LEVERAGING GRAPH BASED SEMANTIC ENRICHMENT FOR ENHANCED AUTOMATED CODE-CHECKING

Tanya Bloch[1], Jiabin Wu[2], Maen Alnuzha[1] and André Borrmann[2]
[1]Technion Israel Institute of Technology, Haifa, Israel
[2]Technical University of Munich, Munich, Germany

Abstract

Despite advances in BIM technology, achieving a comprehensive machine-readable design representation for Automated Code Checking (ACC) remains a challenge. We propose decomposing regulatory checks into sets of semantic enrichment tasks, each tailored with a suitable solution. We demonstrate this approach with a case study on Israeli regulations for security rooms, focusing on two distinct semantic enrichment tasks. We use the Girvan-Newman algorithm to isolate single apartments in a floor plan. A GNN then categorizes security rooms based on the ratio of supporting walls, according to the ranges defined in the regulations. The approach paves the way for a highly automated ACC methodology, with potential implications for diverse regulatory contexts.

Background

The conventional approach to ACC involves two pivotal tasks: the translation of human-written text into machine-readable rules and the establishment of a machine-readable representation of the building design. The ability to translate regulatory documents into hard-coded rule sets is limited due to ambiguous, subjective, or vague regulatory statements, requiring manual interpretation and contextual knowledge (Nawari, 2019; Zhang and El-Gohary, 2017). Despite the prevalence of BIM technology and the adoption of Industry Foundation Classes schema (IFC) as an industry standard, achieving a machine-readable representation of the design that contains all semantic information required for ACC, remains a challenging task that often leads to manual preprocessing of the BIM models.

Moreover, matching regulatory concepts to those represented in the computer-readable design remains highly reliant on human understanding. Recent research proposed implementing ML approaches for ACC, eliminating the need for compiling explicit rule sets (Bloch et al., 2023). However, the efficacy of ML-based code checking is contingent on comprehensive input information. As BIM models often lack the necessary highly rich semantic information, semantic enrichment offers an automated solution for supplementing it.

Complex code clauses typically require querying the topological relationships among building elements, for example, checks of wall continuity across security rooms in Israeli Home Front Command regulations, rules of accessible route and line of visibility in the New Zealand Building Code, checks on support of plaster skins in the International Residential Code, rules related to the travel path and distance in the ADA Standards for Accessible Design and the International Building Code.

Verifying building regulations necessitates a thorough understanding of the complex relationships among various elements throughout the building's structure. Consequently, code checking requires extended data structures or a proof-of-solution describing how the design proves compliance rather than merely fulfilling prescribed criteria. For example, graph processors can be integrated to address implicit spatial properties (Solihin and Eastman, 2015). In this work, we decompose the ACC task into a series of semantic enrichment tasks aiming to automatically supplement needed information for the checking process, focusing on complex requirements that involve relational aspects of the design. We turn to graph-based semantic enrichment techniques to address these specifications, leveraging the inherent relational structure within the design data.

Semantic enrichment

Semantic enrichment of BIM models emerged as a solution for lifting the need for multiple domain-specific Model View Definitions (MVDs) by reasoning over explicitly represented information to derive new facts about the model (Belsky et al., 2016). While some efforts previously focused on querying BIM to extract implicit information stored in models (Borrmann et al., 2006; Mazairac and Beetz, 2013; Wülfing et al., 2014), they often fell short in providing explicit representations of inferred information as part of the building information, limiting downstream applications. The SEEBIM system, proposed by Belsky et al. (2016), leveraged domain expert knowledge, represented as logical statements, to infer new facts and explicitly add them to the model. Wu and Zhang (2019) also implemented a rule-based, iterative method for classifying BIM objects in IFC, leveraging geometric features to identify objects with similar geometric representations. They also recognized that pure geometry, without contextual information is limited in the ability to distinguish between elements with similar geometry. In parallel, semantic enrichment solutions that are based on semantic web technology have been suggested and illustrated as well. In fact, one of the motivations for using the Web ontology, as presented by Pauwels et al. (2017), is utilizing logical inference and proofs, which involves the use of First Order Logic (FOL) to derive new insights from the initial building model. Combining inferred information with the original data represents the core objective of semantic enrichment.

Departing from the FOL methodology, Bloch & Sacks (2018) explored different methods for semantic enrichment. Their work compares a rule-based and a ML-based approach for semantic enrichment by demonstrating both for classifying room types in residential apartments. They found machine learning worked better than rules for this specific task. With the recent developments in the ML domain, and the introduction of Graph Neural Networks (GNNs) (W. L. Hamilton et al., 2017), the applicability and benefit of such models has been demonstrated for the same room type classification problem (Wang et al., 2022). The major benefit of such models is the ability to leverage not only geometric but also contextual information about the building elements. Motivated by limitations in existing BIM software regarding semantic representation and interoperability, the authors proposed leveraging graph-based data structures coupled with enhanced GNN architectures that incorporate both node and edge features. They develop the SAGE-E algorithm and generate the Room Graph dataset containing apartment layouts. Experiments demonstrate the superior performance of SAGE-E over conventional ML methods for room type classification, validating the promise of GNNs for BIM semantic enrichment. This groundbreaking effort establishes a research foundation for applying GNNs to augment BIM semantics. As suggested in Bloch and Sacks (2020), different semantic enrichment tasks can benefit from different approaches to solution. Until now, the more classic graph-based methods remain mostly unexplored in the context of semantic enrichment.

Graph representation of building design information

Building models encapsulate a wealth of architectural and engineering design knowledge, along with specific native design intentions tailored to meet the predefined requirements. In the realm of representing the embedded design properties and the complex topological relationships among building elements, graph structures have emerged as a prevalent tool (Vilgertshofer and Borrmann, 2017).

The adoption of graph-based methodologies facilitates the transformation of building models of structures into networks. These networks comprise nodes and edges, which represent building objects and interrelationships. The graph representation hinges on the specific nature of the building's design structure and the intended application scenarios. Furthermore, the choice of the graph structures depends on the varying objectives of the query. For example, dependency graphs are utilized to predict the clash change components (Hu et al., 2023), and parametric building graphs are employed to match detailing patterns (Abualdenien and Borrmann, 2021).

Method

In this work, we propose a comprehensive, hybrid approach for addressing the challenges of ACC. We demonstrate the approach through a case study, focusing on the effectiveness of graph-based semantic enrichment in addressing the requirements defined by the Israeli Home Front Command regulations for security rooms (Home Front Command, 2010). The experiment described below is structured to fulfill two essential tasks integral to the checking process. The first task is to correctly identify the security rooms in the BIM model. As previous work on the subject demonstrated effective solutions under the assumption that we can isolate individual apartments in the building, we implement community detection to automatically isolate spatial groupings representing individual apartments. The second task addressed in the described experiment is automatically recognizing the ratio of vertical connectivity between the walls of the security rooms. Recognizing the complexity of calculating this ratio, we propose a ML-based classification approach using GNNs. The GNN model was trained to categorize the ratio of vertical continuity into predefined ranges specified by the regulations, streamlining the checking process without the need for precise numeric calculations. The checking process is then complete using a set of logical statements. By implementing diverse computational techniques, we demonstrate an enhanced ACC process with no additional information requirements for the modeler.

Experiment and results

Israeli regulations for security rooms

Security rooms (bomb shelters) are designated spaces within a building specifically designed and constructed to provide protection during emergency situations related to military actions. These spaces are fully constructed from reinforced concrete and equipped with specialized windows and doors that are designed to withstand shrapnel from a blast. As per Israeli Home Front Commands regulations (2010), there are several restrictions for the components of the individual security rooms, such as room dimensions, thickness of slabs and walls, dimensions of openings, etc. However, collections of security rooms are also examined together. The security rooms must be vertically stacked so that the reinforced concrete walls are continuous and reach the foundations. Due to functional design requirements, parking, storage, or other non-protective spaces (security space) often reside above or below the security rooms, where reinforced concrete walls may interfere with the functionality of these spaces. Alterations like wall removals, added openings, or weaker materials in those spaces can disconnect the security room from its base.

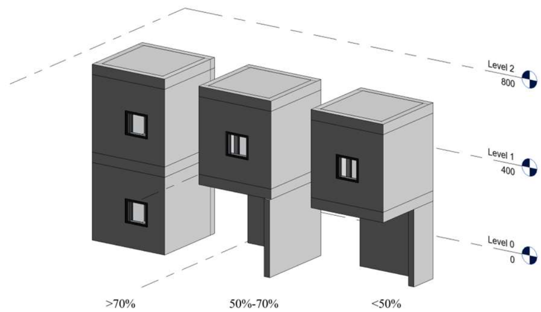

Figure 1: Ratio of vertical continuity of security room walls.

Regulations thus allow for some vertical discontinuity contingent upon certain conditions. One key determinant that dictates additional measures required is the percentage of the security room walls that are continuous and reach the foundation. We will refer to this as the "ratio of vertical continuity."

The regulations delineate three ranges, as illustrated in Figure 1, for the ratio of vertical continuity: spaces with over 70% continuous walls require no further enhancements, effectively extending the requirements for individual security rooms through all associated levels. Spaces with 50-70% vertical continuity and spaces showing less than 50% vertical continuity must incorporate thicker structural walls and slabs and are restricted in the accepted sizes of openings. Note that the ratio of vertical continuity can be calculated per wall individually, thereby dictating additional constraints on the permissible horizontal displacement between walls. And can be calculated for the entire space as the sum of the inner boundaries defined by the walls that compose the space. The ratio of vertical continuity for spaces determines the requirements for the concrete thickness of slabs and walls in the security shaft. The ability to check the compliance of a design to the concrete thickness requirements is the focus of this work. The calculation of the ratio of vertical continuity for walls and spaces involves a large set of guidelines, conditions, and exceptions. The topology of the walls supporting the security rooms, their orientation, sizes of openings and distances between them, and their proximity to existing wall corners, etc., can impact the calculation of the vertical continuity ratio since, under specific conditions, openings must be subtracted from it.

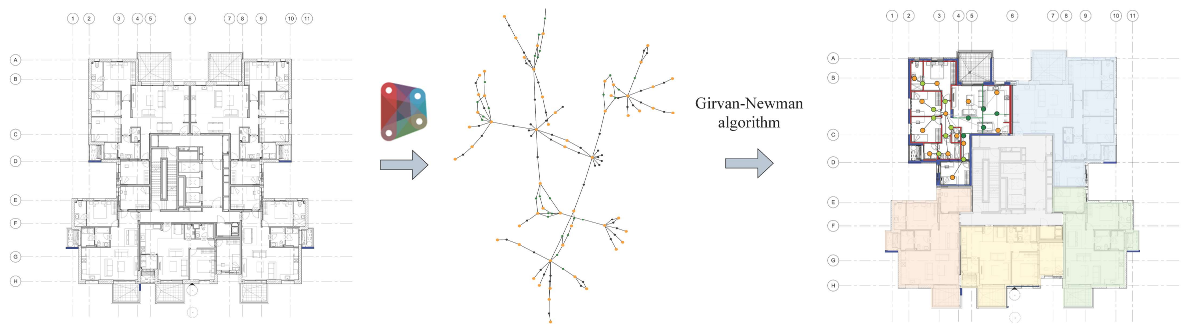

Figure 2: Graph representation of individual floors and community detection for isolating apartments.

To automate this complex calculation based on BIM data, parametric and relational details of building components that are not explicitly present in the models must be considered. While some of the needed information is intrinsic to individual components, some rely on relationships between model elements, which may be more difficult to retrieve.

In this work, we demonstrate a workflow towards a highly automated compliance checking of these regulations using a case study of a residential building that includes overall five security shafts.

Community detection and room classification

As explained in the previous sections, any kind of ACC routine is contingent upon information requirements that are not always given in the BIM model in their explicit form. Spaces are the most primitive building elements that are crucial for many checks. Given that there are no standards or enforced naming conventions, the room names provided by designers become unreliable, which poses challenges for downstream applications. Given that our goal is to avoid introducing information specifications to the design process, the implementation of semantic enrichment to support ACC emerges as a promising solution. In the context of automated space function classification, it has been demonstrated that semantic enrichment for classifying abstract elements, such as spaces, is better addressed with ML than rules (Bloch and Sacks, 2018). Later, a graph-based approach for room type classification has also been demonstrated using GNNs (Wang et al., 2022). In both cases, however, the basic assumption was that we operate within the boundaries of an apartment, and the question of how we can isolate the apartments without additional information requirements for the designers remained open. We propose an automatic solution by implementing a community detection algorithm.

Community detection algorithms are useful for analyzing complex networks to identify subgroups within them. In the context of buildings, where spatial relationships and connectivity are crucial, algorithms for community detection can help uncover distinct spatial groupings indicative of separate apartments. In this test case, we implement the Girvan-Newman algorithm (Girvan and Newman, 2002) for community detection, which was originally designed for social networks. The principle of this algorithm is isolating communities in a graph by iteratively removing edges with high *betweenness centrality,* a measure of how often an edge lies on the shortest path between two nodes. It is important to note that while the Girvan-Newman algorithm is considered relatively slow, it has been determined that it can be effectively used on graphs with less than 1000 edges (Fortunato, 2010). As we are concerned with floorplan layouts that translate into small graphs (in our case, no more than 200 edges), the algorithm provided accurate results within seconds. Other community detection algorithms were not examined.

To implement the suggested technique, we followed the workflow described in Figure 2. Utilizing the API of the BIM authoring tool (Autodesk Revit) and Dynamo (Autodesk, 2022), we collect the building elements that engage in any connection relationship within the spatial layouts, including spaces and doors. Additionally, designers typically use separation lines to divide distinct functional areas within shared spaces. Thus, both the connecting doors, typically situated in boundary walls, and separation lines are recognized as accessible or connecting elements. Furthermore, we perform topological queries on the BIM model to determine the relationships between spaces and accessible elements, forming the undirected edges within the graph structure. For each selected element, the spatial data (floor level) and the associated semantic data (element type) are queried and attached to the related graph node. Within the graph structure representing the whole building model, the elements that belong to each floor are determined by grouping the unconnected subgraphs. As a result, each floor plan of the model is individually processed, leading to the automated generation of accessibility graph structures for our test case.

Once graph extraction was complete, the Girvan-Newman algorithm for community detection was applied on the typical floors, and all individual apartments were correctly identified, after which the previously developed methods for room type classification can be applied. At the end of this stage, we assume all the security rooms in the model are identified and correctly tagged.

Graph Neural Networks for semantic enrichment

Calculating the ratio of vertical connectivity is a complex task. However, the absolute numerical value itself holds minimal significance. What is important is discerning the category of further requirements associated with the ratio—whether it falls within the 70%, 50-70%, or below 50% range. In essence, complex calculation can be avoided by implementing a ML model tailored specifically for this classification task. Given that the calculation heavily relies on the relationships between walls, GNNs emerge as a valuable tool for this classification task. Unlike traditional models, GNNs possess the unique ability to account for and leverage the complex interconnections among building elements.

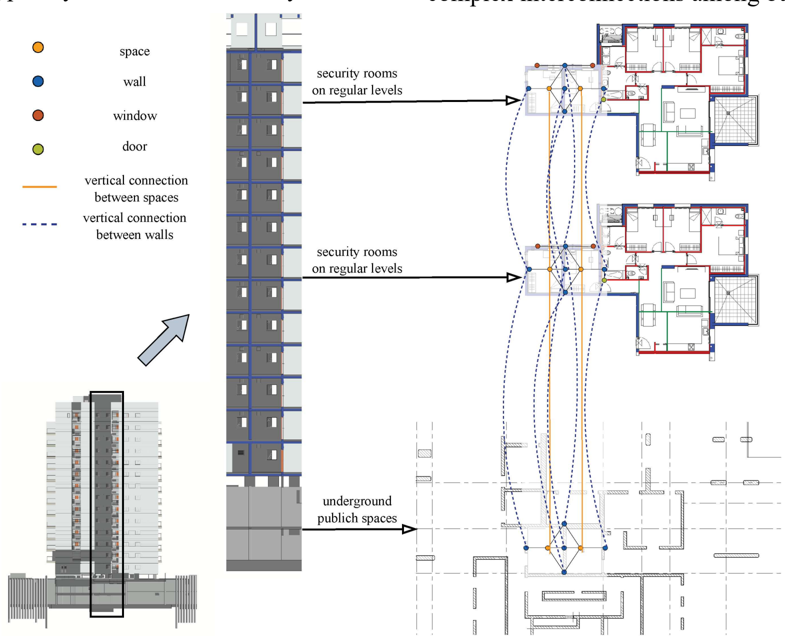

Figure 3: Graph representation of security shafts - extracting the vertical graphs.

To enable the application of GNNs, two steps are implemented. First, we aggregate all security rooms within a single shaft and link them to any continuous walls that support the security rooms. Given the high complexity of the interconnections among security-room-related elements, we interpret the design model to generate a shaft-centered graph structure utilizing the authoring tool's API via Dynamo (Autodesk, 2022). The topological query analyzes all shaft-related building elements across all floors.

By taking the bounding box coordinates of all shafts as horizontal references, we identify all security spaces that align horizontally with the shafts on all floors. We then filter out the irrelevant spaces and perform a bounding box geometric analysis at the space level to calculate the connecting relationships in the vertical direction. To achieve this, we vertically extend the space bounding boxes with a pad dimension threshold, which equals to the identified slab thickness. We also analyze the bounding relationships, including the boundary interactions between spaces and walls and the hosting relationships between walls and openings like windows and doors. This analysis translates the topological relationships of building elements and their hosts into the graph structure. Similarly, we apply the bounding box analysis to the selected walls to provide edge information about whether the structural walls of security rooms are continuously stacked until the foundations are reached. The abovementioned topological queries and geometric calculations are accomplished in an automated manner by taking the shaft-related building elements extracted from the design authoring tool as a basis.

The generated shaft-centered graph structure includes nodes that correspond to the individual components from the building models - walls, spaces, doors, and windows (Figure 3). Each node contained continuous geometric features like minimum and maximum x, y, z coordinates, area, length, height, and width, as well as intrinsic categorical features indicating the specific component type. The graph edges represent connection relationships between components based on the BIM layouts. Within the security shaft, each space is connected to the spaces above and below. Moreover, these spaces link with the associated walls, while the doors and windows are connected to the host wall. The walls of each space are linked with the aligned walls above and below that belong to security rooms on a level above and level below.

and the resulting graph representation was used to train a GNN model for node classification to categorize shaft components by vertical connection length ratio. To train the model, a set of 710 BIM models of shafts representing security rooms and other components that may be found in these shafts were generated and labeled manually using the Revit platform. To introduce realistic design variations into the models, the researchers received guidance from one of Israel's building control companies, in addition to reviewing several building control reports. One example of such a design variation is interconnected security shafts (where two security rooms share a wall) versus shafts comprising individual security rooms. Overall, the data set consists of 51,630 nodes and 78,811 edges, capturing the connections and relationships between the various components in the shafts.

The use of synthetic data comes in response to the great challenge of collecting relevant data, as industry stakeholders are usually reluctant to share BIM models. Hence, we follow the workflow suggested by (Bloch et al., 2023), where fully synthetic data was used for training a GNN model for ACC. It could be shown that the trained model had similar accuracy rates when applied for making predictions for real design.

In this case, the labels correspond to the three distinct classes characterized by the proportions of vertical connection ratios, as defined in the regulations. Namely, 70%, 50-70% range, and less than 50%. As illustrated in Figure 4, there are four types of elements represented as nodes in the graphs: spaces, walls, doors, and openings. Hence, we introduce an additional class that is related to "not applicable" (NA) nodes representing windows and doors.

The graph representations of these models were obtained using an automated workflow based on data extracted from the models to a spreadsheet. Since the synthetic data includes only individual security room shafts, the graph extraction process was different, but the resulting graph representation structure is identical to the structure obtained by the process depicted in Figure 4.

Using the described synthetic data set, a two-layer Graph Attention Network (GAT) model (Veličković et al., 2017) was trained to perform a node classification task. The training set was split into 70% for training, 15% for validation and 15% for testing. The testing phase provided a comprehensive assessment of the model's capabilities, even though the test phase was performed using a portion of the synthetic data set. The testing accuracy achieved 94.58% reflecting the model's proficiency in correctly classifying nodes within the graph. Precision, recall, and F1 score further validate the model's effectiveness, with values exceeding 94%. The described results are the outcome of iteratively evaluating multiple configurations of GNN models (such as GCN (Kipf and Welling, 2016) and SAGE (W. Hamilton et al., 2017)), model architectures and graph structures. Within the variations in graph structure, we evaluated graphs with different building elements to node mappings and with different feature vectors assigned to each node.

Table 1 illustrates the confusion matrix providing the results of predictions vs. actual labels of the test set. It provides insights into the model's classification behavior, with high counts along the diagonal indicating accurate predictions with a notable emphasis on correctly identifying all the 2025 instances in the "not applicable" class (class 4). For class 1 (>70%), the total count of nodes was 3,778, among these 3,650 instances were correctly predicted. However, 62 instances were misclassified as class 2 (50-70%) and additional 66 instances were misclassified as class 3 (<50%).

Table 1 Confusion matrix for synthetic test set

		Actual			
		>70%	**50-70%**	**<50%**	**NA**
Predicted	**>70%**	3650	62	66	0
	50-70%	196	449	19	0
	<50%	74	3	1201	0
	NA	0	0	0	2025

Class 2 (50-70%) showed the lowest accuracy, with 664 instances, out of which 449 instances correctly predicted, but 196 misclassified as class 1, and 19 misclassified as class 3. These misclassifications can be attributed to the imbalanced data set and insufficient number of examples covering these cases. Class 3 (<50%) predictions reached a better classification accuracy in comparison to class 2.

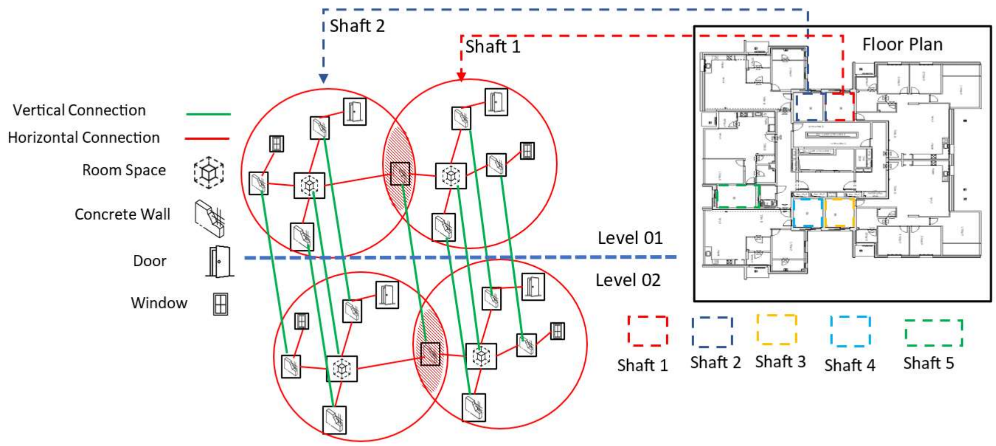

Figure 4: Graph representation of security shafts.

The total count of true class was 1278, out of which 1201 instances were correctly predicted. However, 74 instances were misclassified as Class 1, and an additional three instances were misclassified as Class 2. This also indicates the difficulty in identifying class 2 as it has high similarity to both class 1 and class 3. A better representation of this class can improve the results. In case the model is still not accurate enough in identifying class 2, we might think of a slightly different workflow that relies on distinguishing between the common case (>70% continuity) and the other cases.

After testing the model using a portion of the synthetic data, it was utilized to generate classifications on a real test case model. The real model is a residential building comprising 11 floors. Each floor accommodates five apartments, while the ground and first floors contain extra common and technical service areas. Additionally, the building includes two underground levels dedicated to parking, storage, and technical rooms. Each apartment in the building features a dedicated security room.

Notably, the model has five security shafts, where the first and second shafts are interconnected, as are the third and fourth shafts, via a shared separating wall (see Figure 4). The fifth shaft stands independently. Applying the classifier to the described real case, the model predictions achieved an accuracy of 86.84%. The confusion matrix for predictions is described in Table 2.

Table 2 Confusion matrix for the real case

		Actual			
		>70%	50-70%	<50%	NA
Predicted	>70%	208	6	6	0
	50-70%	9	21	9	0
	<50%	23	2	32	0
	NA	0	0	0	102

The complexity of the chosen test case is a result of aiming to maximize the available parking spaces, making the configuration of the walls on the lower levels of the building is not a standard case. While most of the elements follow the preferred guidelines of at least 70% vertical connection ratio, we can see representation of the other cases as well. It's noteworthy that the real-world instances include unique configurations that are not encountered and were not present in the training data.

Results of classification were injected back into the BIM model to be used for further checking requirements for reinforced concrete elements' thickness. Based on the predicted class for the spaces, a set of logical rules was inactivated to check the model for compliance with requirements for reinforced concrete element thickness. The overall checking process thus becomes "hybrid" by relying partially on graph-based and rule-based techniques. The supporting wall ratio for each space determines what requirements are applicable for the walls and slabs in the safety areas. Obviously, a misclassification of the elements by the GNN can lead to incorrect checking results. In this case, 90% of the security rooms in the model (nodes representing spaces) were classified correctly by the GNN model. The overall checking results based on this classification showed that out of a total of 265 walls and slabs checked, only 27 elements were incorrectly identified as "not compliant" while they truly are compliant with the code. Namely, 27 false positive cases. Zero false negative cases were identified. In the context of ACC, this distinction is very important since the non-compliant cases will have to be reevaluated by the designer\checker, but the false negatives may be approved despite not adhering to the code. ML models are not deterministic and may provide unreliable results. This limitation can be mitigated by further development of data sets, refinement of the models, and by constant validation against real cases. Additionally, by incorporating human oversight into the process, a more robust and reliable compliance-checking framework can be established.

Discussion and Conclusions

The extensively researched ACC domain may benefit from a significant shift in the overall approach. Hard-coded rules, being explicitly predefined and manually encoded, are often too restrictive. Furthermore, there is a tendency to confine the entire process to one single method, predominantly relying on rule-based systems. As a result, we miss opportunities for leveraging mutually beneficial methods that may lead to higher levels of automation in the overall process. Such approaches may incorporate learning techniques in addition to the human coded predetermined rules. The flexibility provided by the learning techniques is crucial in the field of ACC, where building regulations can be ambiguous or complex. One of the main limitations of such models is that they are not deterministic therefore, may be unreliable in the process of ACC. However, with large enough data sets and continuous exposure to diverse examples, these models can enhance their performance, making them well-suited for handling tasks from the code compliance checking domain. This work introduces a novel approach for ACC, leveraging semantic enrichment to enhance the level of automation that can be achieved. The approach is based on the capacity to decompose regulatory clauses into distinct semantic enrichment tasks, each addressed by a suitable solution. We demonstrate this approach with a test case, addressing specific requirements of Israeli regulations for security rooms. As described in this paper, most of the methods implemented in this case are based on graph theory, including community detection algorithms, and Graph Neural Networks. The introduced workflow is highly automated, with very little information requirements for the designers, showcasing the ability to reach a higher level of automation in the process by implementing diverse methods for every code clause. Although the training process includes manual work, it is not part of the checking as the model can be pretrained. The checking process itself relies on minimal data requirements and minimal manual BIM preprocessing, mainly making sure that the spaces have been created and that the security rooms are properly tagged.

Limitation

Several limitations are acknowledged in this study. Firstly, the transformation of the accessibility and connectivity relationships into graphs requires a well-defined BIM model, where space elements are comprehensively placed and correctly separated in the building layouts. While we aim not to add information requirements or specifications, good quality modeling practice is expected. The utilized building graphs, which are only parts of the various graph structures, are selected according to the topological features of the envisaged regulatory requirement. Thus, the applicability of the proposed graph-based enrichment for code checking on other regulations is to be justified.

Second, while we demonstrate the potential of GNNs for ACC, we acknowledge the limitation of the developed classifier, which is tailored to the context of the Israeli regulations for security rooms. Implementing a similar technique for different regulations would necessitate the collection or creation of new, context-specific synthetic datasets for training the models. Furthermore, the automated routines for data extraction must be modified to generate graphs with needed structures. Another limitation of using a synthetic data set for training is that real-world data may follow assumptions that are not well represented in the synthetic set. While we tried to limit this problem by working under the guidance of a building control company, our access to the design documents was limited. This underscores the importance of future efforts to address the scalability of the approach to accommodate a broader spectrum of regulations, including generating graph datasets tailored to support various requirements. The demonstrated promising model performance across various evaluation metrics is encouraging to continue developing this research direction.

Contribution

This work contributes to the ACC domain, addressing several existing challenges and paving the way for a more automated and adaptable approach to code compliance checking. The proposed approach for code checking, which leverages graph representation, introduces a novel way of thinking about ACC processes, where we do not have to be confined solely to rules. This approach facilitates a more comprehensive investigation of design knowledge. It has the capacity to elucidate not only the instances of non-compliance but also to subsequently link other elements that may be interconnected with these issues. Such enriched results empower designers with detailed insights, enabling more effective resolution strategies (Wu et al., 2023). This approach lays the groundwork for the advanced automated design adaptation method known as “Design Healing.” By providing a more nuanced understanding of the design and its compliance with building codes, designers can leverage this foundational data to make informed adjustments, enhancing the overall efficacy and compliance of the building model.

This work lays the groundwork for future research on automated code checking, encouraging the exploration of hybrid models that harness the strengths of diverse computational techniques for enhanced regulatory compliance. The proposed workflow is fully automated, with minimal information requirements to be supplemented by users. The success of the approach underscores the potential of semantic enrichment to enhance the level of automation achieved in ACC.

Acknowledgments

The research was funded by the Israeli Ministry of Construction and Housing. The presented research also received funding from the TUM International Graduate School of Science and Engineering (IGSSE).

References

Abualdenien, J., Borrmann, A., 2021. PBG: A parametric building graph capturing and transferring detailing patterns of building models, in: Proc. of the CIB W78 Conference 2021.

Autodesk, 2022. Dynamo.

Belsky, M., Sacks, R., Brilakis, I., 2016. Semantic enrichment for building information modeling. Comput.-Aided Civ. Infrastruct. Eng. 31, 261–274. https://doi.org/10.1111/mice.12128

Bloch, T., Borrmann, A., Pauwels, P., 2023. Graph-based learning for automated code checking – Exploring the application of graph neural networks for design review. Adv. Eng. Inform. 58, 102137. https://doi.org/10.1016/j.aei.2023.102137

Bloch, T., Sacks, R., 2020. Clustering Information Types for Semantic Enrichment of Building Information Models to Support Automated Code Compliance Checking. J. Comput. Civ. Eng. 34, 04020040.

Bloch, T., Sacks, R., 2018. Comparing machine learning and rule-based inferencing for semantic enrichment of BIM models. Autom. Constr. 91, 256–272. https://doi.org/10.1016/j.autcon.2018.03.018

Borrmann, A., Van Treeck, C., Rank, E., 2006. Towards a 3D spatial query language for building information models, in: Proc. Joint Int. Conf. of Computing and Decision Making in Civil and Building Engineering (ICCCBE-XI).

Fortunato, S., 2010. Community detection in graphs. Phys. Rep. 486, 75–174. https://doi.org/10.1016/j.physrep.2009.11.002

Girvan, M., Newman, M.E.J., 2002. Community structure in social and biological networks. Proc. Natl. Acad. Sci. 99, 7821–7826. https://doi.org/10.1073/pnas.122653799

Hamilton, W., Ying, Z., Leskovec, J., 2017. Inductive representation learning on large graphs. Adv. Neural Inf. Process. Syst. 30.

Hamilton, W.L., Ying, R., Leskovec, J., 2017. Representation Learning on Graphs: Methods and Applications. CoRR abs/1709.05584.

Home Front Command, 2010. Specifications for Building Shelters. Protective structures department, Home Front Command, Ramle, Israel.

Hu, Y., Xia, C., Chen, J., Gao, X., 2023. Clash context representation and change component prediction based on graph convolutional network in MEP disciplines. Adv. Eng. Inform. 55, 101896. https://doi.org/10.1016/j.aei.2023.101896

Kipf, T.N., Welling, M., 2016. Semi-supervised classification with graph convolutional networks. ArXiv Prepr. ArXiv160902907.

Mazairac, W., Beetz, J., 2013. BIMQL An open query language for building information models. Adv. Eng. Inform. 27, 444–456. https://doi.org/10.1016/j.aei.2013.06.001

Nawari, 2019. A Generalized Adaptive Framework (GAF) for Automating Code Compliance Checking. Buildings 9, 86. https://doi.org/10.3390/buildings9040086

Solihin, W., Eastman, C., 2015. Classification of rules for automated BIM rule checking development. Autom. Constr. 53, 69–82. https://doi.org/10.1016/j.autcon.2015.03.003

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y., 2017. Graph attention networks. ArXiv Prepr. ArXiv171010903.

Vilgertshofer, S., Borrmann, A., 2017. Using graph rewriting methods for the semi-automatic generation of parametric infrastructure models. Adv. Eng. Inform. 33, 502–515. https://doi.org/10.1016/j.aei.2017.07.003

Wang, Z., Sacks, R., Yeung, T., 2022. Exploring graph neural networks for semantic enrichment: Room type classification. Autom. Constr. 134, 104039. https://doi.org/10.1016/j.autcon.2021.104039

Wu, J., Dubey, R.K., Abualdenien, J., Borrmann, A., 2023. Model Healing: Toward a framework for building designs to achieve code compliance, in: ECPPM 2022-eWork and eBusiness in Architecture, Engineering and Construction 2022. CRC Press, pp. 450–457.

Wu, J., Zhang, J., 2019. New automated BIM object classification method to support BIM interoperability. J. Comput. Civ. Eng. 33, 04019033.

Wülfing, A., Windisch, R., Scherer, R., 2014. A visual BIM query language, in: eWork and eBusiness in Architecture, Engineering and Construction: ECPPM 2014. p. 157.

Zhang, J., El-Gohary, N.M., 2017. Integrating semantic NLP and logic reasoning into a unified system for fully-automated code checking. Autom. Constr. 73, 45–57. https://doi.org/10.1016/j.autcon.2016.08.027