibilities for integration into various workflows
Evolution	Scalability on new projects	Functional on untrained project	Yes/No	Yes
Evolution	Model updates	Ability to reload the model (binary)	Yes/No	Operational "load model" function
Evolution	Structured and secure export	Structured and locked format	Yes/No	Examination of exported JSON files

processes, which often lead to cognitive overload for BIM coordinators. Following a rigorous Design Science Research methodology, the project resulted in the design of a functional artifact tested on real-world public sector projects. The tool combines a structured metadata extractor, a supervised classification model (Random Forest, MLP, or XGBoost), and a user-friendly interface. Experimental results demonstrated a significant improvement in the clash review process, with an F1-score exceeding 0.95 and a recall above 0.95, confirming the plug-in's ability to replicate expert coordinator decisions while reducing processing time from several hours to just a few minutes. This project is also notable for its direct integration of a machine learning solution within the Navisworks environment, without requiring data export or advanced technical skills. This operational orientation, combined with compliance with ISO 19650 standards (traceability, security, interoperability), enhances the industrial and normative relevance of the work.

Although the results obtained are promising, several areas for improvement have been identified to enhance the robustness and relevance of the developed plug-in. A first area concerns the improved utilization of geometric data available in Navisworks. Currently, certain information such as volumes or tolerance distances is not fully exploited. Integrating advanced queries via the COM API could enable more precise extraction of these volume-related features and improve classification quality. Accounting for clearance zones also represents a critical challenge. This involves distinguishing critical interferences from those that are acceptable based on discipline-specific clearance rules. Such processing requires a refinement of contextual evaluation rules, which are currently simplified.

From an algorithmic perspective, it is necessary to increase both the size of the training dataset and the diversity of project types. Working with datasets from various typologies (residential, healthcare, industrial) would improve the model's generalization capability, particularly when faced with unfamiliar structures or modeling conventions. Additionally, the approach could benefit from deeper collaboration with BIM coordinators to refine business rules. Moreover, a limitation of this study is the potential subjectivity in the clash labeling process, as it relied on the collaborative judgment of two BIM coordinators without a formal annotation protocol or quantified inter-rater agreement due to time and project constraints. While domain-driven features were integrated to support the classification, the absence of a standardized protocol may have introduced variability. Future studies could address this by implementing a stricter annotation process with a broader group of experts and inter-rater reliability metrics to enhance objectivity and reproducibility. By tailoring these rules to specific disciplines and project contexts, the model's decisions could better align with real-world practices. Finally, several developments aimed at industrializing the solution should be considered. These include delivering a stable production version with a finalized user interface, actionable analysis logs, and a model update system. In the longer term, integrating a hybrid analysis framework—combining business rules, machine learning, and deep learning (e.g., for automatic interpretation of clash images or critical zones)—would significantly broaden the analytical scope, enabling multi-scale coordination in complex projects.

Nonetheless, some limitations remain. Although Navisworks is widely used in industry, the methods and findings presented in this study may not be fully generalizable to other software environments or project contexts. The implementation relies on features and workflows specific to Navisworks, which may limit the applicability of the approach in different technological settings. Future work could investigate the adaptability of the proposed framework across alternative platforms to enhance its generalizability. Moreover, the size and diversity of the dataset needs further improvement to ensure optimal generalization. Additionally, the integration of more refined domain-specific rules and visual features extracted from the BIM model could enhance the robustness of the model in more varied contexts. Several future directions can be envisioned: expanding the dataset, industrializing the plug-in for large-scale deployment, integrating computer vision, and implementing the solution within a digital twin framework. Future work will focus on validating the proposed approach through real case studies to better illustrate its practical advantages and effectiveness, and on deploying the proposed approach in live project environments, enabling quantitative assessment of its impact on coordination efficiency, clash resolution time, and decision-making processes. Future work could also benefit from a more explicit integration of practitioner feedback, either through hybrid models combining machine learning with rule-based logic, or through active learning frameworks where model predictions are iteratively validated and refined with expert input.

CRediT authorship contribution statement

Rayane Ailem: Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Conrad Boton:** Writing – review & editing, Writing – original draft, Supervision, Project administration, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that there is no conflict of interest and no competing interests.

Data availability

Data will be made available on request.

References

[1] C. Boton, L. Rivest, O. Ghnaya, M. Chouchen, What is at the root of construction 4.0: a systematic review of the recent research effort, Arch. Comput. Methods Eng. (2020), https://doi.org/10.1007/s11831-020-09457-7.

[2] C. Boton, D. Forgues, Practices and processes in BIM projects: an exploratory case study, Adv. Civ. Eng. 2018 (2018) 1–12, https://doi.org/10.1155/2018/7259659.

[3] S. Luo, J. Yao, S. Wang, Y. Wang, G. Lu, A sustainable BIM-based multidisciplinary framework for underground pipeline clash detection and analysis, J. Clean. Prod. 374 (2022), https://doi.org/10.1016/j.jclepro.2022.133900.

[4] S. Mehrbod, S. Staub-French, N. Mahyar, M. Tory, Beyond the Clash: Investigating Bim-Based Building Design Coordination Issue Representation and Resolution. http://www.itcon.org/2019/3, 2019.

[5] E. Tuval, S. Isaac, Online planning and Management of Design Coordination Tasks with BIM: challenges and opportunities, J. Manag. Eng. 38 (2022), https://doi.org/10.1061/(asce)me.1943-5479.0001027.

[6] A.O. Akponeware, Z.A. Adamu, Clash detection or clash avoidance? An investigation into coordination problems in 3D BIM, Buildings 7 (2017), https://doi.org/10.3390/buildings7030075.

[7] I. Bitaraf, A. Salimpour, P. Elmi, A.A. Shirzadi Javid, Improved building information Modeling based method for prioritizing clash detection in the building construction design phase, Buildings 14 (2024), https://doi.org/10.3390/buildings14113611.

[8] A. Harode, W. Thabet, X. Gao, Developing a machine-learning model to predict clash resolution options, J. Comput. Civ. Eng. 38 (2024), https://doi.org/10.1061/jccee5.cpeng-5548.

[9] Y. Hu, D. Castro-Lacouture, Clash Relevance Prediction Based on Machine Learning, 2018, https://doi.org/10.1061/(ASCE)CP.1943-5487.0000810.

[10] D. Kazado, M. Kavgic, R. Eskicioglu, Integrating building information Modeling (BIM) and sensor technology for facility management, J. Inform. Technol. Const. (ITcon) 24 (2019) 441. http://www.itcon.org/2019/23.

[11] H. Ahmadpanah, BIM and Machine Learning Integration in The Design Phase, Using ML to Reduce the Irrelevant Clashes in Clash Detection Report, 2022, https://doi.org/10.31224/2730.

[12] S.K. Baduge, S. Thilakarathna, J.S. Perera, M. Arashpour, P. Sharafi, B. Teodosio, A. Shringi, P. Mendis, Artificial intelligence and smart vision for building and construction 4.0: machine and deep learning methods and applications, Autom. Constr. 141 (2022), https://doi.org/10.1016/j.autcon.2022.104440.

[13] D. Bassir, H. Lodge, H. Chang, J. Majak, G. Chen, Application of artificial intelligence and machine learning for BIM: review, Int. J. Simul. Multidiscip. Des. Optim. 14 (2023) 5, https://doi.org/10.1051/smdo/2023005.

[14] P. Ghimire, K. Kim, M. Acharya, Opportunities and challenges of generative AI in construction industry: focusing on adoption of text-based models, Buildings 14 (2024), https://doi.org/10.3390/buildings14010220.

[15] Vratika Gupta, Vinay Kumar Mishra, Priyank Singhal, Amit Kumar, An overview of supervised machine learning algorithm, in: 2022 11th International Conference on System Modeling & Advancement in Research Trends (SMART), IEEE, 2022, https://doi.org/10.1109/SMART55829.2022.10047618.

[16] UK BIM Alliance, Information Management According to BS EN ISO 19650 Guidance Part 1: Concepts. https://imiframework.org/wp-content/uploads/2019/10/Information-Management-according-to-BS-EN-ISO-19650_-Guidance-Part-1_Concepts_2ndEdition.pdf, 2019.
[17] A. Khorchi, C. Boton, An OpenBIM-based 4D approach to support coordination meetings in virtual reality environments, J. Build. Eng. (2024) 108647, https://doi.org/10.1016/j.jobe.2024.108647.
[18] A.R. Hevner, S.T. March, J. Park, S. Ram, Design science in information systems research, MIS Q. 28 (2004) 75–105, https://doi.org/10.2307/25148625.
[19] J. Livari, J. Venable, Action research and design science research-seemingly similar but decisively dissimilar, Eur. Conf. Inform. Syst. (ECIS) (2009) 8–10. https://aisel.aisnet.org/ecis2009/73.
[20] S. Gregor, A.R. Hevner, Positioning and presenting design science research for maximum impact, MIS Q. 37 (2013) 337–355, https://doi.org/10.25300/MISQ/2013/37.2.01.
[21] Sepehr Abrishami, Jack Goulding, Farzad Rahimian, Generative BIM workspace for AEC conceptual design automation: prototype development, Eng. Constr. Archit. Manag. (2021), https://doi.org/10.1108/ECAM-04-2020-0256.
[22] Saeid Emamgholian, Jacynthe Pouliot, Davood Shojaei, A conceptual framework for automatic modelling and conflict detection of 3D land-use regulation restrictions to support issuing planning permits, Autom. Constr. 156 (2024) 104220, https://doi.org/10.1016/j.landusepol.2023.106972.
[23] Asem Zabin, Vicente A. González, Yang Zou, Robert Amor, Applications of machine learning to BIM: a systematic literature review, Adv. Eng. Inform. 50 (2022) 101419, https://doi.org/10.1016/j.aei.2021.101474.
[24] K. Peffers, T. Tuunanen, M. Rothenberger, S. Chatterjee, A design science research methodology for information systems research, J. Manag. Inf. Syst. 24 (2007) 45–77, https://doi.org/10.2753/MIS0742-1222240302.
[25] M. Bedoiseau, D. Martin, C. Boton, Use of KROQI as a Level-2 common data environment in the French construction industry, Sustainability 14 (2022), https://doi.org/10.3390/su141610455.
[26] C. Preidel, A. Borrmann, C. Oberender, M. Tretheway, Seamless integration of common data environment access into BIM authoring applications: The BIM integration framework, in: EWork and EBusiness in Architecture, Engineering and Construction: ECPPM 2016: Proceedings of the 11th European Conference on Product and Process Modelling (ECPPM 2016), Limassol, Cyprus, 7–9 September 2016, 2017, pp. 119–128.
[27] O. Kramer, Scikit-learn, in: Machine Learning for Evolution Strategies. Studies in Big Data, Springer, Cham, 2016, pp. 45–53, https://doi.org/10.1007/978-3-319-33383-0_5.
[28] I. Adegun, BIM Clash Report Analysis Using Machine Learning Algorithms, University of Alberta, 2024, https://doi.org/10.7939/r3-jnhc-4261.
[29] W.Y. Lin, Y.H. Huang, Filtering of irrelevant clashes detected by BIM software using a hybrid method of rule-based reasoning and supervised machine learning, Appl. Sci. (Switzerland) 9 (2019), https://doi.org/10.3390/app9245324.