

Article

Automated Building Information Modeling Compliance Check through a Large Language Model Combined with Deep Learning and Ontology

Nanjiang Chen ¹ , Xuhui Lin ², Hai Jiang ¹ and Yi An ^{3,*}

¹ Department of Industrial Engineering, Tsinghua University, Beijing 100084, China; cnj22@mails.tsinghua.edu.cn (N.C.); haijiang@tsinghua.edu.cn (H.J.)

² The Bartlett School of Sustainable Construction, University College London, London WC1E 6BT, UK; zczlx13@ucl.ac.uk

³ Department of Engineering, Cardiff University, Cardiff CF24 3AA, UK

* Correspondence: any4@cardiff.ac.uk

Abstract: Ensuring compliance with complex industry standards and regulations during the design and implementation phases of construction projects is a significant challenge in the building information modeling (BIM) domain. Traditional manual compliance checking methods are inefficient and error-prone, failing to meet modern engineering demands. Natural language processing (NLP) and deep learning methods have improved efficiency and accuracy in rule interpretation and compliance checking. However, these methods still require extensive manual feature engineering, large, annotated datasets, and significant computational resources. Large language models (LLMs) provide robust language understanding with minimal labeled data due to their pre-training and few-shot learning capabilities. However, their application in the AEC field is still limited by the need for fine-tuning for specific tasks, handling complex texts with nested clauses and conditional statements. This study introduces an innovative automated compliance checking framework that integrates LLM, deep learning models, and ontology knowledge models. The use of LLM is motivated by its few-shot learning capability, which significantly reduces the need for large, annotated datasets required by previous methods. Deep learning is employed to preliminarily classify regulatory texts, which further enhances the accuracy of structured information extraction by the LLM compared to directly feeding raw data into the LLM. This novel combination of deep learning and LLM significantly enhances the efficiency and accuracy of compliance checks by automating the processing of regulatory texts and reducing manual intervention. This approach is crucial for architects, engineers, project managers, and regulators, providing a scalable and adaptable solution for automated compliance in the construction industry with broad application prospects.

Keywords: automated compliance check; large language models (LLMs); deep learning; ontology knowledge models; BIM; design regulations



Citation: Chen, N.; Lin, X.; Jiang, H.; An, Y. Automated Building Information Modeling Compliance Check through a Large Language Model Combined with Deep Learning and Ontology. *Buildings* **2024**, *14*, 1983. <https://doi.org/10.3390/buildings14071983>

Academic Editor: Heap-Yih Chong

Received: 30 May 2024

Revised: 19 June 2024

Accepted: 22 June 2024

Published: 1 July 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Building information modeling (BIM) has become an essential tool in the domain of architecture, engineering and construction (AEC), revolutionizing the way projects are planned, executed, and maintained [1]. BIM enhances project delivery efficiency, optimizes resource allocation, and significantly improves the overall project value and performance [2]. However, as the use of BIM technology becomes more widespread, ensuring compliance with complex and evolving industry standards and regulations during the design and implementation phases of construction projects presents new challenges [3].

Traditional compliance checking methods, which rely on manual processes, are not only inefficient but also prone to errors [4,5]. This inefficiency and susceptibility to errors are contrary to the principles of lean and efficient management in engineering. As projects

grow larger and regulations become more complex, the limitations of traditional methods become more evident [6], creating an urgent need for more efficient and accurate methods to address these challenges.

In promoting compliance checking in the architecture, engineering, and construction (AEC) industry, many researchers worldwide have extensively studied automated rule checking (ARC) systems and have established several ARC systems in different countries and regions, such as Singapore's CORENET, Norway's HITOS project, Australia's Building Codes Board project, the International Code Council project, and the U.S. General Services Administration project. The rule-checking process can be roughly divided into four stages: (1) rule extraction—translating rules expressed in natural language into a computer-processable format; (2) building model preparation—preparing the necessary information for the checking process; (3) rule execution—using computer-processable rules to check the prepared model; and (4) reporting the checking results [4]. However, most of the mentioned ARC systems are based on manual rule interpretation methods, which are inefficient in maintaining and modifying hard-coded rules.

Therefore, researchers have started proposing semi-automated and automated methods to interpret regulatory texts into computer-processable formats, aiming for a more flexible, transparent, and convenient ARC process. For example, the RASE methodology [7] introduces a semi-automated rule interpretation method to help AEC experts analyze the semantic structure of regulatory requirements, using document annotation techniques to mark different components of the regulatory requirements. However, these methods require a significant amount of manual effort to annotate regulatory documents and create query statements or pseudo-code from them. To address this issue, researchers worldwide have started introducing natural language processing (NLP) techniques [8], a widely used method for processing and understanding human language-based text, to automate rule extraction from regulatory documents. Typically, automated rule interpretation methods based on NLP include two tasks: (1) information extraction—extracting semantic information from textual regulatory documents, and (2) information transformation—converting the extracted information into logical clauses to support reasoning in compliance checking. Linking information extraction and information transformation tasks greatly facilitates the automated rule interpretation process. However, whether using statistical methods, word vectors, or pre-trained language models like BERT, a large amount of manually annotated training data is still needed for specific downstream tasks, and achieving high accuracy requires significant technical investment, such as adjusting model architectures and hyperparameters.

With the emergence of LLMs, such as OpenAI's GPT series [9], Meta's Llama models [10], and Claude, the cost of technical investment has been significantly reduced. These models, through extensive pre-training on large-scale, diverse corpora, demonstrate strong natural language understanding and generation capabilities. In practical applications, these models can adapt to specific downstream tasks with minimal instruction learning or fine-tuning, resulting in an "emergent capability" phenomenon. Emergent capability refers to the model's ability to handle unseen data and exhibit exceptional performance, surpassing the limitations of traditional models. This performance improvement not only reduces development and maintenance costs but also expands the application range of the models, which show great potential and advantages in tasks such as extracting structured information from regulatory texts in automated BIM compliance checking. However, the current application of LLMs in automated compliance checking (ACC) still faces challenges, particularly in handling complex, nested, and conditional statements in regulatory texts, where they exhibit certain limitations.

Based on this, this study proposed an innovative framework for automated BIM compliance checking, combining LLMs, deep learning models, and ontology knowledge models to address the challenges and limitations of traditional compliance checking. The innovation of this framework lies in the integration of LLMs and deep learning models. The use of LLMs was motivated by their few-shot learning capability, which significantly

mitigates the limitation of requiring large, annotated datasets that was prevalent in previous methods. Previous studies have typically relied on either rule-based or machine learning methods that require extensive manual intervention and large amounts of labeled data. Deep learning models were employed to preliminarily classify regulatory texts. This pre-classification step, followed by feeding both the classified results and the regulatory texts into the LLM, enhanced the accuracy of structured information extraction by the LLM compared to directly feeding raw regulatory texts into it. Additionally, domain ontology knowledge models provide strong support for understanding and processing professional contexts and concepts in the construction field. By combining LLMs, deep learning models, and domain ontology knowledge models, the innovative framework can not only accurately understand and interpret professional and complex construction regulatory texts but also ensure that the extracted rules and information can be accurately applied to specific construction projects and practices.

The remaining parts of this paper are organized as follows: Section 2 provides a literature review, outlining the historical progress in the field of compliance checking in the construction industry and emphasizing the current research gaps. Section 3 constructs the overarching framework for automated compliance checking based on the identified research gaps. Section 4 validates the proposed framework through practical case studies. Section 5 discusses the overall experimental results of the automated compliance checking, its limitations, and future development directions. Section 6 concludes the entire study.

2. Literature Review

The current research landscape in automated compliance checking has explored various approaches. Early methods primarily relied on hardcoded rules and manual interpretation. For instance, a 2009 survey by Eastman et al. reviewed several rule-checking projects, including Singapore's CORENT, Norway's Statsbygg, and the US General Services Administration (GSA) initiatives [4]. These methods involved experts manually extracting and coding rules from legal texts. While these methods may be successful in their implementations, they have several drawbacks: they are expensive to maintain, difficult to modify, and lack a generalized framework for rules and regulations modeling. These approaches are often referred to as 'Black Box' or 'Gray Box' methods [11].

To address these limitations, semi-automated methods like the RASE methodology were introduced, improving efficiency by translating regulatory texts into machine-processable formats using logical operators [7]. The RASE method employs four logical operators—requirement, applicability, selection, and exception—to convert textual regulations into computable rules, significantly enhancing the accuracy of rule interpretation. However, these methods still require significant manual effort.

The introduction of natural language processing (NLP) techniques in compliance checking marked a significant advancement [12]. Early NLP algorithms for compliance checking can be broadly categorized into two methods: rule-based methods and statistical methods [8]. While rule-based methods often perform better in terms of accuracy and recall, they require more human labor. Zhang and El-Gohary pointed out that domain-specific regulatory texts are more suitable for automated NLP compared to general non-technical texts (e.g., news articles, general websites) [13].

In 2011, Zhang and El-Gohary proposed a semantic and syntactic information extraction method for compliance checking, aimed at automatically extracting structured information from unstructured texts [14]. Additionally, El-Gohary et al. introduced the concept of ontology, which is used to represent domain knowledge [15]. In 2014, Zhou and El-Gohary introduced machine learning-based text classification algorithms as a preliminary step before information extraction to categorize text in regulatory documents into predefined categories, thus improving the efficiency of subsequent semantic information extraction and compliance reasoning [16]. In 2015, Zhang and El-Gohary proposed an innovative method for automatically extracting rules from building regulations and converting them into logical clauses suitable for automated reasoning [17]. This method combined

rule-based semantic NLP techniques with a set of semantic mapping and conflict resolution rules to automate the conversion process. In 2017, Zhou and El-Gohary proposed an ontology-based information extraction method to support fully automated building energy compliance checks [18]. This method involved classifying and preprocessing complex text content, then using ontology pattern matching, sequential dependency extraction, and cascading extraction techniques to handle the text's complexity.

Despite significant progress in using NLP for automated compliance checking, especially in information extraction and rule conversion, these technologies still face challenges in scalability and adaptability. Traditional NLP methods, while effective in specific contexts, often require extensive manual feature engineering, necessitating significant expert time and effort to define and adjust features. Additionally, manually developed rules may need to be reevaluated and adjusted when applied to different types of building regulations or when regulations are updated.

With the continuous development of deep learning, new possibilities have emerged for interpreting regulatory texts and ensuring compliance. Deep learning models automatically extract and learn features through multi-layer neural networks, eliminating the need for manually defined features. This allows models to identify useful features in regulatory texts without human intervention, thus reducing the burden of manual feature engineering. Additionally, deep learning models typically learn in an end-to-end manner, from raw input (text) directly to output (machine-processable form). This eliminates the need for multiple processing stages in traditional machine learning, such as feature extraction and feature selection.

In 2019, Zhang and El-Gohary proposed a new machine learning-based approach, which uses recurrent neural networks to extract hierarchical information from building regulations [19]. In 2021, Zhang and El-Gohary highlighted that deep learning models, compared to traditional machine learning models, have more parameters and typically require larger datasets for training. However, the AEC field lacks sufficient annotated training datasets, and creating these datasets is costly. To address this issue, they proposed using transfer learning strategies, allowing deep neural network models to be trained on both general domain and AEC-specific annotated data [20]. Zheng et al. proposed a fire compliance checking framework for building BIM models based on a knowledge model and the BERT model [21]. This approach significantly improved the model performance and generalization ability. However, these models require substantial computational resources and data, and their internal mechanisms' complexity makes interpretability a significant issue.

However, LLMs, such as GPT-4 [9], PaLM [22], Galactica [23] and Llama [10], have shown exceptional capabilities in few-shot learning [24] and understanding complex language structures [25], which offers promising solutions for the above-mentioned gaps. LLMs can address the limitations of deep learning by providing robust language understanding with minimal labeled data, adapting to evolving regulations, and accurately extracting structured information from regulatory texts.

Recent advancements in automated compliance checking have introduced LLMs to enhance automation. Liu et al. (2023) presented a method for automated compliance checking of building design regulations through prompt engineering, leveraging GPT-3 and GPT-3.5 models [26]. That study evaluated these models' performance in processing building design specifications through a series of experiments. Their research demonstrated the potential of LLMs in handling and understanding complex text tasks, particularly in dealing with large-scale, complex text contexts. By designing various types of prompts (such as zero-shot learning, one-shot learning, and few-shot learning) and utilizing fine-tuning processes, their study significantly improved model performance. However, their study primarily focused on relatively simple regulatory texts. More complex texts with nested clauses and conditional statements may require additional strategies for effective processing.

To sum up, the key studies related to automated compliance checking is provided in Table 1. Despite the progress made with NLP and deep learning technologies in automated compliance checking, several key challenges and research gaps remain:

- **Traditional Methods:** Traditional methods are expensive to maintain, difficult to modify, and lack a generalized framework for rules and regulations modeling.
- **NLP and Deep Learning:** These methods have improved efficiency and accuracy in rule interpretation and compliance checking. However, NLP and deep learning methods still require extensive manual feature engineering, large, annotated datasets, and significant computational resources. Additionally, compared to other industries, the AEC field has relatively less available data, making it challenging for deep learning models to learn sufficiently.
- **Large Language Models:** LLMs provide robust language understanding with minimal labeled data due to their pre-training and few-shot learning capabilities. They adapt to evolving regulations and accurately extract structured information from regulatory texts. However, their application in the AEC field is still limited by the need for fine-tuning for specific tasks and for handling complex texts with nested clauses and conditional statements.

Table 1. Key studies related to automated compliance check.

Study	Method	Advantages	Disadvantages	Key Contributions
Eastman et al. (2009) [4]	Manual rule extraction and coding		High maintenance cost, difficult to modify, lack of a generalized framework	Reviewed several rule-checking projects
RASE Methodology (2011) [7]	Semi-automated rule interpretation	Improved efficiency	Requires significant manual effort	Translates regulatory texts using logical operators
Zhang and El-Gohary (2011) [14]	Semantic and syntactic information extraction	Automatically extracts structured information	Requires large, annotated datasets	Proposed an automated compliance checking method
Zhou and El-Gohary (2014) [16]	Machine learning text classification	Improved efficiency of semantic information extraction	Requires large datasets and feature engineering	Improved text classification using machine learning
Zhang and El-Gohary (2015) [17]	Rule extraction and logical conversion	Automates the conversion process	Requires complex conflict resolution rules	Innovatively extracts building regulation rules
Zhou and El-Gohary (2017) [18]	Ontology-based information extraction	Supports fully automated energy compliance checks	Difficult to handle text complexity	Uses ontology pattern matching and cascading extraction
Zhang and El-Gohary (2019) [19]	Recurrent neural networks	Automatically extracts hierarchical information	Requires large datasets and computational resources	Uses deep learning to extract building regulation information
Zhang and El-Gohary (2021) [20]	Transfer learning	Reduces data requirements	Requires high computational resources	Combines domain-specific and general domain data
Zheng et al. (2021) [21]	BERT model	Improved model performance and generalization ability	Lack of interpretability	Proposed a fire compliance checking framework for BIM models
Liu et al. (2023) [26]	Prompt engineering with GPT models	Reduces need for extensive manual feature engineering, adapts to evolving regulations, accurately extracts structured information from regulatory texts	Primarily focuses on relatively simple regulatory texts. More complex texts with nested clauses and conditional statements may require additional strategies for effective processing	Presented a method for automated compliance checking using GPT models

3. Materials and Methods

To address the limitations of previous research and harness the strengths of LLMs, this study proposed an innovative automated compliance checking framework. This framework integrated LLM, deep learning models, and ontology knowledge models to enhance the efficiency and accuracy of compliance checks. The framework comprised four

main components: the construction of a lightweight domain ontology model, preliminary classification of regulatory texts using deep learning, extraction of structured information from regulatory texts using LLM, and the implementation of compliance checks within the ontology model.

The novelty of this approach lies in the combination of LLM and deep learning models, which significantly reduces the need for manual intervention and enhances automation. The rationale for using this combined approach is as follows: First, the ontology model provides standardized knowledge representation for complex objects in the construction industry, facilitating data sharing and reuse, while enhancing querying and reasoning capabilities, which are critical for automated compliance checking. Second, deep learning techniques are employed to preliminarily classify the regulatory texts, effectively handling large volumes of unstructured data and automatically extracting useful features. This pre-classification step improves the accuracy of structured information extraction by the LLM compared to directly feeding raw regulatory texts into it. Finally, LLMs, through extensive pre-training on large-scale and diverse corpora, acquire extensive language and common-sense knowledge, allowing them to perform few-shot learning. This capability significantly reduces the need for large, annotated datasets, adapts to evolving regulations, and accurately extracts structured information from regulatory texts. By integrating these technologies, the proposed framework not only enhances the efficiency and accuracy of compliance checking but also offers a scalable and adaptable solution for the construction industry.

The proposed automated compliance checking framework is illustrated in Figure 1, detailing each step from data input to compliance check result output, ensuring transparency and efficiency throughout the process. This methodology not only enhances the efficiency and accuracy of compliance checks but also provides a sustainable, adaptable automated solution for the construction industry with broad application prospects and practical value.

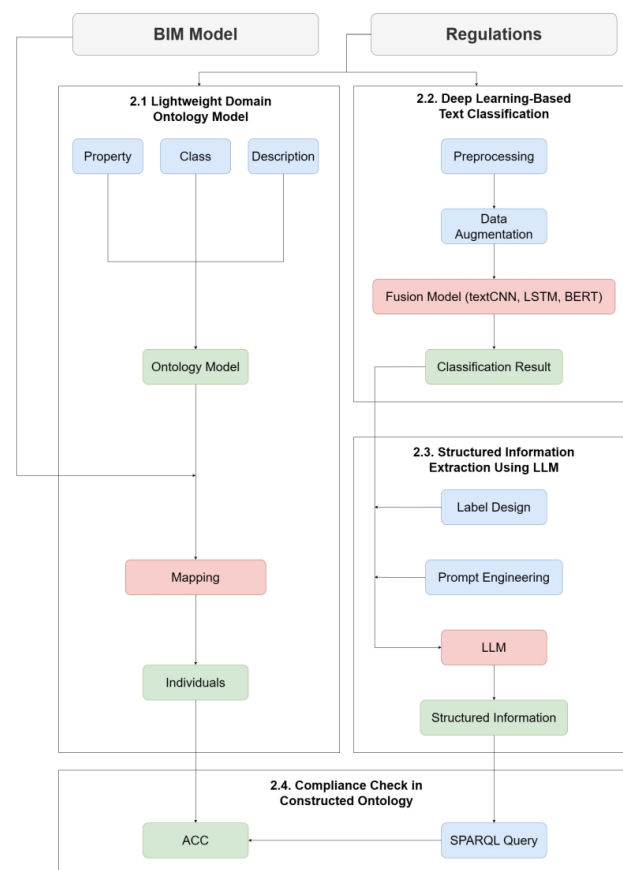


Figure 1. The overall framework of the proposed automated compliance check process.

3.1. Standards and Regulations

The compliance checking framework matches the BIM model with specific standards and regulations from the Residential Design Code GB-50096-2011 [27]. This code was issued by the Ministry of Housing and Urban–Rural Development and the General Administration of Quality Supervision, Inspection, and Quarantine of China. The standards include quantitative clauses such as area and height requirements and qualitative clauses like spatial functionality and building aesthetics.

3.2. Lightweight Domain Ontology Model

Ontology is defined as a formal representation of a set of concepts within a domain and the relationships between those concepts. In this study, an ontology model for residential building design was constructed using Protégé software (version 5.5.0, developed by the Stanford Center for Biomedical Informatics Research at the Stanford University School of Medicine). The concepts, relationships, and descriptions in this model were extracted and summarized from the Residential Design Code GB-50096-2011 [27].

Classes in the ontology model represent collections of instances. To build a lightweight ontology model, a semi-automated method proposed by Qiu et al. [28] was adopted. This method involves the following steps:

- Term Extraction: Using term frequency-inverse document frequency (TF-IDF) to extract relevant terms from regulatory texts;
- Semantic Clustering: Merging extracted terms using semantic clustering techniques;
- Manual Adjustment: Manually adjusting and supplementing the overall probability and hierarchical structure to ensure accuracy and relevance.

Attributes in the ontology model define binary relationships between instances, including data attributes and object attributes:

- Data Attributes: Connect instances with data values, typically basic data types such as strings, integers, floating-point numbers, or dates;
- Object Attributes: Define the relationships between two instances within the ontology model.

To improve accuracy and consistency, three types of metadata were used to describe the concepts in the building design domain:

- Concept Name: The primary name of the concept;
- Alias: Alternative names or synonyms for the concept;
- Definition: Detailed descriptions of the concept, taken from the “Terms” section of GB-50096-2011.

By constructing this lightweight domain ontology model, a standardized knowledge representation was provided that enhanced data sharing, reuse, querying, and reasoning capabilities, which are critical for automated compliance checking in the construction industry.

3.3. Deep Learning-Based Text Classification

One of the core challenges in automated compliance checking for building information modeling (BIM) is accurately extracting key information from complex regulatory texts. To enhance the precision and efficiency of information extraction by the LLM, this study incorporated deep learning models for preliminary text classification.

3.3.1. Text Preprocessing

In the field of natural language processing (NLP), text preprocessing is a crucial step that involves preparing and cleaning raw text by removing noise, redundancy, irrelevant, and harmful information [29]. When dealing with regulatory texts in building design, text preprocessing is particularly important because these texts often contain complex structures, technical terms, and formatting norms. Through preprocessing, irrelevant information can be removed and terms can be standardized, enabling subsequent models to more effectively identify and extract the key information. This not only helps improve

the efficiency of subsequent models but also ensures the accuracy and reliability of the information extraction. The text preprocessing steps adopted in this study are as follows:

- Noise Removal: Eliminating non-essential parts such as headers, footers, page numbers, and dates;
- Sentence and Paragraph Segmentation: Breaking down the text into smaller units like sentences or paragraphs to facilitate easier processing;
- Term Standardization: Normalizing professional terms, abbreviations, and non-standard expressions to reduce ambiguity and improve model accuracy.

By implementing these detailed preprocessing steps, the regulatory texts are transformed into a clean, structured, and standardized format suitable for deep learning models. This ensures that the models can effectively identify and extract key information, improving the accuracy and efficiency of the text classification task.

3.3.2. Training Data Preparation

A wide range of regulatory texts from building codes were collected and annotated. The annotation process involved classifying the text based on the complexity of the information, resulting in different layers of information:

- Single-layer Information: Simple regulations, e.g., “The area of the living room should not be less than 10 square meters” (labeled as $y = 0$);
- Double-layer Information: Regulations involving two conditions or entities, e.g., “The apartment with at least one balcony should have a minimum balcony area of 8 square meters” (labeled as $y = 1$);
- Triple-layer Information: More complex regulations involving multiple conditions or entities, e.g., “The apartment consisting of a bedroom, living room, dining room, and balcony should have a minimum living room illumination standard of 250 lux” (labeled as $y = 2$).

To enhance the model’s generalization capability and avoid overfitting, data augmentation techniques such as synonym replacement and sentence reorganization were employed. This approach increases the diversity of the training data, allowing the model to learn from a broader set of examples.

3.3.3. Text Classification

For accurate text classification, three commonly used models were compared (TextCNN [30], LSTM [31], and BERT [32]) and a hybrid model that combined their strengths was proposed:

- TextCNN Model: The TextCNN model was constructed with a 100-dimensional word embedding layer, followed by three one-dimensional convolutional layers with kernel sizes of 3, 4, and 5, each having 150 feature maps. These layers capture local features through sliding windows, which are then max-pooled and fed into a fully connected layer;
- LSTM Model: The LSTM model used a 100-dimensional word embedding layer, followed by an LSTM layer with an output dimension of 50. The LSTM layer processes the sequential data, and its output is flattened and fed into a fully connected layer;
- BERT Model: The BERT-base-Chinese model was used as the base, tokenizing the text using the AutoTokenizer and fine-tuning the model with the collected dataset. BERT’s architecture includes multiple transformer layers that capture deep semantic relationships within the text;
- Hybrid Model: The hybrid model integrated BERT for deep semantic feature extraction, CNN for local feature capture, and LSTM for sequence dependency. The outputs of these models were combined using a transformer encoder layer, followed by fully connected layers for the final classification.

By combining these models, their individual strengths were leveraged to improve the accuracy and efficiency of text classification, which is critical for the subsequent structured information extraction.

3.4. Structured Information Extraction Using Large Language Model

In order to extract structured information from regulatory texts, GPT-4 from OpenAI was utilized. Based on the classification results from the deep learning model, “one-shot learning” approach was adopted, setting labels and providing examples to guide the LLM in extracting structured information.

3.4.1. Label Design

In the field of building design, regulatory texts typically contain two types of conditions: (1) Quantitative conditions, such as “The area of the living room should not be less than” and (2) Existence conditions, such as “Each apartment should have a balcony”. Regulatory texts usually comprise four parts:

- The object to be checked;
- The property of the object to be checked;
- The relationship between the first two, either a quantitative condition or an existence condition;
- The actual value for quantitative conditions or a T/F Boolean value for existence conditions.

Based on these characteristics, this study set four labels for regulatory texts: “Object”, “Property”, “Condition”, and “Value”. The definitions for each label are shown in Table 2.

Table 2. Label Definitions.

Name	Title 2
Object (object)	The object to be checked in the regulatory text
Property (prop)	The property of the object to be checked
Condition (cmp)	The condition to be met, either quantitative or existence
Value (value)	The actual value for quantitative conditions or T/F Boolean for existence conditions

In this study, the selected regulatory text clauses can be divided into three types based on the information hierarchy level according to the classification results from the previous sections:

- Regulations containing one layer of information: For instance, “The area of the living room should not be less than 10 square meters” belongs to this category, including one set of “Object-Property-Condition-Value” labels;
- Regulations containing two layers of information: For example, “The apartment consisting of a bedroom, living room, kitchen, and bathroom should have a minimum total area of 30 square meters” belongs to this category, including two sets of “Object-Property-Condition-Value” labels;
- Regulations containing three layers of information: These are more complex regulations involving multiple conditions or entities, resulting in three sets of “Object-Property-Condition-Value” labels.

As shown in Figure 2, these labeling categories help in structuring the regulatory texts effectively, ensuring accurate information extraction and compliance verification.

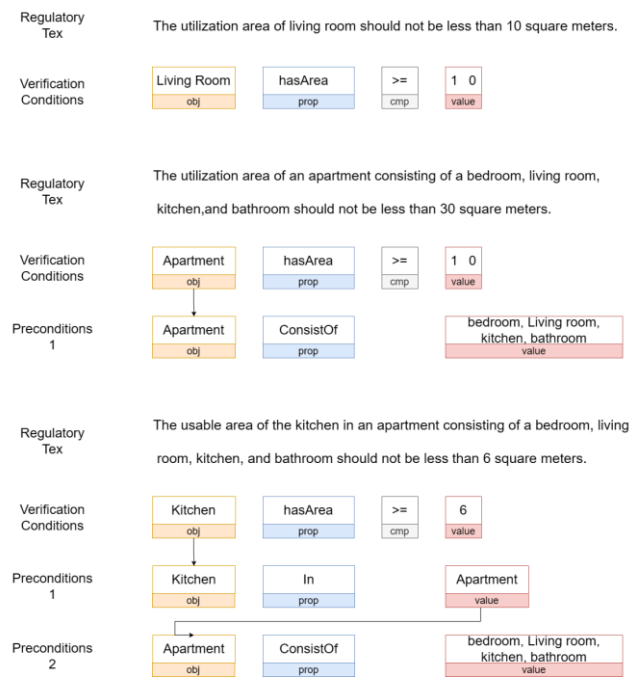


Figure 2. Label design.

3.4.2. One-Shot Learning with LLM

In this study, a “one-shot learning” approach was employed to fine-tune the LLM for structured information extraction. This method involves providing the model with a few annotated examples for each label category. By doing so, the LLM can learn to generalize from these examples and accurately extract structured information from new, unseen regulatory texts.

The process includes the following steps:

- **Annotation of Examples:** A small set of regulatory text examples are manually annotated with the designed labels;
- **Model Fine-Tuning:** The annotated examples are used to fine-tune the LLM, enabling it to recognize and extract the labeled components from similar texts;
- **Extraction and Structuring:** The fine-tuned LLM processes the regulatory texts, extracting the relevant components and organizing them into a structured format.

By using LLM for structured information extraction, the need for manual intervention is significantly reduced, enhancing the automation and efficiency of compliance checking. This approach ensures that regulatory texts are accurately processed and converted into actionable queries, facilitating effective compliance verification.

3.5. Compliance Check in Constructed Ontology

After extracting the structured information, the next step is to implement compliance checking within the ontology model using SPARQL queries. This involves a rule-based algorithm that translates the extracted components into a query format suitable for compliance verification.

The conversion process is rule-based and follows these steps:

- **Mapping Labels to Ontology Classes:** Each label (object, property, condition, value) is mapped to corresponding classes and properties in the ontology model. For example, the label “Object” might be mapped to a class such as living room, and the label “Property” might be mapped to a property such as area;
- **Formulating SPARQL Queries:** Using the mapped classes and properties, SPARQL queries are formulated to check compliance. The structured information extracted from the regulatory texts is used to construct these queries;

- Executing Queries: The formulated SPARQL queries are executed against the ontology model to verify whether the instantiated objects comply with the regulatory requirements.

4. Results

To validate the proposed automated compliance checking framework, a case study on representative construction projects was conducted. The case study involved applying the developed framework to real-world BIM models and analyzing the compliance results. A nine-story residential building was selected as the primary case study to demonstrate the effectiveness of the proposed approach.

4.1. BIM Model and Ontology Integration

Due to the confidentiality of building BIM models, a custom residential design model was constructed for this study to validate the proposed compliance checking framework. The model represents a nine-story residential building, with each floor containing four residential units and a public area. The overall model and the floor plan of one of the floors is depicted in Figure 3.

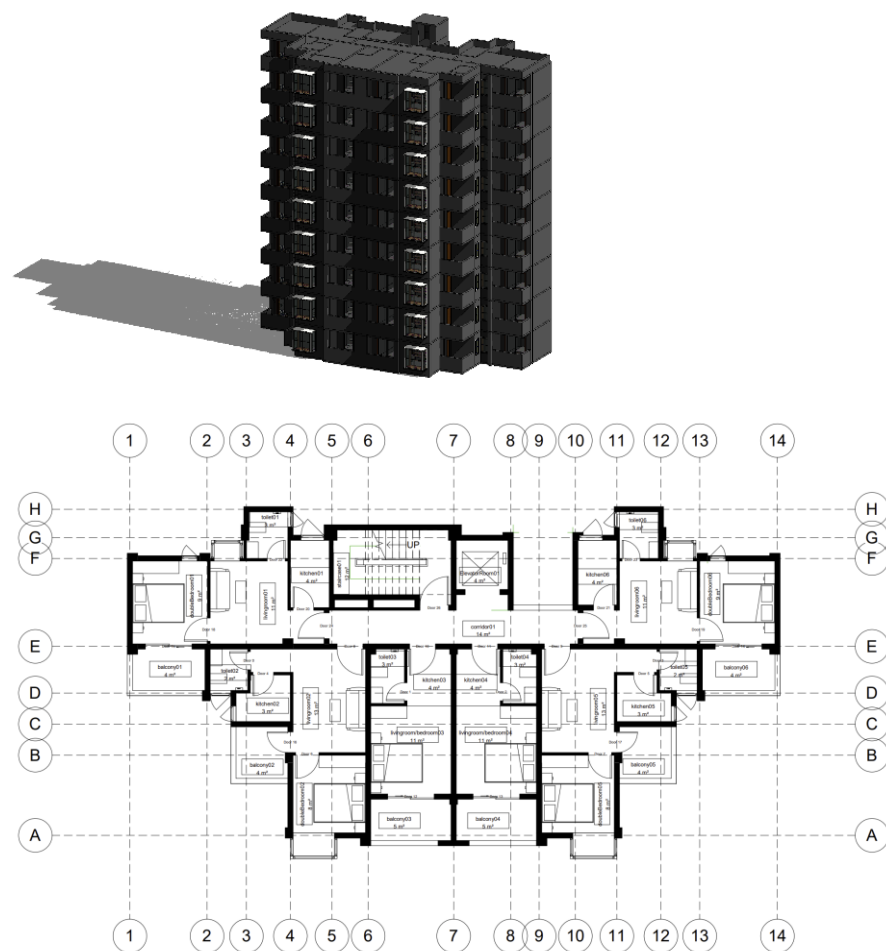


Figure 3. Overall Building and Floor Plan. Note: A~H in the figure represent the room names, and the numbers 1~14 represent the room numbers.

To map the information from the building BIM model into the constructed ontology model, it is necessary to convert the visual information into data. Industry foundation classes (IFC) is a common collaboration format in BIM projects, supported by most BIM software. Therefore, using IFC files as the base format for data exchange enhances scalability. However, since IFC uses the EXPRESS Schema, it cannot be directly parsed by ontology

reasoning engines. Additional steps are required to convert IFC data into the RDF format usable by the ontology model.

First, based on the constructed ontology model, concepts in the IFC schema were aligned with those in the ontology model using a predefined mapping table. This alignment and extraction process was conducted in Python, where the information was organized and stored in dictionary form. This preparation is essential for mapping the information into the ontology knowledge model. The key information extracted includes “Room”, “Room Number”, “Floor Number”, “Utilization Area”, and “Apartment Type”, as this study primarily focuses on area requirements in residential design.

Next, to match the building model information with regulatory information in the ontology knowledge model, the extracted data were mapped into the ontology knowledge model. This process involved writing the model information into RDF format using the .ttl language based on the designed classes, properties, and descriptions in the ontology knowledge model. The RDF format file was then mapped into the constructed ontology knowledge model, as shown in Figure 4. This step converted objects in the building model into instantiated objects in the ontology knowledge model, with corresponding “data properties” and “object properties”. A brief description of the structure of the constructed ontology model is illustrated in Figure 5.

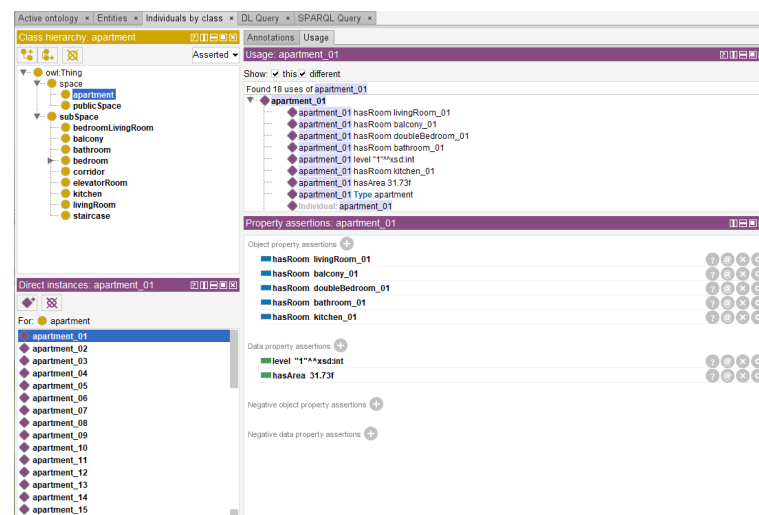


Figure 4. Individuals in ontology model.

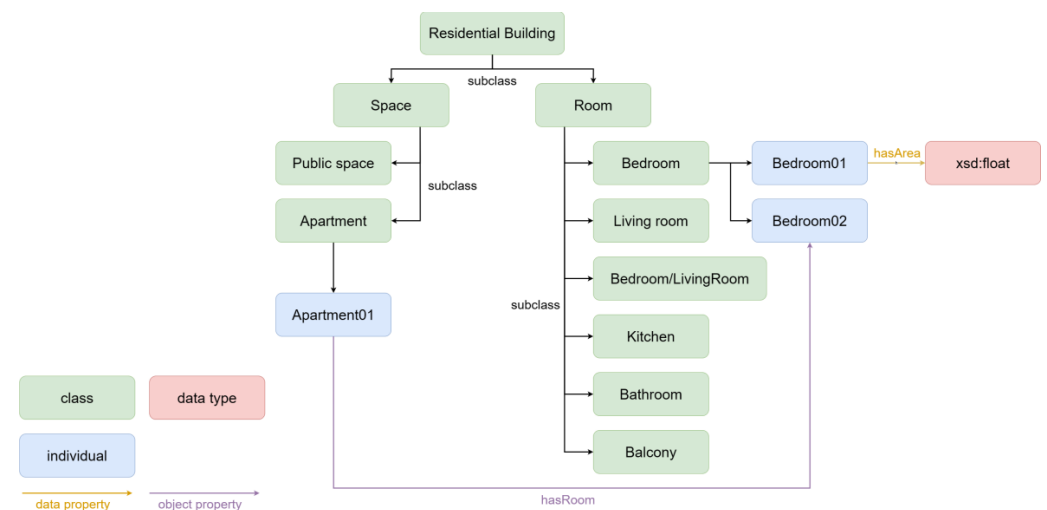


Figure 5. Ontology model structure.

By integrating the BIM model with the ontology model, the framework ensures that the visual and data information from the building model are accurately represented and can be used for effective compliance checking.

4.2. Text Classification

This section explores the performance of the Text Convolutional Neural Network (TextCNN), Long Short-Term Memory Network (LSTM), Transformer model, BERT model, and BERT-TextCNN-LSTM fusion model in the task of text classification. To objectively evaluate the performance of each model, statistical analysis methods including accuracy, loss value graphs, classification reports (such as *precision*, *recall*, *F1 – score*), and confusion matrices were used. The formulas for calculating *precision*, *recall*, and *F1 – score* are as follows:

$$precision = \frac{true\ positive}{true\ positive + false\ position}$$

$$recall = \frac{true\ positive}{true\ positive + false\ negative}$$

$$F1 - score = 2 \times \frac{precision \times recall}{precision + recall}$$

4.2.1. TextCNN

The TextCNN model constructed in this study uses standard architecture. It first converts the input text into fixed-dimensional vector representations through a 100-dimensional word embedding layer (nn.Embedding). Then, the model employs three one-dimensional convolutional layers (nn.Conv1d) with kernel sizes of three, four, and five, each having 150 feature maps. The convolution layers use a sliding window mechanism to extract local features from different regions of the text, enhancing the model's sensitivity to key semantics through activation functions. These features are then integrated to comprehensively understand the text content in subsequent layers. The convolution layers process text data in parallel, capturing local information of different lengths in sentences. After the convolution layers, max pooling (F.max_pool1d) is used to extract the most significant features, which are concatenated to form the input to a fully connected layer (nn.Linear), mapping the features to classification labels. To prevent overfitting, a dropout layer (nn.Dropout) with a rate of 0.1 is applied before the fully connected layer. During training, the model used cross-entropy loss function (nn.CrossEntropyLoss) and Adam optimizer (torch.optim.Adam). Specific learning rates and training epochs were chosen based on preliminary experiments to balance learning efficiency and computational resource usage.

During training, the model's performance was evaluated through accuracy and loss values. Figure 6 (left) shows the accuracy of the performance of the model on the training and validation sets, exhibiting an overall upward trend despite slight fluctuations on the validation set after 10 training epochs. Similarly, Figure 6 (right) shows the loss values on the training and validation sets, demonstrating an overall downward trend.

The final performance of the model was measured by precision, recall, and F1-score, with the calculated results shown in Table 3. Additionally, the confusion matrix (Figure 7) provides a visual representation of the model's prediction accuracy across different classification labels. The results indicate that TextCNN performed excellently in extracting local text features, with near-perfect precision for Class 0 (one-layer information regulatory texts). Specifically, all samples predicted as Class 0 by the model were correct. Recall was also very high, with only one sample misclassified. The F1-score for Class 0 was close to 1, showing excellent performance for this category. However, the model showed limitations in capturing long-distance text dependencies. For Class 1 and Class 2 (two-layer and three-layer information regulatory texts), the model's precision and recall decreased. The confusion matrix revealed significant confusion between Class 1 and Class 2, with 18% of Class 1 misclassified as Class 2, and 25% of Class 2 misclassified as Class 1. This result

highlights TextCNN’s advantage in capturing local features and its deficiency in capturing sufficient context information.

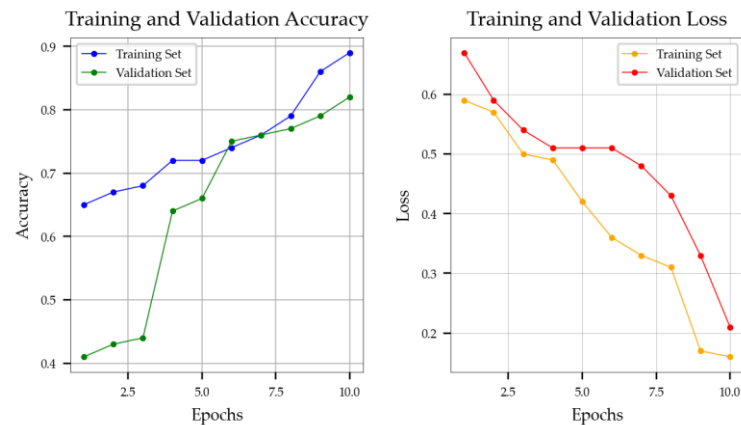


Figure 6. TextCNN training and testing performance.

Table 3. TextCNN Performance Metrics.

	Precision	Recall	F1-Score	Sample
0	1.000	0.990	0.995	100
1	0.759	0.820	0.788	100
2	0.806	0.750	0.777	100

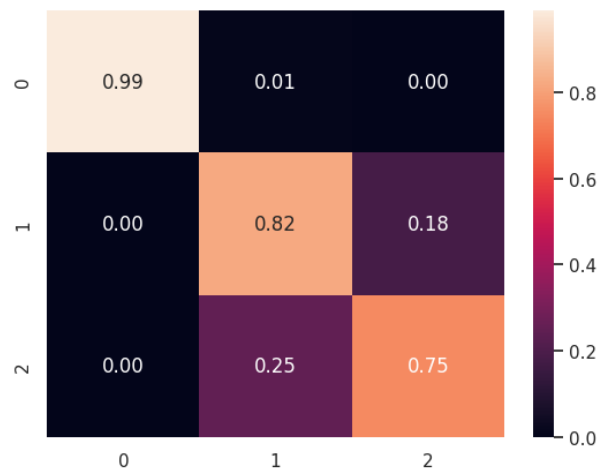


Figure 7. TextCNN confusion matrix.

Overall, the TextCNN model shows significant differences in performance across different categories. It performs excellently in simple text classification tasks (one-layer information), but its performance decreases with increasing information layers in the text. To further improve the model performance, adjustments to kernel sizes, increasing the convolutional layer depth, exploring different regularization techniques, or combining it with other models can address TextCNN’s limitations in capturing long-distance dependencies. Additionally, given the data distribution characteristics of regulatory texts—most being one-layer information—the strong performance of TextCNN in processing these texts justifies its use as a baseline model for evaluating subsequent models.

4.2.2. LSTM

In this study, an LSTM-based neural network model was designed to handle and classify regulatory texts. Similarly, the model first converts each word in the regulatory text

into 100-dimensional vector representations through an embedding layer (nn.Embedding). Following this, an LSTM layer (nn.LSTM) with an output dimension of 50 processes the output of the embedding layer, capturing the sequential characteristics of the text. The output of the LSTM layer is flattened (nn.Flatten) across time steps to form the input to a fully connected layer (nn.Linear), mapping the 2500-dimensional data to classification labels. Like the TextCNN model, the LSTM model was optimized using the cross-entropy loss function (nn.CrossEntropyLoss) and Adam optimizer (torch.optim.Adam), ensuring efficient gradient descent and parameter updates.

The LSTM model was trained over 10 epochs, with the accuracy and loss values on the training and validation sets shown in Figure 8 (left) for accuracy and Figure 8 (right) for loss values. The figures indicate a smooth training process, with overall increasing accuracy and decreasing loss values across 10 epochs, showing no signs of overfitting or underfitting.

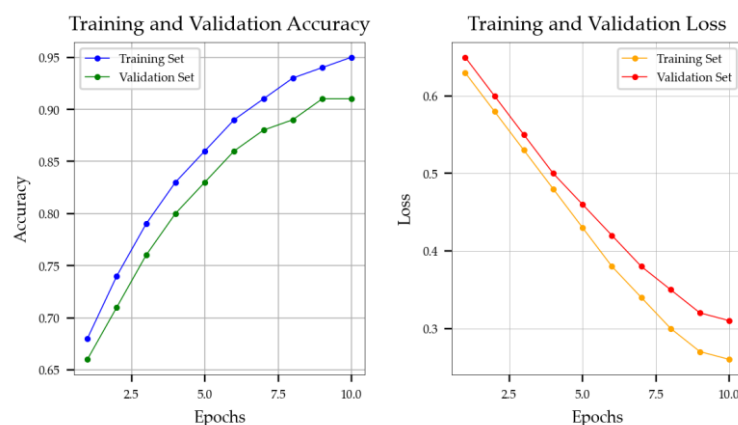


Figure 8. LSTM training and testing performance.

The statistical analysis results of the model's performance are shown in Table 4, presenting precision, recall, and F1-score. The confusion matrix (Figure 9) further provides a detailed view of the model's predictions across different categories. For Class 0, the model's performance was similar to TextCNN, showing perfect precision. Specifically, all samples predicted as Class 0 by the model were correct, with high recall as well, misclassifying only eight samples. The F1-score for Class 0 is 0.995, indicating excellent performance for this category. In capturing long-distance text dependencies, the LSTM model showed superior performance to TextCNN but still has limitations. For two-layer and three-layer information regulatory texts (Class 1 and Class 2), the precision and recall decreased, with the confusion matrix showing significant confusion between Class 1 and Class 2. Specifically, 12% of Class 1 were misclassified as Class 2, and 11% of Class 2 were misclassified as Class 1.

Table 4. LSTM performance metrics.

	Precision	Recall	F1-Score	Sample
0	1.000	0.920	0.958	100
1	0.822	0.880	0.850	100
2	0.881	0.890	0.886	100

The LSTM model shows better recognition capability for different levels of information compared to TextCNN, achieving slightly lower performance in handling simple one-layer information but showing improvement in handling complex regulatory texts. Despite performance improvements, the LSTM model still shows some degree of misclassification, suggesting that even LSTM cannot fully capture all complex dependencies. Future work can explore more advanced regularization techniques, such as variational dropout, or try

more complex LSTM variants like bi-directional LSTM and consider combining LSTM with other types of networks to further enhance its ability to handle complex texts.

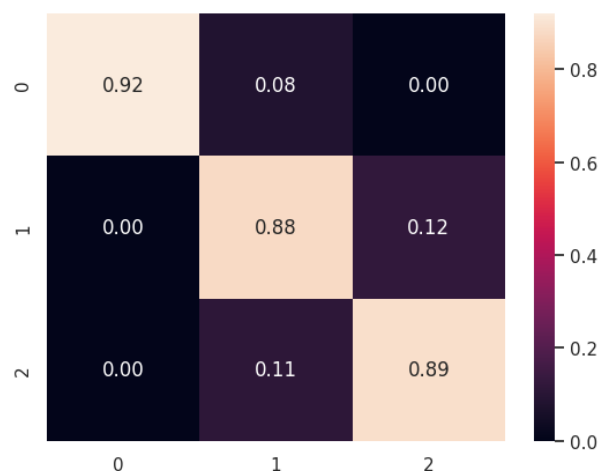


Figure 9. LSTM confusion matrix.

4.2.3. BERT

BERT (Bidirectional Encoder Representations from Transformers) [19] is a pre-trained model based on the Transformer architecture. It learns rich language representations through pre-training on a large corpus and can be fine-tuned for various NLP tasks. The unique aspect of BERT is its bidirectional training structure, allowing the model to consider both the left and right context of each word in the text, capturing deeper semantic information. In regulatory text classification tasks, BERT's advantage is significant as regulatory texts typically have rich context crucial for understanding and classification. Additionally, BERT has shown excellent performance in processing Chinese texts, making it an ideal choice for this study.

This study used the BERT-base-Chinese model as the foundation, a BERT model pre-trained on Chinese corpora. First, the text was processed using BERT's tokenizer (AutoTokenizer), converting it into a format understandable by the model. Then, the pre-trained BERT model block (AutoModelForSequenceClassification) was fine-tuned to output the specified number of classification labels. The model was fine-tuned using the same dataset as the previous models.

The model is defined as a wrapper class (bertModel), encapsulating the pre-trained BERT module. During training, the model was optimized using the cross-entropy loss function and AdamW optimizer, designed specifically for fine-tuning pre-trained models. The learning rate was set to 2×10^{-5} through cross-validation. The model's performance on the training and validation sets is shown in Figure 10 after 10 training epochs. The figures demonstrate BERT's excellent performance in text classification tasks after fine-tuning.

The statistical analysis results of the model's performance are shown in Table 5, including precision, recall, and F1-score. The confusion matrix (Figure 11) provides a detailed view of the model's predictions across different categories. Statistical analysis results showed that for one-layer information regulatory text classification tasks, the BERT model achieved 0.969 precision and 0.94 recall, slightly lower than the TextCNN model but still presenting high performance. In complex regulatory text classification tasks, the BERT model's performance far exceeded TextCNN and previous models, achieving 0.865 precision and 0.900 recall for two-layer information regulatory texts, and 0.929 precision and 0.920 recall for more complex tasks. Overall, the BERT model presents the most balanced and best overall performance among all models.

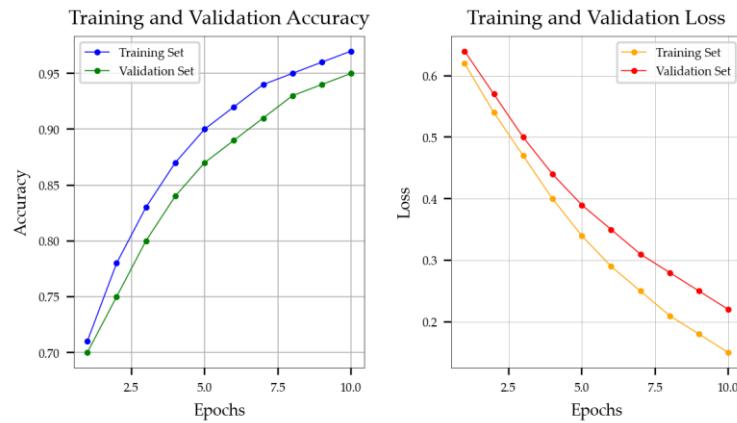


Figure 10. BERT training and testing performance.

Table 5. BERT performance matrices.

	Precision	Recall	F1-Score	Sample
0	0.969	0.940	0.954	100
1	0.865	0.900	0.882	100
2	0.929	0.920	0.925	100

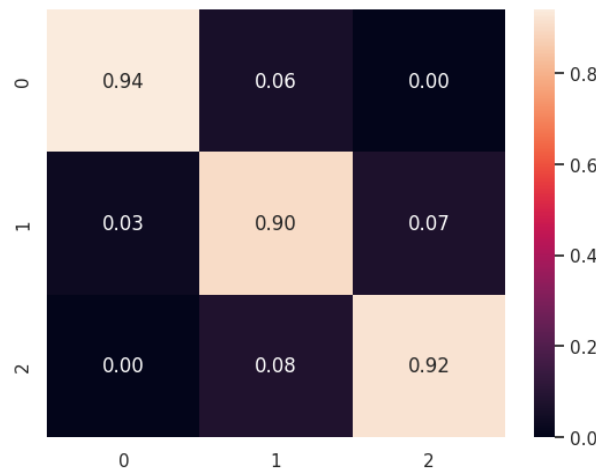


Figure 11. BERT confusion matrix.

Overall, BERT outperforms previous models such as TextCNN and LSTM in handling complex regulatory text classification tasks. BERT’s superior performance in complex tasks is due to its bidirectional training structure and pre-training context, which can deeply understand the complexity and diversity of language. However, in simple regulatory text classification tasks (one-layer information), BERT’s metrics are slightly lower than TextCNN, but still high. This might be because TextCNN is more efficient in handling simple tasks with obvious local features. For these tasks, complex models like BERT may not provide additional advantages, and their high complexity may not be as effective as models focused on local feature extraction. Simple tasks may not fully utilize BERT’s context understanding capability, which is its advantage in complex tasks. Therefore, in future work, combining BERT with models focusing on different aspects, such as TextCNN’s local feature extraction and LSTM’s sequential feature understanding, may provide a more balanced solution for efficiently handling tasks of varying complexity.

4.2.4. Proposed Fusion Model

The three deep learning models constructed in the previous subsections demonstrate different performance levels in regulatory text classification tasks. Specifically, the TextCNN model performs excellently in simple regulatory text classification but shows lower accuracy for texts with multiple layers of information. This may be due to the limitations of convolutional neural networks in capturing long-distance dependencies. The LSTM model performs better than TextCNN in classifying complex texts due to its ability to handle sequential data, especially long-term dependencies. However, its overall accuracy still has room for improvement. The BERT model shows the best overall performance among all models, demonstrating strong feature extraction capability and excellent generalization performance, especially in complex regulatory text classification tasks.

Based on the above analysis, this subsection constructed a fusion model based on TextCNN, LSTM, and BERT, aiming to combine the advantages of each model, such as TextCNN's fast feature extraction and excellent local feature capturing ability, LSTM's ability to handle sequential dependencies, and BERT's strong contextual understanding, to improve the model's classification performance for multi-layer information regulatory texts.

Specifically, this study first initialized the BERT model, using the pre-trained Hfl/Chinese-MacBERT-base model to extract deep semantic information from the text. Then, a CNN module was constructed, containing multiple convolutional layers to capture key local features of the text through different convolution kernels. For the LSTM module, its recursive nature was used to process and memorize long-distance dependencies in the text. The features output by these three models were sent to a Transformer encoder layer, which enhanced the interaction between features from different sources through the self-attention mechanism. Finally, the combined features were classified through a series of fully connected layers. The entire network was optimized using the cross-entropy loss function and the AdamW algorithm, aiming to achieve higher classification performance by combining the strengths of different models.

The accuracy and loss value changes of the model over 10 training epochs are shown in Figure 12, demonstrating the excellent performance of the model after training. The fusion model achieved an accuracy of 0.99 and a loss value of 0.06 on the training set and an accuracy of 0.97 and a loss value of 0.14 on the validation set. These results outperform the individual models in the previous experiments.

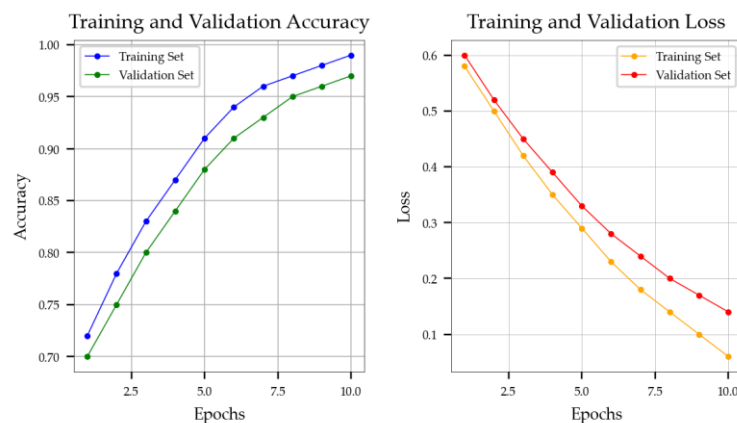
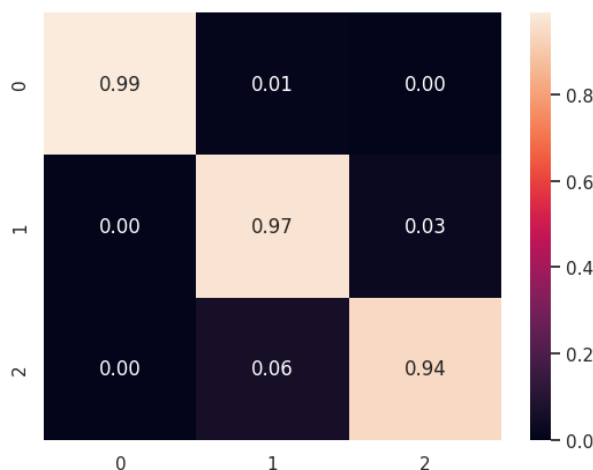


Figure 12. Fusion model training and testing performance.

In terms of statistical metrics, as shown in Table 6, for one-layer information regulatory text classification tasks, the fusion model's performance matches the best performance of the individual models, namely TextCNN. For complex text classification tasks, the fusion model's precision, recall, and F1-score outperform all previous individual models. The confusion matrix (Figure 13) further shows that the fusion model outperforms previous models across various types of clauses.

Table 6. Fusion model performance matrices.

	Precision	Recall	F1-Score	Sample
0	1.000	0.990	0.995	100
1	0.933	0.970	0.951	100
2	0.969	0.940	0.954	100

**Figure 13.** Fusion model confusion matrix.

In summary, the fusion model excels in regulatory text classification tasks, fully demonstrating the integration capability of different models' strengths. The combination of TextCNN's local feature extraction, LSTM's sequential dependency handling, and BERT's contextual understanding makes the model particularly effective in handling complex texts. Despite significant achievements, further exploration of new fusion strategies and learning techniques is needed to optimize the model, especially in data-scarce or insufficiently diverse situations. Future research can focus on improving the model's generalization ability or customizing improvements for specific regulatory texts to achieve more precise text classification and information extraction in practical applications.

4.3. Structured Information Extraction

In this study, GPT-4 from OpenAI was used. For the classification results from the deep learning fusion model, a "one-shot learning" approach was adopted, where a labeled training sample was provided for each category to the LLM for in-context learning (ICL), as shown in Figure 14. These samples were designed to guide the LLM in accurately identifying and understanding the specific structure of different categories of regulatory texts. Specifically, each training sample contained examples of representative regulatory texts and their corresponding labels, enabling the model to recognize the categories these texts belong to during the learning process.

Next, based on the classification results from the deep learning fusion model, appropriate prompts were assigned to each type. This can be achieved using custom Python functions. These prompts further guide the LLM in making accurate classification decisions based on contextual information when processing actual texts. By inputting these precisely labeled samples, the LLM learns how to map regulatory texts to predefined label systems. Additionally, the model parameter "temperature" was set to 0. In AI language models, the "temperature" parameter controls the randomness of the generated content, with a value range between 0 and 1. Setting this parameter to 0 makes the model more likely to choose the highest probability words, generating more deterministic and consistent text. This means the model's responses will be the most likely and least variable, suitable for applications requiring high accuracy and consistency, such as structured information extraction from the regulatory texts used in this study. Conversely, a value closer to 1

would result in more diverse and creative text generation. Therefore, setting this parameter to 0 ensures that the output information is precise and reliable, reducing uncertainty and variability, thus improving efficiency and accuracy in structured tasks.

```

for (x,0):
Original Regulation Text:
"The utilization area of living room should not be less than 10 square meters. "
Expected Output:
{"Verification Conditions":{"obj": 'Living Room','prop': 'hasArea','cmp': '>=', 'value': 10}}
-----
for (x,1):
Original Regulation Text::
"The utilization area of an apartment consisting of a bedroom, living room, kitchen, and bathroom should not be
less than 30 square meters."
Expected Output:
{"Preconditions":{"obj": 'apartment','prop': 'hasRoom','value': ['bedroom', 'living room', 'kitchen', 'bathroom']},
'Verification Conditions':{'obj': 'apartment','prop': 'hasArea','cmp': '>=', 'value': 30}}
-----
for (x,2):
Original Regulation Text::
'The usable area of the kitchen in an apartment consisting of a bedroom, living room, kitchen, and bathroom should
not be less than 6 square meters.'
Expected Output:
{"Preconditions1":{"obj": 'kitchen','prop': 'in','value': 'apartment'},
'Preconditions2':{'obj': 'apartment','prop': 'hasRoom','value': ['bedroom', 'living room', 'kitchen', 'bathroom']},
'Verification Conditions':{'obj': 'kitchen','prop': 'hasArea','cmp': '>=', 'value': 6}}

```

Figure 14. One-shot learning example.

Finally, the output from the LLM needs to be converted from a string format to JSON format for easier subsequent processing. This step can also be achieved using custom Python JSON libraries. This way, the structured information generated by the LLM can be directly integrated into subsequent data processing and analysis workflows. Figure 15 shows some of the structured information extracted from GPT-4. This information includes different categories of regulatory texts and their corresponding labels, clearly demonstrating the efficiency and accuracy of the LLM in identifying and classifying regulatory texts.

```

{"Precondition":{"obj": 'apartment unit','prop': 'contains rooms','value': ['bedroom/living room', 'kitchen', 'bathroom']},
'Check Condition':{'obj': 'apartment unit','prop': 'usable area','cmp': '>=', 'value': 22}}
-----
{"Check Condition":{"obj": 'double bedroom','prop': 'usable area','cmp': '>=', 'value': 9}}
-----
{"Check Condition":{"obj": 'single bedroom','prop': 'usable area','cmp': '>=', 'value': 5}}
-----
{"Precondition1":{"obj": 'kitchen','prop': 'belongs to','value': 'residential unit'},
'Precondition2':{'obj': 'residential unit','prop': 'contains rooms','value': ['bedroom/living room', 'kitchen', 'bathroom']},
'Check Condition':{'obj': 'kitchen','prop': 'usable area','cmp': '>=', 'value': 3.5}}
-----
{"Check Condition":{"obj": 'bedroom/living room','prop': 'usable area','cmp': '>=', 'value': 12}}
-----
{"Precondition1":{"obj": 'kitchen','prop': 'belongs to','value': 'residential unit'},
'Precondition2':{'obj': 'residential unit','prop': 'contains rooms','value': ['bedroom', 'living room', 'kitchen', 'bathroom']},
'Check Condition':{'obj': 'kitchen','prop': 'usable area','cmp': '>=', 'value': 4.0}}
-----
{"Precondition":{"obj": 'bathroom','prop': 'contains facilities','value': ['toilet', 'wash basin']},
'Check Condition':{'obj': 'bathroom','prop': 'usable area','cmp': '>=', 'value': 1.8}}
-----
{"Precondition":{"obj": 'bathroom','prop': 'contains equipment','value': ['toilet', 'shower']},
'Check Condition':{'obj': 'bathroom','prop': 'usable area','cmp': '>=', 'value': 2.0}}

```

Figure 15. Structured information extracted by GPT-4.

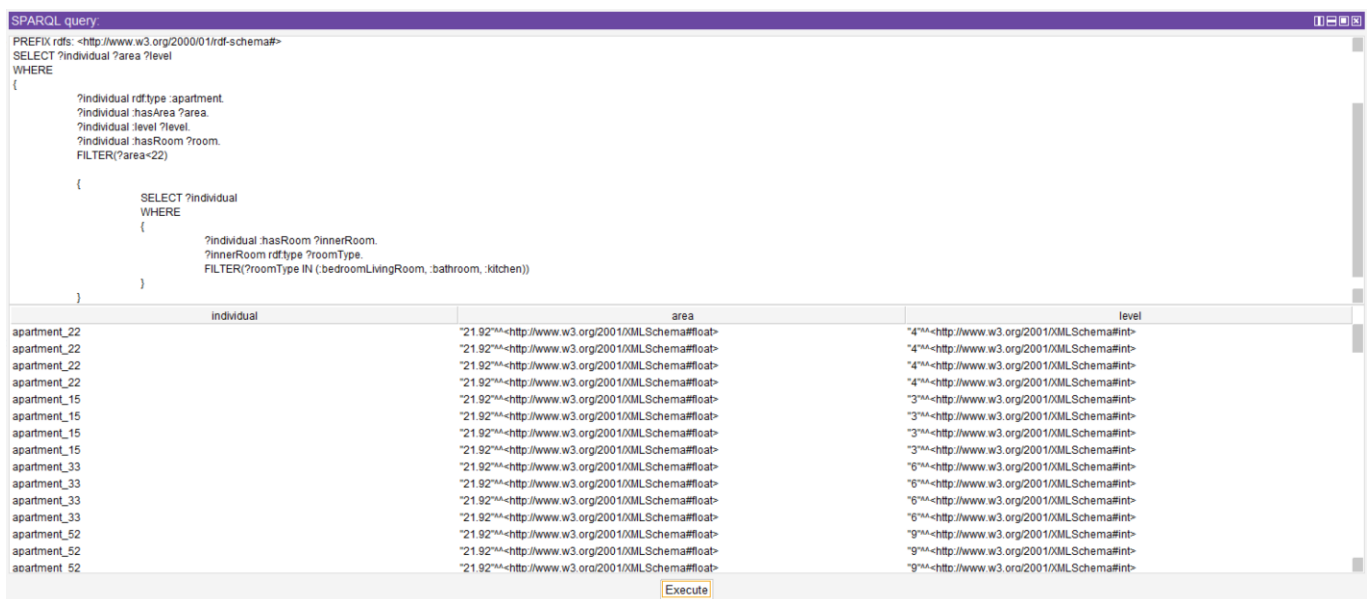
4.4. Compliance Check

This section demonstrates how to implement rule queries using SPARQL in the ontology model constructed in Protégé to identify instances imported from the Revit model that conflict with regulatory clauses.

Based on the output results from the LLM, the regulations to be queried in this study are mainly divided into three types: single-layer information regulations, double-layer information regulations, and triple-layer information regulations. For single-layer information regulations, which involve a single entity type and its attributes, such as “The area of the living room should not be less than 10 square meters”, the corresponding

SPARQL query retrieves all living room entities and filters out instances with an area less than 10 square meters. For double-layer information regulations, which introduce relationships between entities, such as “The total area of an apartment consisting of a bedroom, living room, kitchen, and bathroom should not be less than 30 square meters”, the corresponding SPARQL query retrieves apartments meeting the area requirement and ensures they contain all specified room types. For triple-layer information regulations, which involve combinations of multiple entities and attributes, such as “The minimum usable area of an apartment consisting of a bedroom, living room, kitchen, and bathroom should not be less than 30 square meters”, the corresponding SPARQL query retrieves not only the kitchens and their areas but also confirms that these kitchens are part of a compliant apartment.

Based on the characteristics of these three types of regulations, rule-based Python code was developed to convert the output from the LLM into SPARQL query syntax. The query results from the ontology knowledge model constructed in this study are shown in Figure 16. It can be seen that for the regulatory clause “The minimum usable area of an apartment consisting of a combined living room and bedroom, kitchen, and bathroom should not be less than 22 square meters”, the automated compliance checking framework constructed in this study can quickly identify non-compliant objects in the BIM model used in this research and locate their respective floors, facilitating subsequent adjustments and modifications to the design.



The screenshot shows a SPARQL query window with the following query:

```

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT ?individual ?area ?level
WHERE
{
  ?individual rdfs:type :apartment.
  ?individual rdfs:hasArea ?area.
  ?individual rdfs:level ?level.
  ?individual rdfs:hasRoom ?room.
  FILTER(?area < 22)

  {
    SELECT ?innerRoom
    WHERE
    {
      ?innerRoom rdfs:type ?roomType.
      FILTER(?roomType IN (:bedroomLivingRoom, :bathroom, :kitchen))
    }
  }
}

```

The results table below shows the output of the query:

individual	area	level
apartment_22	"21.92"<http://www.w3.org/2001/XMLSchema#float>	"4"<http://www.w3.org/2001/XMLSchema#int>
apartment_22	"21.92"<http://www.w3.org/2001/XMLSchema#float>	"4"<http://www.w3.org/2001/XMLSchema#int>
apartment_22	"21.92"<http://www.w3.org/2001/XMLSchema#float>	"4"<http://www.w3.org/2001/XMLSchema#int>
apartment_22	"21.92"<http://www.w3.org/2001/XMLSchema#float>	"4"<http://www.w3.org/2001/XMLSchema#int>
apartment_15	"21.92"<http://www.w3.org/2001/XMLSchema#float>	"3"<http://www.w3.org/2001/XMLSchema#int>
apartment_15	"21.92"<http://www.w3.org/2001/XMLSchema#float>	"3"<http://www.w3.org/2001/XMLSchema#int>
apartment_15	"21.92"<http://www.w3.org/2001/XMLSchema#float>	"3"<http://www.w3.org/2001/XMLSchema#int>
apartment_33	"21.92"<http://www.w3.org/2001/XMLSchema#float>	"6"<http://www.w3.org/2001/XMLSchema#int>
apartment_33	"21.92"<http://www.w3.org/2001/XMLSchema#float>	"6"<http://www.w3.org/2001/XMLSchema#int>
apartment_33	"21.92"<http://www.w3.org/2001/XMLSchema#float>	"6"<http://www.w3.org/2001/XMLSchema#int>
apartment_52	"21.92"<http://www.w3.org/2001/XMLSchema#float>	"9"<http://www.w3.org/2001/XMLSchema#int>
apartment_52	"21.92"<http://www.w3.org/2001/XMLSchema#float>	"9"<http://www.w3.org/2001/XMLSchema#int>

Figure 16. Example of SPARQL Query in Protégé.

This section demonstrates through specific rule query examples how to use SPARQL in Protégé to implement rule queries and identify instantiated objects in the BIM model that conflict with regulatory clauses. By carefully designing the queries, the information extracted from the LLM was successfully mapped to the ontology model and effectively conducted compliance checking. This result proves the practicality and efficiency of the automated compliance checking framework proposed in this study, especially in handling complex regulatory information.

The overall case flow from the initial regulatory text to the final compliance checking example is shown in Figure 17. The experimental results reveal the strong potential of the method proposed in this study. Even when faced with complex queries containing multi-layer information, our method can quickly locate and identify non-compliant BIM objects. This not only improves the accuracy of compliance checking but also significantly

shortens the time required for the review process, demonstrating the high efficiency of our method in practical engineering applications.

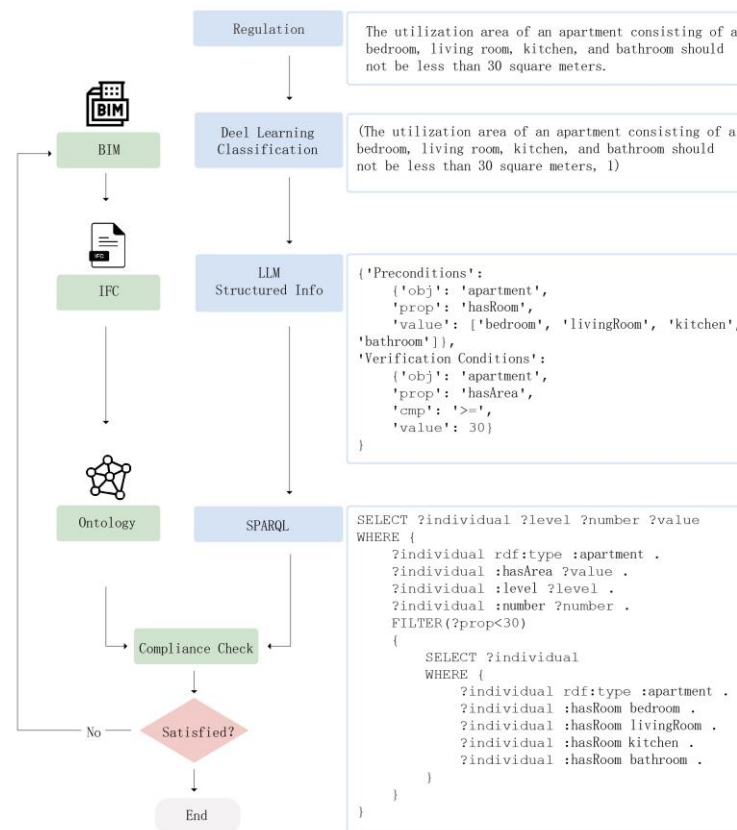


Figure 17. Overall process flow diagram.

5. Discussion

In this study, an innovative framework combining ontology models, deep learning models, and LLMs for automated compliance checking of building BIM was proposed. The framework consists of four main components: lightweight domain ontology model construction, deep learning for preliminary classification of regulatory texts, LLMs for structured information extraction, and compliance checking within the ontology model. Compared to previous studies, the main innovation of this research lies in the combination of LLMs and deep learning models for structured information extraction from regulatory texts. LLMs, with their few-shot learning capabilities acquired through pre-training on large corpora, address the research gaps of requiring large, annotated datasets, scalability, and computational resources. The integration of deep learning improves the accuracy of structured information extraction by LLMs compared to using LLMs alone. This innovative approach enhances the automation and accuracy of the entire compliance checking process, overcoming the limitations of previous methods reliant on manual interpretation and operations, significantly improving the processing speed and efficiency.

5.1. Discussion of Deep Learning Model Text Classification

This study explored the application and performance of various deep learning models in the classification task of building BIM regulatory texts. By conducting a detailed analysis of the performance of different models, we aimed to identify the strengths and limitations of each model in handling regulatory texts with different levels of information complexity. This analysis not only provides us with a deeper understanding of the models' capabilities in processing complex texts but also offers crucial insights for future model selection and optimization. A comparison of the accuracy and loss performance of the four different

models on the training and testing sets is shown in Figure 18. Below is an evaluation and analysis of the performance of TextCNN, LSTM, Transformer, BERT, and the fusion model in this study:

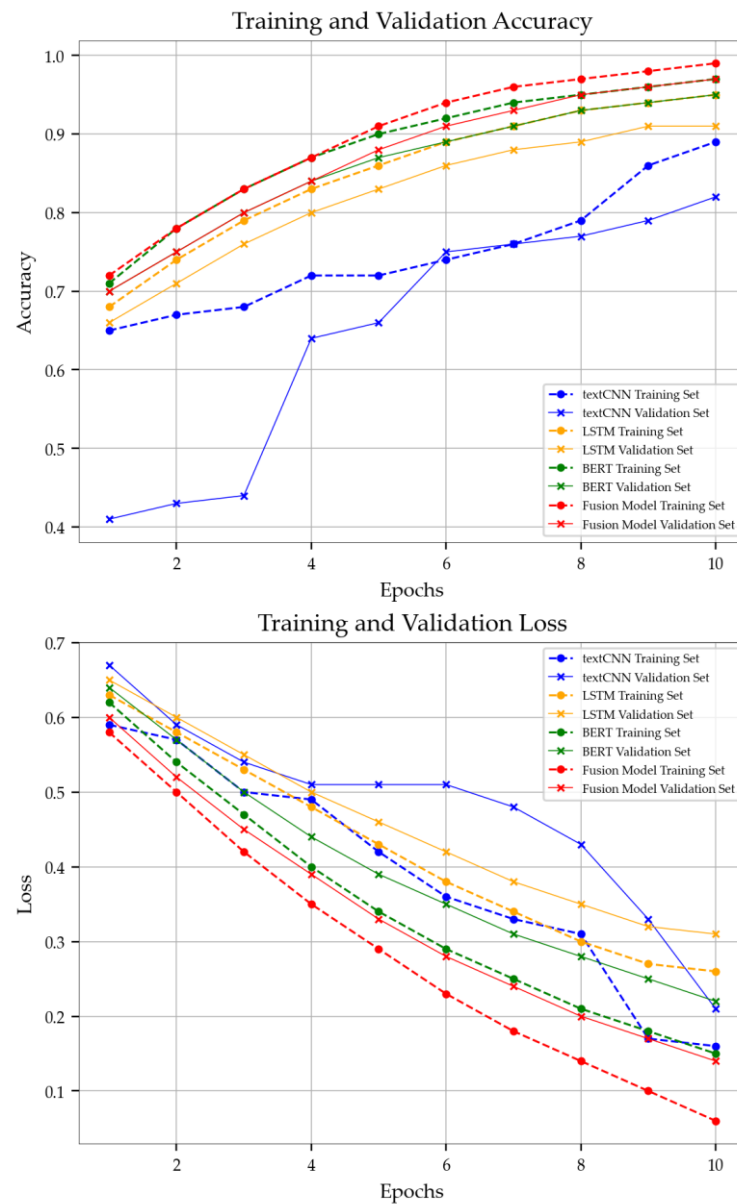


Figure 18. Performance comparison of four models.

- **TextCNN Model:** The TextCNN model showed excellent performance in handling texts with simple structures (i.e., regulatory texts with only one layer of information) but gradually declined in performance when dealing with texts containing more layers of information. To improve the model's performance, adjustments can be made by modifying the convolution kernel size, increasing the depth of convolution layers, adopting different regularization techniques, or combining with other models to compensate for TextCNN's limitations in capturing long-distance dependencies.
- **LSTM Model:** LSTM demonstrated a more balanced capability in handling different levels of information. Although its performance was slightly inferior to TextCNN for simple information layer regulations, LSTM showed a significant advantage in handling complex regulations containing two or more layers of information due to its ability to capture long-term dependencies. Future research could consider using

variants such as bidirectional LSTM or combining it with other network types, such as attention mechanisms, to further enhance its ability to handle complex texts.

- **BERT Model:** BERT outperformed the other models in handling complex regulatory texts, thanks to its bidirectional training structure and deep contextual understanding. However, its performance was slightly lower than TextCNN when dealing with simple regulations, indicating that complex models might not bring additional performance improvements in simple tasks and might be less efficient due to their complexity.
- **Fusion Model:** The fusion model combined the local feature extraction of TextCNN, the sequential dependency handling of LSTM, and the contextual understanding of BERT, showing excellent performance in multi-layer information text classification tasks. Future work should continue to explore new fusion strategies and algorithms to further improve the model's accuracy and generalization capabilities, especially in data-scarce or insufficiently diverse situations.

Each model demonstrated unique characteristics and applicability in regulatory text classification tasks. TextCNN is suitable for handling simple structured texts, while LSTM and BERT show stronger performance in handling multi-layer information texts. Despite Transformer theoretically having high processing efficiency, practical applications require more data support to overcome overfitting issues. The fusion model, by integrating the strengths of different models, exhibited outstanding performance in handling complex text processing tasks. Future research should build on these findings to further explore new model structures and training strategies, particularly in situations with limited data, to effectively improve the model's generalization capability and accuracy. This will not only enhance the efficiency of automated compliance checking but also promote the broader application of deep learning in the field of building information management.

5.2. Discussion of Large Language Model Structured Information Extraction

This study utilized the large language model GPT-4 for structured information extraction from building BIM regulatory texts. Before information extraction, texts were pre-classified using the deep learning fusion model, helping the model to more effectively understand and process different categories of regulatory texts. Additionally, the in-context learning (ICL) strategy was employed, providing two different prompts for each category of regulatory texts, significantly improving the performance of the LLM. The following sections provide a detailed analysis of the main advantages of GPT-4 in this study:

- **Few-shot Learning Capability:** GPT-4 exhibited excellent few-shot learning capability, quickly learning and adapting to new tasks from a limited number of examples. Through the ICL strategy, even for less common regulatory categories, the model could correctly understand and execute information extraction tasks by analyzing the provided few prompts. This capability is particularly useful for regulatory texts where specific information expression may vary across different documents.
- **Language Generation and Understanding:** Due to its extensive training on a wide range of language structures and contexts, GPT-4 excelled in understanding complex language expressions and generating accurate information. When processing building regulatory texts, the model accurately identified and extracted key information such as performance standards and design requirements, while generating clear and precise outputs, crucial for automated document processing.
- **Broad Knowledge Base:** The extensive and diverse datasets encountered during the training of GPT-4 endowed it with a broad knowledge base. This enabled the model to better understand and handle diverse building regulations and technical terms. This characteristic is especially important when dealing with building project documents involving multi-disciplinary knowledge.

GPT-4 demonstrated strong capabilities in structured information extraction from building BIM regulatory texts, particularly in few-shot learning, language understanding and generation, and leveraging its broad knowledge base. Future work can explore further optimizing prompt design, enhancing the model's sensitivity to specific building terms,

and implementing and testing the model's performance in broader building information management systems. Additionally, instructional fine-tuning of the LLM can be considered to eliminate the preliminary classification step by the deep learning fusion model, further improving the efficiency of compliance checking. Through continuous optimization and adjustment, it is expected that this LLM will play a greater role in automated compliance checking and other related fields.

5.3. Limitations and Future Directions

This study successfully achieved an innovative framework combining ontology models, deep learning models, and LLMs for automated compliance checking of building BIM. Although the model demonstrated efficiency and accuracy in handling multi-layer information regulatory texts, especially in structured information extraction using GPT-4, there are still some limitations that need to be addressed in future research:

- **Expansion to Quantitative and Qualitative Regulations:** This study mainly focused on quantitative clauses of regulatory texts, such as area and height requirements, which are parameters that can be directly evaluated through numerical values. These quantitative clauses were effectively handled in the current model. However, the comprehensiveness of building regulations extends beyond this. Qualitative clauses, such as building aesthetics, spatial functionality, or the appropriateness of building layouts, offer more interpretative space and require subjective judgment. For example, "The kitchen should be located near the entrance of the apartment" involves considerations of spatial layout and functionality, which are not only difficult to quantify but may also vary in applicability depending on different contexts. Future research could explore combining qualitative analysis, such as using Natural Language Generation (NLG) technology, to process and interpret these qualitative clauses. This might require developing new algorithms or improving existing models to better understand and apply abstract concepts in these clauses. Additionally, combining qualitative analysis with user experience and expert knowledge could play a key role in automated compliance checking, providing architects and engineers with deeper and more intuitive decision support.
- **Testing Across Different Aspects of Building Design:** This study primarily focused on compliance checking of residential area requirements in building design regulations. To verify the effectiveness of the proposed innovative framework, further testing is required in other aspects of building design, such as structural integrity, electrical installations, and mechanical installations. These areas present different types of regulatory challenges and complexities and testing the framework across these domains will help ensure its robustness and generalizability. Future research should aim to apply the model to these additional areas to fully validate its effectiveness and versatility.
- **Instructional Fine-tuning of Large Language Models:** While GPT-4 demonstrated significant potential in structured information extraction from regulatory texts in this study, its performance still relies on the preliminary classification of regulatory texts using the deep learning fusion model. Future research could focus on more refined "instructional fine-tuning" of these LLMs, aiming to eliminate the preliminary text classification step, further improving efficiency. Instructional fine-tuning is a specific training technique where the model is retrained using a small number of targeted training samples designed to reflect instructions and expected responses in specific application scenarios. Although instructional fine-tuning typically requires high computational power due to the substantial modification of model parameters, techniques like low-rank adaptation (LoRA), which introduce low-rank matrices into the pretrained weights, can significantly reduce the number of parameters to be adjusted, maintaining model performance while greatly improving training and inference efficiency.

By delving into these directions, the current system's limitations can be overcome, continuously enhancing the performance and flexibility of automated compliance checking and promoting the widespread application of deep learning in the field of building information management.

6. Conclusions

This study explored an innovative framework for automated compliance checking of building BIM based on LLMs and ontology knowledge models. By integrating advanced natural language processing techniques and deep domain knowledge, this study successfully proposed and implemented an innovative framework for automating and intelligent processing of compliance checking in the building design process.

First, the technical innovation lies in introducing a combination of deep learning and LLMs into the compliance checking of building BIM models, improving the efficiency and accuracy of compliance checking. This integration not only simplifies the understanding and processing of complex regulatory texts but also makes the extraction of structured information from regulatory texts more accurate and efficient. This methodological development provides a new perspective for handling highly specialized and structured building regulatory texts, enhancing the capability of automated processing.

Second, in practical applications, this study demonstrated the effectiveness of the innovative framework in BIM projects through case studies on specific building projects. These practical application cases not only validated the practicality of the method but also showcased the potential for applying this technology in real-world scenarios.

Furthermore, this study promoted interdisciplinary research and technological development. It demonstrated how to bridge the fields of architecture, computer science, and philosophy, providing innovative ideas for solving complex industry challenges. This interdisciplinary collaboration offers valuable references for future research and technological innovations.

Future research can expand in several directions. On the one hand, since the case studies in this research currently focus on quantitative regulatory clauses such as area and height requirements, the framework can be extended to qualitative regulatory clauses, such as how to automatically identify and process provisions like "The kitchen should be located near the entrance of the apartment", which requires more detailed analysis and understanding of deep semantics in the text. On the other hand, the framework should be tested across different aspects of building design beyond residential area requirements to ensure its robustness and generalizability, like structural integrity, electrical installations, and mechanical installations. Additionally, compared to in-context learning (ICL), instructional fine-tuning of LLMs can be explored to eliminate the preliminary deep learning fusion model classification step, further improving the efficiency of compliance checking. These potential research directions will not only expand the scope and depth of automated compliance checking but also promote the practicality and flexibility of this method in building information model management.

In summary, this study provides an effective method for automated compliance checking of building BIM models, not only contributing to the automation and intelligence of the construction industry but also offering valuable experience and insights for research and practice in other related fields.

Author Contributions: Conceptualization, N.C.; Methodology, N.C. and X.L.; Software, N.C.; Validation, N.C. and X.L.; Formal analysis, N.C.; Investigation, N.C.; Resources, H.J.; Data curation, N.C.; Writing—original draft, N.C.; Writing—review and editing, N.C. and X.L.; Visualization, N.C. and X.L.; Supervision, X.L., H.J. and Y.A.; Project administration, Y.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Solihin, W.; Eastman, C. Classification of rules for automated BIM rule checking development. *Autom. Constr.* **2015**, *53*, 69–82. [[CrossRef](#)]
2. Borrmann, A.; König, M.; Koch, C.; Beetz, J. Building Information Modeling: Why? What? How? In *Building Information Modeling: Technology Foundations and Industry Practice*; Borrmann, A., König, M., Koch, C., Beetz, J., Eds.; Springer International Publishing: Cham, Switzerland, 2018; pp. 1–24.
3. Nawari, N. The Challenge of Computerizing Building Codes in a BIM Environment. In *Computing in Civil Engineering (2012), Proceedings of the 2012 ASCE International Conference on Computing in Civil Engineering, Clearwater Beach, FL, USA, 17–20 June 2012*; Issa, R.R., Flood, I., Eds.; American Society of Civil Engineers: Reston, VA, USA, 2012; pp. 285–292.
4. Eastman, C.; Lee, J.-M.; Jeong, Y.-S.; Lee, J.-K. Automatic rule-based checking of building designs. *Autom. Constr.* **2009**, *18*, 1011–1033. [[CrossRef](#)]
5. Tan, X.; Hammad, A.; Fazio, P. Automated Code Compliance Checking for Building Envelope Design. *J. Comput. Civ. Eng.* **2010**, *24*, 203–211. [[CrossRef](#)]
6. Ismail, A.S.; Ali, K.N.; Iahad, N.A. A Review on BIM-based automated code compliance checking system. In Proceedings of the 2017 International Conference on Research and Innovation in Information Systems (ICRIIS), Langkawi, Malaysia, 16–17 July 2017; pp. 1–6.
7. Hjelseth, E.; Nisbet, N.N. Capturing normative constraints by use of the semantic mark-up RASE methodology. In Proceedings of the CIB W78-W102 Conference, French Riviera, France, 25–28 October 2011.
8. Nadkarni, P.M.; Ohno-Machado, L.; Chapman, W.W. Natural language processing: An introduction. *J. Am. Med. Inf. Assoc.* **2011**, *18*, 544–551. [[CrossRef](#)] [[PubMed](#)]
9. Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F.L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S. GPT-4 Technical Report. *arXiv* **2023**, arXiv:2303.08774.
10. Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. LLaMA: Open and Efficient Foundation Language Models. *arXiv* **2023**, arXiv:abs/2302.13971.
11. Nawari, N. A Generalized Adaptive Framework (GAF) for Automating Code Compliance Checking. *Buildings* **2019**, *9*, 86. [[CrossRef](#)]
12. Fuchs, S. *Natural Language Processing for Building Code Interpretation: Systematic Literature Review Report*; Technical Report for University of Auckland: Auckland, New Zealand, May 2021.
13. Zhang, J.; El-Gohary, N. Semantic NLP-Based Information Extraction from Construction Regulatory Documents for Automated Compliance Checking. *J. Comput. Civ. Eng.* **2013**, *30*, 141013064441000. [[CrossRef](#)]
14. Zhang, J.; El-Gohary, N. Automated Information Extraction from Construction-related Regulatory Documents for Automated Compliance Checking. In Proceedings of the 2011 CIB World Congress, Cape Town, South Africa, 13–14 October 2011.
15. El-Gohary Nora, M.; El-Diraby Tamer, E. Domain Ontology for Processes in Infrastructure and Construction. *J. Constr. Eng. Manag.* **2010**, *136*, 730–744. [[CrossRef](#)]
16. Zhou, P.; El-Gohary, N. Semantic-based text classification of environmental regulatory documents for supporting automated environmental compliance checking in construction. In Proceedings of the Construction Research Congress 2014: Construction in a Global Network, Atlanta, GA, USA, 19–21 May 2014; pp. 897–906.
17. Zhang, J.; El-Gohary, N. Automated Information Transformation for Automated Regulatory Compliance Checking in Construction. *J. Comput. Civ. Eng.* **2015**, *29*, B4015001. [[CrossRef](#)]
18. Zhou, P.; El-Gohary, N. Ontology-based automated information extraction from building energy conservation codes. *Autom. Constr.* **2017**, *74*, 103–117. [[CrossRef](#)]
19. Zhang, R.; El-Gohary, N. A machine learning-based approach for building code requirement hierarchy extraction. In Proceedings of the CSCE Annual Conference, Laval, QC, Canada, 12–15 June 2019.
20. Zhang, R.; El-Gohary, N. A deep neural network-based method for deep information extraction using transfer learning strategies to support automated compliance checking. *Autom. Constr.* **2021**, *132*, 103834. [[CrossRef](#)]
21. Zheng, Z.; Zhou, Y.-C.; Lu, X.-Z.; Lin, J.-R. Knowledge-informed semantic alignment and rule interpretation for automated compliance checking. *Autom. Constr.* **2022**, *142*, 104524. [[CrossRef](#)]
22. Chowdhery, A.; Narang, S.; Devlin, J.; Bosma, M.; Mishra, G.; Roberts, A.; Barham, P.; Chung, H.; Sutton, C.; Gehrmann, S.; et al. PaLM: Scaling Language Modeling with Pathways. *J. Mach. Learn. Res.* **2022**, *24*, 1–113.
23. Taylor, R.; Kardas, M.; Cucurull, G.; Scialom, T.; Hartshorn, A.S.; Saravia, E.; Poulton, A.; Kerkez, V.; Stojnic, R. Galactica: A Large Language Model for Science. *arXiv* **2022**, arXiv:abs/2211.09085.
24. Brown, T.B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language models are few-shot learners. In Proceedings of the 34th International Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 6–12 December 2020; p. 159.
25. Zhao, W.X.; Zhou, K.; Li, J.; Tang, T.; Wang, X.; Hou, Y.; Min, Y.; Zhang, B.; Zhang, J.; Dong, Z. A survey of large language models. *arXiv* **2023**, arXiv:2303.18223.
26. Liu, X.; Li, H.; Zhu, X. A GPT-based method of Automated Compliance Checking through prompt engineering. Presented at 30th European Group for Intelligent Computing in Engineering—‘Towards Sustainable, Smart and Resilient Buildings, Infrastructures and Cities’, London, UK, 4–7 July 2023.

27. GB50096-2011; Design Code for Residential Buildings. China Architecture & Building Press: Beijing, China, 2011.
28. Qiu, J.; Qi, L.; Wang, J.; Zhang, G. A hybrid-based method for Chinese domain lightweight ontology construction. *Int. J. Mach. Learn. Cybern.* **2018**, *9*, 1519–1531. [[CrossRef](#)]
29. Albishre, K.; Albathan, M.; Li, Y. Effective 20 Newsgroups Dataset Cleaning. In Proceedings of the 2015 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT), Singapore, 6–9 December 2015; pp. 98–101.
30. Kim, Y. Convolutional Neural Networks for Sentence Classification. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Doha, Qatar, 25–29 October 2014.
31. Hochreiter, S.; Schmidhuber, J. Long Short-term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)] [[PubMed](#)]
32. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.