University of Nottingham
UK | CHINA | MALAYSIA

A Framework for Knowledge Representation Learning-based Building Control

Kevin Luwemba Mugumya

Thesis submitted to University of Nottingham for the degree of

Doctor of Philosophy

May 2025

Acknowledgements

I owe a big debt of gratitude to the numerous people who have helped me over the last few years with advice, encouragement, and direction as I worked on this thesis. First, I want to thank my main supervisor, Dr Jing Ying Wong, for providing structure to my research and helping me navigate the bureaucratic hurdles in academia and industry. I also want to extend my gratitude to my second supervisor, Prof. Andy Chan, with whom I had several insightful and helpful conversations regarding my work. Prof. Andy's work ethic has been an invaluable example to my overall growth as a researcher. I am also very grateful to Dr. Tomas Maul for agreeing to serve as my thesis's internal and second examiner.

To my industry collaborators from MES Group, you have been the cornerstone of my financial support and exposure to the industry dynamics. You have constantly challenged me to be better, do better, and strive for nothing but the best.

Lusubilo Singogo, you were my unofficial therapist and brainstorming partner in the early days of my PhD. You patiently heard me rant about my PhD and never got tired of it. Thank you for reminding me over our countless beers that quitting was never an option.

To my father, Vincent, you not only introduced me to science and mathematics but also taught me to think creatively. I owe everything that I am to you. My siblings, Susan, Maria, Joseph, Junior, Angela and Cynthia, uncle Dan, my mothers Mary and Betty - your pep talks always landed at exactly the right moments.

My beautiful lady Sonia, you're the calm in my chaos and the reason I keep going. And Solèil Mugumya, our tiny beam of sunlight, you arrived just in time to remind me why finishing matters. This one's for you.

Dedication

This thesis is dedicated to my father, Vincent; my daughter, Solèil; my late mother, Esther; and my late sister, Carine

Abstract

Current Building Automation Systems (BASs) have crucial context-awareness limitations that must be addressed before they can reach human-like levels and better adapt to the dynamic needs of modern buildings. Among other limitations, our buildings still lack sensors, actuators, and control agents that can learn reliable models of the environment and plan complex action sequences. Moreover, modern Machine Learning (ML)-backed BASs, though trained on massive datasets, are usually overly specialised (trained for one task) and brittle (prone to errors). In contrast, human learning is very efficient, and with only a few examples, we can find intuitive ways to complete a task while generalising our knowledge to other tasks. To address the above limitations, this thesis proposes a foundational framework that aims to advance the context-awareness capabilities of BASs using knowledge graphs and Knowledge Representation Learning (KRL). At the framework's core is the notion of using Semantic Web Technologies (SWT) to model the semantic relationships between different building components. These relationships are then packaged inside a network-like data structure called a Building Information Modeling (BIM)-based Knowledge Graph (BIM-KG)[1], and KRL is applied to learn the hidden patterns within the BIM-KG. During the learning phase, KRL utilises message-passing to propagate the learnt information throughout all nodes/entities in the BIM-KG. This research hypothesises that building automation agents can leverage this notion of message-passing to aggregate contextual information from all entities in the graph and use it to continuously update their understanding of a building's systems and components. The perception is that imbuing building automation agents with holistic information about the buildings they control can presumably support context-aware decision-making during downstream automation tasks.

To test the research hypothesis, a *three-phase* investigation was carried out: literature

[1] Knowledge graphs derived from BIM with the help of SWTs are referred to as BIM-based Knowledge Graphs (BIM-KGs) for the remainder of this thesis

review, framework development, and framework applicability. Phase one focused on *situating the research* within the scholarly discourse of BIM, BIM-KGs, building automation, and KRL. The results show that since 2010, SWTs have been a driving force advancing BIM research in the Architecture, Engineering, Construction and Facility Management (AEC/FM) fields by providing the mechanics to represent complex relationships within the built environment. Concurrently, KRL has seen significant development in domains such as bioinformatics, where it has been used to understand complex biological relationships and processes. However, despite the apparent suitability of applying KRL to the BIM field, such integration has not materialised and remains largely unexplored. To get around these research shortcomings, the next phase of this thesis was to *develop a framework* for applying KRL to BIM-KGs using performance analysis experiments. Five baseline KRL models were chosen for this. The chosen models are well-regarded techniques from existing studies, cover a wide range of methodologies, and have been extensively investigated in the context of drug discovery, whose data structures closely mirror those of BIM-KGs. Two publicly available BIM-KGs datasets were used in these experiments. The overall goal was not to identify the best KRL model configurations. Instead, the study examined more closely how model performance can be affected by modifications to the training step, selection of hyperparameters and their optimisation. The experimental results were used to define the prerequisites for integrating KRL with BIM-KGs in a domain-independent framework. This means that although a building automation use case is used to formulate the framework, it can assumingly be applied to other AEC/FM domains such as heritage, quantity-takeoff and energy analysis. The experimental findings show that RotatE and TransE consistently outperform other models across both datasets, establishing themselves as robust baselines when integrating KRL with BIM-KGs. It is also interesting to see that older models like TransE can still be competitive with optimised training and Hyper-parameter Optimization (HPO) configurations. Adam and NSSA emerged as favourable training setup choices, suggesting their potential as initial benchmarks for future evaluations. Despite extensive hyperparameter searches, there was considerable variance among top-performing model configurations, indicating the need for nuanced parameter combinations. This complexity suggests that manual tuning may not yield optimal results, advocating for the adoption of HPO strategies. Furthermore, the disparity in hyperparameters between the two datasets underscores the influence of dataset-specific parameters. Finally,

random search methods, when repeated sufficiently, yield configurations closely comparable to more systematic approaches, albeit in less time.

To illustrate the applicability of the framework, phase three lays out a high-level system architecture consisting of a BIM model, Internet of Things (IoT) devices, and a prototype program of the framework wrapped inside an Application Programming Interface (API). The API consists of a server-side module and a client-side module. The server-side module demonstrates how a building automation system can communicate with KRL configurators, external services such as BIM-KG databases, sensor data stores, and Message Queuing Telemetry Transport (MQTT) brokers. The client-side module consists of a Graphical User Interface (GUI) with a Construction Operations Building Information Exchange (COBie) handler service that facilitates the curation of BIM-KGs from COBie files and an interrogation service that facilitates declarative interrogation of the server-side module using SPARQL Protocol and RDF Query Language (SPARQL) and Graph Query Language (GraphQL).

In conclusion, for KRL to impact the AEC/FM domain, this work emphasizes the critical importance of comprehensively reporting model architectures, training setups, and hyperparameters to enhance trust, reproducibility, and understanding of KRL-based methods among AEC/FM stakeholders and researchers. This insight highlights a prevalent issue in the AEC/FM field where results are often difficult to replicate due to incomplete documentation.

Table of Contents

List of Publications

- Mugumya, K.L., Wong, J.Y., Chan, A. and Yip, C.C. (2019). The role of linked building data (LBD) in aligning augmented reality (AR) with sustainable construction. *International Journal of Innovative Technology and Exploring Engineering*, 8 (S4), pages 366-372.

- Wong, J.Y., Yip, C.C., Mugumya, K.L., Tan, B.H. and Anwar, M.P. (2019). Effectiveness of top-down construction method in Malaysia. *International Journal of Innovative Technology and Exploring Engineering*, 8(6), pages 386-392.

- Mugumya, K.L., Wong, J.Y., Chan, A., Yip, C.C. and Ghazy, S. (2020). Indoor haze particulate control using knowledge graphs within self-optimizing HVAC control systems. *In IOP Conference Series: Earth and Environmental Science (Vol. 489, No. 1, p. 012006).* IOP Publishing.

- Ghazy, S., Tang, Y.H., Mugumya, K.L., Wong, J.Y. and Chan, A. (2022). Future-proofing Klang Valley's veins with REBET: A framework for directing transportation technologies towards infrastructure resilience. *Technological Forecasting and Social Change*, 180.

List of Figures

List of Tables

List of Abbreviations

AEC/FM Architecture, Engineering, Construction and Facility Management

The combined disciplines of architecture, engineering, and construction, along with facilities management. In essence, the term encompasses the full lifecycle of built environments—from initial design and construction to long-term maintenance and operational management.

AHU Air Handling Unit

A critical component within Heating, Ventilation and Air Conditioning (HVAC) systems for conditioning and circulating air, ensuring indoor environments maintain comfort, safety, and energy efficiency.

AMR Adjusted Mean Rank

A performance metric used to evaluate predictive rankings. It is defined as the ratio of the Mean Rank (MR) to the expected MR. It lies on the open interval (0, 2) where lower is better.

API Application Programming Interface

A set of protocols, routines, and tools that allow different software applications to communicate with one another. It defines the methods and data structures needed for this interaction, ensuring that various systems, services, or components can work together seamlessly.

AUC Area under the ROC Curve

A statistical measure, usually calculated from a Receiver Operating Characteristic (ROC) curve, that represents the overall performance of a binary classifier in Machine Learning (ML), indicating how well it can distinguish between positive and negative classes. A higher Area under the ROC Curve (AUC) value *signifies better discriminatory power. Essentially, it reflects the probability that a randomly chosen positive instance will be ranked higher than a randomly chosen negative instance by the classifier.*

BAS Building Automation System

A centralised platform that focuses on controlling and monitoring of a building's various operational components—such as HVAC, lighting, security, and energy management—through automated processes.

BMS Building Management System

A centralised system that encompasses BAS functionalities but also integrates data from these automated systems with facility management and design information, often through Building Information Modeling (BIM) integration. This integration ensures that design, construction, and facility management phases are closely connected for optimal performance throughout the building's lifecycle.

BCEL Binary Cross-Entropy Loss

A loss function commonly used in binary classification tasks, where the goal is to distinguish between two classes. In the context of BIM and AEC/FM, it plays an essential role in developing predictive models for building systems. For instance, when predicting whether an HVAC system is operating normally or experiencing a fault, models can output a probability (between 0 and 1) for each state. Binary Cross Entropy Loss quantifies the difference between these predicted probabilities and the actual outcomes, guiding the model's training to improve accuracy.

BIM Building Information Modeling

A digital process that creates and manages a comprehensive representation of a building's physical and functional characteristics throughout its lifecycle.

BIM-KG BIM-based Knowledge Graph

An advanced representation of building information that uses graph-based data structures to capture, organise, and relate the vast array of data within a BIM model. It extends traditional BIM by embedding semantic relationships between various building elements, systems, and operational data, thus facilitating more intuitive data integration, analysis, and decision-making.

BOT Building Topology Ontology

A minimal ontology for describing the core topological concepts of a building.

CAD Computer-Aided Design

A foundational technology used to create precise digital drawings and models of building components and systems.

CAFM Computer-Aided Facility Management

The use of computer-based systems to manage a building's physical assets, streamline maintenance processes, and optimise operational performance.

CMMS Computerized Maintenance Management System

Software that centralises maintenance information and facilitates the processes of maintenance operations of a built facility.

CNN Convolutional Neural Network

A deep learning architecture that uses convolutional layers to automatically extract and analyse features from visual and spatial data.

COBie Construction Operations Building Information Exchange

A standardised data schema that streamlines the collection and handover of essential building asset information from design and construction to support efficient facility management.

CRUD Create, Read, Update and Delete

The four essential functions used to manage and manipulate data in digital systems.

CWA Closed World Assumption

A principle that presumes any fact not explicitly recorded in a system is false, simplifying data interpretation by treating missing information as non-existent.

DL Description Logic

A family of formal logic-based languages for knowledge representation.

FCU Fan Coil Unit

A device used to heat or cool a room without the need for ductwork. It consists of an indoor coil, a fan and an outdoor condensing unit. The fan forces air through the *indoor coil, which is filled with refrigerant, either cooling or heating it, depending on the desired temperature in the room.*

FM Facility Management

A process that ensures physical assets and environments are managed effectively to meet the needs of their users.

GAT Graph Attention Network

A Graph Neural Network (GNN) variant that uses attention mechanisms to process and learn from graph-structured data.

GCN Graph Convolutional Network

A GNN variant that extends convolution operations to graph-structured data by aggregating and transforming features from interconnected nodes.

GNN Graph Neural Network

A class of deep learning models that directly operate on graph structures, learning complex relationships between interconnected entities.

GraphSAGE Graph Sampling and Aggregation

An inductive GNN algorithm that generates node embeddings by sampling and aggregating features from each node's local neighbourhood, enabling the model to generalise and infer representations for unseen nodes or new subgraphs.

GraphQL Graph Query Language

A flexible query language and runtime for APIs that enables web clients to request precisely the data they need, reducing overhead and improving efficiency.

GUI Graphical User Interface

A visual way for users to interact with complex systems.

HVAC Heating, Ventilation and Air Conditioning

Integrated systems responsible for regulating indoor environmental conditions by managing temperature, humidity, and air quality.

HPO Hyper-parameter Optimization

Techniques to find the best hyper-parameters for a learning algorithm.

IDM Information Delivery Manual

A BIM specification which defines the constraints that determine who provides which information when and to whom for a specific use case.

IFC Industry Foundation Classes

An open, vendor-neutral data model or specification for representing and exchanging building and construction data. It facilitates interoperability among diverse BIM software applications by defining a common format for exchanging detailed information about building elements, systems, and their relationships, thereby supporting the creation of comprehensive digital twins throughout the building lifecycle.

IoT Internet of Things

Interconnected devices that collect and share data over the internet.

KRL Knowledge Representation Learning

A branch of machine learning that focuses on deriving low-dimensional embeddings from structured data, such as knowledge graphs, while preserving the underlying semantic relationships, enabling complex reasoning in downstream tasks.

KRL-based BCF Knowledge Representation Learning-based Building Control Framework

A core deliverable in this thesis and a guiding methodology for integrating KRL techniques into building control systems. This methodology aims to increase transparency of KRL methods in AEC/FM through detailed reporting of model architectures, training configurations, and hyperparameters to foster trust, reproducibility, and enhanced understanding among stakeholders and researchers.

LBD Linked Building Data

The application of Semantic Web and Linked Data principles to integrate diverse building-related datasets using common ontologies and standardised identifiers to enhance interoperability.

LBDCG Linked Building Data Community Group

A collaborative network of industry professionals dedicated to promoting and standardising linked data principles within the AEC/FM and BIM sectors. By fostering open data exchange, shared ontologies, and best practices, the group aims *to improve interoperability, enhance data integration, and enable more informed decision-making in digital twin environments.*

LCWA Local Closed World Assumption

A reasoning approach in which, for a specific subset of a dataset or domain, any fact not explicitly stated is assumed to be false, thereby simplifying inference and decision-making within that local context while acknowledging that the overall knowledge base may be incomplete.

LLM Large Language Model

A type of machine learning model built on deep neural networks with extensive parameters that enable it to understand, generate, and interpret natural language at scale. It is trained on vast text corpora for sophisticated language tasks.

ML Machine Learning

A suite of techniques that enable systems to learn patterns from data and make informed predictions or decisions without being explicitly programmed.

MR Mean Rank

A performance metric that calculates the average rank position of relevant items in a list, serving as a measure of how well a ranking model prioritises desired outcomes.

MRL Margin Ranking Loss

A loss function designed for ranking-based ML tasks. It encourages a predefined margin between the scores of paired inputs.

MRR Mean Reciprocal Rank

A metric that measures how well a system ranks search results. It is used to evaluate the effectiveness of ranking systems in search engines, recommendation systems, and question-answering systems.

MQTT Message Queuing Telemetry Transport

An OASIS standard messaging protocol for the IoT. It is designed as an extremely lightweight publish/subscribe messaging transport that is ideal for connecting remote devices with a small code footprint and minimal network bandwidth.

MVD Model View Definition

A specification that defines a tailored subset of a BIM model, ensuring that only the relevant data is exchanged for specific workflows or applications such as facility management, coordination, or cost estimation.

NRML Non-Relational Machine Learning

A term coined in this thesis to represent a ML paradigm that leverages literal-valued data from a single type of entity, such as all sensors in a building, where each sensor has associated datatype properties (e.g., reading, calibration date or location). Non-Relational Machine Learning (NRML) assumes that the literal values of different entities are independent.

OPM Ontology for Property Management

An ontology for describing temporal properties that are subject to changes as the building design evolves.

OWA Open World Assumption

A principle in knowledge representation that assumes the absence of a statement in a dataset does not imply its falsehood, thereby allowing for the possibility that additional information might exist.

OWL Ontology Web Language

A Semantic Web language for creating and sharing ontologies.

PCB Printed Circuit Board

The baseboard for assembling electronic components and their connections to support many types of electronic devices.

PDF Portable Document Format

A file format for consistent document presentation and exchange.

PM Project Manager

Oversees planning, execution, and completion of a project.

QName Qualified Name

A compact, namespace-aware identifier that combines a prefix with a local name to ensure unambiguous identification of elements on the Semantic Web.

RAG Retrieval-Augmented Generation

A technique that enhances the accuracy and relevance of Large Language Model (LLM) outputs by allowing them to access and reference external knowledge bases before generating a response.

R-GCN Relational Graph Convolutional Network

An application of the Graph Convolutional Network (GCN) framework to modeling and learning from relational data.

RDF Resource Description Framework

A standard model for data interchange that structures information as triples—comprising a subject, predicate, and object—to enable seamless integration, sharing, and semantic querying of diverse datasets.

RDFS Resource Description Framework Schema

A Semantic Web language that extends the Resource Description Framework (RDF) by providing a vocabulary for defining classes, properties, and hierarchical relationships.

RNN Recurrent Neural Network

A deep learning model that is trained to process and convert a sequential data input into a specific sequential data output. Sequential data is data, such as words, sentences, or time-series data, where sequential components interrelate based on complex semantics and syntax rules.

ROC Receiver Operating Characteristic

A performance evaluation tool for binary classification models that plots the True Positive Rate (sensitivity) against the False Positive Rate (1 - specificity) as the decision threshold varies, offering a visual means to assess the trade-off between correctly identifying positives and avoiding false positives.

SAREF Smart Appliances REFerence

An ontology that provides modular building blocks to represent devices in the smart home environment, such as lists of functions, commands and states that can be combined to create complex functionality in a single device.

SGD Stochastic Gradient Descent

An iterative machine learning algorithm that optimises models by using small batches of data to update parameters.

SEAS Smart Energy Aware Systems

An ecosystem of modules that together, provide semantic vocabulary to describe energy systems and their interrelations.

SPL Softplus Loss

A smooth, differentiable loss function that leverages the softplus activation, defined as softplus$(x) = \ln(1 + e^x)$*, to approximate the behaviour of a rectifier and provide stable gradients during optimisation, making it particularly useful in regression or anomaly detection models.*

SSN Semantic Sensor Network

An ontology for describing sensors and their observations, the involved procedures, the studied features of interest, the samples used to do so, and the observed properties, as well as actuators.

SOSA Sensor, Observation, Sampling and Actuator

A lightweight but self-contained core ontology for the Semantic Sensor Network (SSN) ontology.

SHACL Shape Constraints Language

A World Wide Web Consortium (W3C) standard that defines a language for validating RDF graphs against a set of conditions or "shapes," ensuring that data conforms to expected structures and constraints.

SPARQL SPARQL Protocol and RDF Query Language

A standard query language for semantic data in RDF.

SRL Statistical Relational Learning

A subfield of machine learning that uses probabilistic models to capture the uncertainty and dependency structure of entities in linked data. By doing this, these models can predict complex interdependencies among entities in linked data.

SWT Semantic Web Technologies

A common framework that allows data to be shared and reused across different *content and information applications and systems.*

SWRL Semantic Web Rule Language

A proposed language for expressing rules and logic on the Semantic Web.

TPE Tree-structured Parzen Estimator

A Bayesian optimization method for tuning model hyper-parameters.

URI Uniform Resource Identifier

A standardised string used to uniquely identify and locate resources, enabling seamless data integration and interoperability.

VAV Variable Air Volume

An HVAC component that dynamically adjusts airflow to different thermal zones based on varying load demands, enhancing energy efficiency and occupant comfort, and is often integrated with facility management systems for optimised building performance.

WWW World Wide Web

The global system of interlinked hypertext documents and resources that allows access and sharing over the Internet according to specific rules of the Hypertext Transfer Protocol (HTTP).

Chapter 1

Introduction

The Facility Management (FM) life-cycle of buildings is characterised by a continuous flow and exchange of information. The involved parties are predominantly operational building systems, sensor networks, actuators, building occupants, and control agents[1]. At the foundation of each party exists heterogeneous processes that inhibit the seamless flow of *contextually rich information* needed for several downstream FM tasks, of which building automation is the focal point of the investigation herein.

This introductory chapter starts by framing the research context within the boundaries of ongoing efforts to address the issue encapsulated in the above statement. The core research problem is then proposed, followed by its breakdown into more specific research questions. The aim and objectives of the study are then made explicit, and the scope of work is laid out. Finally, this chapter concludes with a summary of the research contributions and the organisational structure for the rest of the thesis.

1.1 Research Context and Motivation

With most people spending 80–90% of their daily lives indoors, buildings have become the largest consumers of global energy due to heavy reliance on heating and air conditioning (Mannan and Al-Ghamdi, 2021). Undoubtedly, the building industry has continued to put pressure on the sustainability equilibrium of the natural environment (Dong et al., 2021; Woods

[1] In this thesis, the term *agent* is used to mean anything that can perceive the built environment around it, take control actions autonomously to achieve a specific set of goals, and may iteratively improve its performance by learning from the information around it.

et al., 2022). Notably, extremely high temperatures and prolonged heat waves have been recorded in many continents and countries (Akompab et al., 2013; Junk et al., 2019; Hopke, 2020; Miller et al., 2021; Barriopedro et al., 2023; Mario et al., 2024). Moreover, the frequency, intensity, and duration of these heat waves are increasing rapidly, making adaptation to heat a priority (Peng et al., 2011; Mitchell et al., 2016; Baniassadi et al., 2018; Alam et al., 2019; Kriebel-Gasparro, 2022).

Global energy efficiency policies and regulations are rapidly evolving to reverse this trend (Zhou et al., 2020b; Viguié et al., 2020; International Energy Agency (IEA), 2023), and the ripple effects are being felt by building owners. They are increasingly being forced to develop buildings characterised by intricate automation systems and swarms of sensor networks toward optimal performance. With this ever-growing complexity of the built environment, so has the *maintenance challenge* increased. Moreover, the already existing stochastic factors in play, such as occupancy behaviour, tightness of the building envelope, and variable weather patterns, only compound this problem. As a result, developing agents with contextually adaptive control policies has become a finicky process that requires exhaustive thought and care. Curry et al. (2012)'s investigation attributed this puzzle to difficulties in identifying and exploiting the inherent latent dependencies between the factors mentioned above.

1.1.1 The Facility Management Challenge

As soon as a building is commissioned, a chain of events is set in motion to ensure the proper functionality of its systems and that operational efficiency goals are met in compliance with established regulations. Over the years, this FM process has steered towards *occupant-centricity*, which not only means that building occupants are getting more engaged in the operation process of embedded building systems, but also optimization targets are not achieved at the expense of their comfort (Park and Nagy, 2018; Park et al., 2019b,a; O'Brien et al., 2020; Park et al., 2022; Jia et al., 2023; Deng et al., 2023). On that basis, FM qualifies to be a *multi-objective optimization problem* that requires a careful trade-off analysis between conflicting objectives (i.e., achieving both operational and energy efficiency while maintaining acceptable indoor air quality and thermal comfort) (Toffolo and Lazzaretto, 2002; Delgarm et al., 2016; Shaikh et al., 2018; Yong et al., 2020; Wang et al., 2023).

Just like any other stage of a building's life-cycle, FM is a heavily data-driven process that involves multidisciplinary stakeholders constantly exchanging and sharing *heterogeneous* information, which is mainly attributed to their departmentalised data handling cultures. Any deficiencies that arise in managing this heterogeneity can arguably propagate to the building systems in the loop, leading to unintended and unexpected under-performing behaviour. The heterogeneity scope addressed in this thesis is discussed in Subsection 1.6.1.

1.1.2 Digitisation of the Facility Management Process

Traditionally, FM information is collated by the design and construction team and piped to the operations team close to the handover stage of a building. At such a time when project budgets and deadlines are soon approaching their elastic limit, perhaps an important question to ask is *"how often is this information checked for completeness, accuracy, or reliability?"*. The answer to this question is arguably *never*. To complicate matters further, some FM information is stored using traditional Computer-Aided Design (CAD) drawings and paper files, making its utilisation cumbersome and inefficient. As a result, building owners started to embrace Computerized Maintenance Management Systems (CMMSs) and Computer-Aided Facility Management (CAFM) systems to capture FM information in a more structured and digitised way. However, even with these, typical day-to-day operational information is usually locked in a sea of Portable Document Format (PDF) files. All these challenges necessitate an efficient mechanism for capturing and propagating FM information from the outset of a building's design and construction to its operational agents.

To an extent, Building Information Modeling (BIM) has served in this role as the primary driver of digitisation in the Architecture, Engineering, Construction and Facility Management (AEC/FM) industries by providing an efficient way of handling large amounts of building information (*semantic* and *geometric*) centrally within a three-dimensional model (Borrmann et al., 2018). However, several obstacles remain on the critical path to sharing this model information *within* and *outside* the AEC industry, which impedes the incorporation of other disciplines into the BIM framework (Pauwels et al., 2017b; Werbrouck et al., 2018). Literature has attributed this exchange bottleneck to the schema design of BIM's data-exchange model, Industry Foundation Classes (IFC) (Barbau et al., 2012; Beetz et al., 2009; El-Mekawy, 2010;

Gómez-Romero et al., 2015; McGlinn et al., 2016). Until 2016, the IFC schema was only available in its native EXPRESS format, which is cumbersome to work with in domain applications such as building automation, geo-spatial, heritage and facility management (Pauwels and Terkaj, 2016; Pauwels and Roxin, 2017).

Specific to FM is the Construction Operations Building Information Exchange (COBie) standard, a subset of IFC which encapsulates the industry's best practices for exchanging FM information between a construction firm and a facility management team (East et al., 2013; Teicholz, 2013). Although COBie's adoption and interest are on the rise, its spreadsheet architecture is cumbersome to navigate (Anderson et al., 2012; Kumar and Teo, 2021a,b), and there are still many misconceptions surrounding its use and, as a result, it is underutilised.

Meanwhile, independent of IFC and outside the AEC/FM industry, other powerful knowledge representation techniques are trending with various disciplines able to interlink their heterogeneous datasets using Semantic Web Technologies (SWT) underpinned by principles of the World Wide Web (WWW) (Berners-Lee, 1996; Berners-Lee et al., 2001; Berners-Lee, 2006). Only recently has there been an increase in research interest in applying this notion to the BIM ecosystem as a mechanism of integrating, managing and extracting value from its heterogeneous data sources (Barbau et al., 2012; Beetz et al., 2009; Pauwels and Roxin, 2017; Pauwels and Terkaj, 2016).

1.1.3 Building Automation in Facility Management

Building automation is ideally a centralised process that involves the automated control of a building's electrical equipment, such as Heating, Ventilation and Air Conditioning (HVAC), lighting and access control, all driven by sensor networks, actuators and control agents, which follow a set of predefined or self-learned control policies.

As mentioned earlier, FM is a multi-objective optimisation problem, and Machine Learning (ML) is a promising solution that is being widely adopted to solve such problems (Toffolo and Lazzaretto, 2002; Asadi et al., 2012; Shaikh et al., 2018; Chen et al., 2018a; Merlet et al., 2022; Wijeratne et al., 2022). At the foundation of ML is the principle of first developing a statistically driven mathematical model, a mechanism for ingesting data in its rawest form while subsequently learning to extract the most relevant information (typically *hidden* *features* and *patterns*) necessary for performing a specific downstream task. But because the building automation domain is highly fragmented, a naive application of ML would lead to models that apply deductions with low precision, efficiency, and scalability. Several proposals anchored by SWT have been put forward in the literature (Pauwels et al., 2018; Pauwels and Terkaj, 2016; Pauwels et al., 2017a; Rasmussen et al., 2019; Pauwels et al., 2022) to alleviate this fragmentation. The resulting semantic glue has made it easier for facility managers to link and holistically analyse data collected across multiple operational building systems. This work also envisages such an integrator as a data fusion strategy that can be integrated into the learning pipeline of building automation ML models towards improved *collective reasoning*. However, this is still in its infancy due to the limited understanding of the peculiarities arising from linking FM data that is encapsulated in BIM-based Knowledge Graphs (BIM-KGs) with ML models. Certain application fields, such as social network analysis, drug discovery in bioinformatics, and fraud detection in e-commerce, often deal with immensely interwoven and complex dataset structures. Knowledge Representation Learning (KRL), and Statistical Relational Learning (SRL) are subsets of ML that have made significant strides in understanding the idiosyncrasies of these datasets (Nickel et al., 2011, 2012, 2016; Lin et al., 2018; Yi et al., 2022). However, the same cannot be said for their application in the FM domain, yet it exhibits similarly intricate datasets. This thesis aims to explore this research direction.

Before presenting the core problem statement of the thesis, it is necessary to delineate the distinction between KRL and SRL. Both are related but are distinct subsets of ML. We shall start with what they both have in common, that is, the mechanics for extracting knowledge from data and representing it in a structured format for downstream tasks. KRL does this by learning a low-dimensional representation of a dataset (typically a knowledge graph) while preserving the underlying semantic meaning (Liu et al., 2016). By contrast, SRL uses probabilistic models to capture the uncertainty and dependency structure of entities in linked data. This approach allows a model to make probabilistic predictions about the relationships between the linked datasets and to reason about the uncertainty of these predictions (Ginestet, 2010). In this thesis, focus is placed on KRL models because they offer two advantages that align well with the research objectives, that is, intrinsic compatibility with knowledge graphs (Lin et al., 2018) and extensibility to deep learning approaches (Wang et al., 2024). The efforts to integrate KRL with BIM-KGs are still very slow, primarily due to the absence of standardised procedures for training and evaluating KRL models within the BIM context.

1.2 Problem Statement

This thesis is primarily driven by the following research question.

How can FM datasets originating from sources inside and outside of a building be efficiently integrated into the self-learning process of building automation agents?

FM datasets are inherently heterogeneous and fragmented. If *expressive* enough mechanisms are orchestrated to *represent* and *unify* these datasets, the resulting analytics have the potential to confirm known FM inefficiencies, shed light on new ones or prove previous hypotheses wrong. Whilst SWTs have emerged as the promising orchestrator to achieve this, so far, their primary focus has been on achieving semantic interoperability for logical inference and complex querying. However, what is still in its infancy is investigating how to leverage the inherent relational structure of semantically inter-linked FM datasets as a mechanism for message passing and information propagation to facilitate *collective contextual reasoning*[2] in building automation agents. In an attempt to bridge this gap, this thesis builds on the work of several previous researchers to propose a *Knowledge Representation Learning-based Building Control Framework (KRL-based BCF).* To the best of the author's knowledge, no attempts have been made to report model architectures, training setups, and hyperparameters to enhance trust, reproducibility and understanding of KRL-based methods among AEC/FM stakeholders and researchers. It is important to note that this framework should be taken as exemplary rather than exhaustive, and a high-level system architecture is presented to demonstrate how the proposed framework can be deployed in practice.

[2] Inter-linked data exhibits patterns and dependencies that occur between attributes and relationships of different entities of the dataset. ML methods that can exploit these patterns *collectively* in their learning pipeline are referred to in this thesis as exhibitors of *collective contextual reasoning.*

1.3 Research Questions

Based on the above problem statement, a design of the following research questions is deemed appropriate to guide the direction of this thesis.

- **Research Question 1 (RQ1)**: *How can knowledge graphs be used to represent the semantic relationships between different building components and systems using domain-agnostic technologies for efficient utilisation in downstream KRL tasks?*

  This research question addresses an important data management problem in the highly fragmented and data-intensive building automation domain. The question is tackled by first analysing the current literature for relevant theories, methods, and tools that have been developed to capture semantic relationships between different building components and systems concerning automation and control. Specific focus is placed on the use of ontologies and SWTs to formulate BIM-KGs while investigating their fit within the boundaries of KRL.

- **Research Question 2 (RQ2)**: *How can KRL be used to learn the relationships formulated in Research Question 1 for building automation?*

  This research question investigates effective ways to integrate and use linked building data (BIM-KGs) in the training and evaluation of KRL algorithms, and how the reliability and robustness of these algorithms can be ensured. To answer this, a literature review is first conducted to investigate the barriers that currently inhibit the use of KRL with BIM-KGs. Experiments are then designed and conducted to assess the nuances of the combination in question. 5 baseline KRL models and 2 publicly available BIM-KGs are used for this.

- **Research Question 3 (RQ3)**: *How can the prerequisites for integrating KRL with BIM-KGs be formalised in a practical framework to enhance trust, reproducibility and understanding of KRL-based methods among AEC/FM stakeholders and researchers?*

To answer this question, the experimental results from *Research Question 2* are used to delineate the prerequisites in question, which are then used to define a step-by-step framework. To illustrate its implementation, a high-level system architecture is devised consisting of a BIM model, Internet of Things (IoT) devices, and a prototype program of the framework wrapped inside an Application Programming Interface (API). Although a building automation use case is used to formulate the framework, the above setup can presumably be used as a reference point for extensibility to other AEC/FM domains.

1.4 Aim

To propose and evaluate a KRL-based BCFs that leverages the inherent relational structure of semantically inter-linked FM datasets to facilitate collective contextual reasoning in building automation agents.

1.5 Objectives

To achieve the above aim, the following research objectives must be met.

1. To explore the use of knowledge graphs to represent the semantic relationships between different building components and systems using domain-agnostic technologies.
2. To investigate the use of KRL to learn the relationships between different building components and systems within the context of building automation and control.
3. To formulate a practical framework that encapsulates the prerequisites for integrating KRL with BIM-KGs.

1.6 Research Scope and Limitations

This section outlines the boundaries of this thesis and clarifies the specific areas in which the proposed framework has been investigated. It sets the context by explaining the primary focus of the research and delineates the key constraints within which this study operates. The limitations identified highlight areas where further exploration or alternative methods may be necessary.

1.6.1 Data Heterogeneity

In this thesis, heterogeneity refers primarily to data originating from diverse sources rather than fundamentally different modalities. This kind of heterogeneity arises because each department, such as operations, maintenance, or asset management, often follows its own data handling culture, standards, file formats, and software tools. This notion of heterogeneity differs from *multimodal* scenarios in deep learning (Bayoudh et al., 2021; Jabeen et al., 2023), where data might span entirely different modalities (such as point cloud data, video, textual specifications, and sensor streams). Instead, here it refers to domain-specific information that, despite originating from multiple sources, remains primarily in structured or semi-structured formats and can be homogenised by a suitable interoperability standard. Thus, while truly multimodal deep learning is undoubtedly valuable, this is not the focal point of the work herein. Future research could explore approaches incorporating multimodal BIM data with advanced deep learning architectures capable of consuming it. Consequently, although this work's findings on BIM data interoperability apply to many AEC/FM scenarios, they may not directly extend to comprehensive, multimodal KRL applications that integrate computer vision, natural language processing, or other advanced deep learning techniques.

1.6.2 Data Modelling

Within the context of BIM-KGs, this thesis targets a specific set of domain-agnostic data modelling approaches that are anchored by SWTs. Rather than developing new data modelling vocabularies (ontologies), this work adopts already existing ones from the Linked Building Data Community Group (LBDCG)[3]. However, due to the overly flexible and open-ended nature of the Semantic Web, the vocabulary choices are guided by carefully crafted competency questions delineating the objectives a BIM-KG needs to satisfy to stay relevant to the KRL problem at hand. In standard ontology development methodologies, *competency questions* usually denote very specific user-oriented queries that an ontology must be able to answer.

[3] `https://www.w3.org/community/lbd/`

In this work, the term has been used more broadly to define overarching semantic objectives rather than narrowly scoped user requirements.

Building a BIM-KG is not a desired output in this thesis, but rather a deep dive into the technical aspects and key considerations for building an effective BIM-KG for training KRL models.

1.6.3 Integration of KRL with BIM-KGs

To investigate the integration of KRL to BIM-KGs, an experimental approach is adopted using 5 baseline KRL models and 2 publicly available BIM-KG datasets. One key motivation for focusing on these baseline models lies in the relatively nascent intersection of KRL and BIM-KGs. While deep learning approaches are undoubtedly state-of-the-art in many other domains, the maturity of KRL applications in the AEC/FM field (particularly involving BIM-KGs) is still very infant. Employing advanced neural architectures prematurely risks overshadowing or missing the fundamental considerations vital to establishing a stable methodological foundation. As a result, this research prioritises a controlled exploration of well-established KRL models—namely ComplEx, DistMult, RotatE, TransE, and TransH—that have proven effective on widely used benchmark datasets in knowledge graph literature (Dai et al., 2020; Bonner et al., 2022; Ge et al., 2023). Another reason is that these baseline models are easier to interpret and are less resource-intensive to implement and tune, making them suitable for a domain that has not yet standardised key aspects of KRL best practices. By demonstrating how these simpler, yet robust approaches perform on BIM-KGs, this work aims to distil essential insights, such as the impact of data quality, hyperparameter selection, and training procedures, that could otherwise be obscured by the greater complexity and heavier computational requirements of deep neural network models. Once these basic principles are clarified and validated, future research will be better equipped to evaluate whether advanced neural architectures offer a practical advantage for this domain, or whether their additional complexity complicates adoption without providing commensurate gains.

Finding the best KRL model configuration is not the goal of the experiments. Instead, they exclusively focus on examining how modifications to the training step, selection of hyperparameters, their optimisation, and initialisation approaches directly affect model performance. The experimental results are used to analyse and formalise the prerequisites for integrating KRL with BIM-KGs in a domain-independent framework. This means that although a building automation use case is used to formulate the framework, it can presumably be extended and applied to other AEC/FM domains. Throughout the development of the framework, various concepts are presented. However, certain concepts have been identified as non-core and beyond the framework's scope, and they have been appropriately classified as such when they arise. Nevertheless, they are discussed because of their potential for offering extensibility to the framework in future research.

1.6.4 Framework Applicability System Architecture

Because the integration of KRL with BIM-KGs is still in an emergent state, the framework developed in this thesis warrants both theoretical and practical scaffolding to guide real-world implementations. The applicability system architecture introduced in this work is only at a high level. It illustrates the conceptual pathways and overarching design considerations for integrating KRL into real-life building automation workflows, but leaves significant room for deeper exploration. Designing a fully implemented low-level setup requires extensive data harmonisation across several heterogeneous systems, real-time sensor integration, and scalable KRL computational workflows—objectives. This goes beyond the scope of this work but remains critical for the broader adoption of the framework. As such, proposing this setup at a higher level is a deliberate first step emphasizing core modules such as KRL configurators, BIM-KG databases, IoT data flows, and user-facing interfaces, while allowing researchers and practitioners the flexibility to tailor specific modules to their local context. This top-down approach also provides a foundational template that others can adapt and refine, whether to different building sizes, regulatory constraints, or occupant interaction models. Future research will need to address the finer details of KRL model deployment. For example, communication protocols in building automation systems often differ substantially, and occupant behaviour injects real-time variability that can complicate KRL-driven insights. A dedicated, low-level prototype in a real-world building could systematically capture these complexities. Moreover, to lower adoption barriers among industry professionals—many of whom may not be *"tech-savvy"*—the setup presented emphasizes a design ethos that prioritizes accessible interaction methods such as Retrieval-Augmented Generation (RAG) and Large Language Models (LLMs) (Gao et al., 2023; Chen et al., 2024) which offer intuitive interfaces for querying, interpreting, and visualizing KRL-driven methods.

1.7 Research Contributions

This research generally provides a foundation for enhancing trust, reproducibility and understanding of KRL-based methods among AEC/FM stakeholders and researchers. The thesis reflects on the technical aspects of using SWTs to formulate BIM-KGs for KRL tasks. This is an extension to what was previously known; using SWTs to achieve semantic interoperability for mainly logical inference and complex querying tasks. The proposed framework aims to provide facility managers with the foundational basis to develop more context-aware building controllers that better adapt to the stochastic building environment. While linking back to the research questions, the explicit contributions of this work are summarised below:

Contributions from RQ1

RQ1: *How can knowledge graphs be used to represent the semantic relationships between different building components and systems using domain-agnostic technologies for efficient utilisation in downstream KRL tasks?*

1. This work demonstrated the construction of BIM-KGs for KRL by

    - Providing a detailed walkthrough on formulating BIM-KGs using domain-agnostic SWTs.
    - Providing a detailed narrative on how to identify small, modular and extensible ontologies for building reusable BIM-KGs and validating them using standardised mechanisms such as Shape Constraints Language (SHACL) and SPARQL Protocol and RDF Query Language (SPARQL).
    - Highlighting common pitfalls that can have cascading effects on the performance of KRL models, such as structural inconsistencies, data incompleteness, and redundancy.

2. This work provided some guidelines for ensuring BIM-KG Data Quality in KRL by
    - Detailing the preliminary checks that ensure appropriate BIM-KG scoping for downstream KRL.
    - Establishing a foundation for AEC/FM researchers to identify further *"data fitness"* criteria relevant to specific KRL use cases beyond this research's building automation use case.

Contributions from RQ2

RQ2: *How can KRL be used to learn the relationships formulated in RQ1 for building automation?*

1. This work provided a performance analysis narrative of KRL Models on BIM-KGs by
    - Conducting extensive experiments with five baseline KRL models on two publicly available BIM-KG datasets, focusing on understanding the nuances that can affect model performance across various training setups and hyperparameter configurations.
    - Identifying RotatE and TransE, coupled with NSSA loss and the Adam optimizer, as robust baselines for building automation scenarios, suggesting their potential as initial benchmarks for future evaluations.
    - Demonstrating that older models, such as TransE, remain competitive with proper Hyper-parameter Optimization (HPO).
2. Provided insights into the nuances of hyperparameter tuning of KRL models by
    - Systematically comparing different hyperparameter selection and optimisation strategies, such as naive random search and a systematic HPO search, to show their distinct impacts on model performance.
    - Revealing the dataset-specific nature of optimal hyperparameter configurations, pointing to the need for flexible, data-specific HPO approaches in BIM-KG contexts.
3. Devised a foundation for enhancing trust and reproducibility of KRL-based methods in AEC/FM by

- Highlighting best practices, such as clear reporting of model architectures, training setups, and hyperparameters, to foster trust and reproducibility among AEC/FM stakeholders.
- Demonstrating that agreed-upon benchmark training datasets together with a transparent and standardised approach to model selection and HPO can lower KRL adoption barriers in the AEC/FM industry.

Contributions from RQ3

RQ3: *How can the prerequisites for integrating KRL with BIM-KGs be formalised in a practical framework to enhance trust, reproducibility, and understanding of KRL-based methods among AEC/FM stakeholders and researchers?*

1. Domain-agnostic KRL - BIM-KG integration framework
   - Developed a step-by-step framework that encapsulates the technical prerequisites (from BIM-KG construction to KRL model training) needed to integrate KRL into real-world AEC/FM workflows.
   - Structured these prerequisites based on empirical insights from the performance analysis experiments in RQ2, ensuring that the framework addresses practical challenges such as data validation and model tuning.
2. Framework applicability system architecture
   - Devised a high-level practical scaffolding setup to guide real-world implementation of KRL-based methods in AEC/FM. This setup includes a BIM model, IoT devices, and a prototype of the framework accessible via an API.
   - Demonstrated how the framework can be extended to other AEC/FM domains, showing its flexibility beyond the initial building automation use case.
3. Resource sharing for further development
   - Published the datasets, trained models, and visualisation materials to encourage the research community to replicate, validate, or improve upon the research.

1.8 Thesis Outline

1. **Chapter 1**: Introduces the research context, motivation, scope, research questions and objectives of the study.

2. **Chapter 2**: Situates the research within the scholarly discourse of BIM, BIM-KGs, and KRL, and outlines the research gaps this thesis addresses.

3. **Chapter 3**: Details the experimental setups, research methods used and their significance in addressing the objectives.

4. **Chapter 4**: Presents a summary of the experimental results and discusses the findings in alignment with the defined research questions.

5. **Chapter 5**: Dissects the findings of Chapter 4 and reflects on them to provide insight into possible future research directions while delineating the limitations faced in this thesis.

Chapter 2

Literature Review

This chapter introduces several fundamental concepts pertaining to the research and provides a detailed background on the research topic, identifying current gaps and outlining potential areas for future research.

2.1 Introduction

Optimising Building Automation Systems (BASs) has been widely studied, but the incorporation of semantic information remains underexplored, despite its potential to enhance *collective contextual reasoning* in building automation agents. This thesis defines collective contextual reasoning as the ability of building automation agents[1] to reason and make decisions based on the aggregated context of a building. This context can encompass heterogeneous parameters ranging from indoor to outdoor, such as the current state of the building, historical data, indoor comfort goals, weather conditions, and occupant behaviour. These parameters also have unknown latent dependencies that may be statistical rather than deterministic. The previous chapter hypothesised that a holistic representation of building information is a prerequisite for building automation agents to infer hidden patterns in building information.

[1] For simplicity, this thesis will sometimes refer to building automation agents simply as *"agents"*

2.2 The Need for Linked Data in the AEC/FM industry

The AEC/FM industry is underpinned by a continuous flow and exchange of information during the design, construction and maintenance of the built environment (Borrmann et al., 2018). This information is usually fragmented and domain-specific due to the complex and departmental nature of the industry, making reliable exchange and stakeholder collaboration a challenge (Pauwels et al., 2018). Furthermore, this fragmentation hinders the integration of expert knowledge among designers, contractors and facility managers, diminishing their opportunity to optimally influence the design, construction and management of a built asset. Nawi et al. (2014) investigated the fragmentation issues of the AEC/FM industry in detail and highlighted the resulting implications on project cost, schedule, dispute handling and unsustainable design-build routines. Autodesk's 2021 FMI report highlighted some surprising figures on how much data and time is wasted in the AEC/FM industry i.e., 95.5% of the construction data goes unused, 13% of the construction professionals' working hours are spent looking for project information, and 30% of AEC/FM companies are using software that does not integrate. (Thomas and Bowman, 2021). Within the building automation context, most optimisation strategies rely on heterogeneous building information that has been generated from various data islands and often exists in unrelated formats. When used in its raw, unintegrated form, this information is ineffective and has a higher probability of being underutilised in many downstream building automation tasks (Borrmann et al., 2018; Pauwels et al., 2018).

2.3 BIM: A Prerequisite for Linked Building Data

The application of digital tools in building operations remains in its infancy, making it one of the biggest missed opportunities in building maintenance today (Borrmann et al., 2018). Traditionally, at handover, facility managers receive piecemeal operational building information using PDF, compact disc and other storage media. As a consequence, this information is often unstructured and semantically insufficient to support many downstream building operation tasks (Zhang et al., 2015; Chen et al., 2018b; Lu et al., 2019a; Mason and Grijalva, 2019). At the heart of the conversation on how this can be solved is BIM, a workflow that effectively handles vast amounts of building information centrally within an intelligent three-dimensional model. The information management protocols offered by BIM dramatically improve the coordination of FM tasks, semantic enrichment of simulation models for training autonomous energy control algorithms (Mason and Grijalva, 2019) and data-driven optimisation of asset designs (Lu et al., 2019b). Furthermore, during operation, BIM reduces the need for facility managers to manually enter asset data into CMMSs. This minimises costly errors, clashes, data loss (see Figure 2.1) and FM blind spots, making it easier for facility managers to locate, interact with, and report on space and asset data.

For lossless data exchange and software interoperability, the BIM ecosystem relies on IFC, an open, vendor-neutral data exchange format developed by buildingSMART. IFC is underpinned by technologies from EXPRESS (ISO 10303-11, 2004), an object-oriented data modelling language specifically designed for product modelling. A detailed description of its structure is provided by ISO 10303-11 (2004) and Pauwels and Terkaj (2016). Due to the comprehensive and generic nature of IFC, it is extremely powerful in catering for the different needs of presenting building information. However, this not only makes it a complex data model but also never entirely complete, i.e. the generic flexibility gives undesired freedom for domain end users and application implementers by limiting the number of problem-specific constraints at the schema level. As a consequence, it is not uncommon for some software import-export routines (see Figure 2.2) to exercise data loss and errors during implementation (Borrmann et al., 2018). In fact, Zhang et al. (2015) highlights how IFC's generality results in the lack of several problem-specific constraints and McGlinn et al. (2016) delineates how IFC does not cover all data structures to meet the requirements of

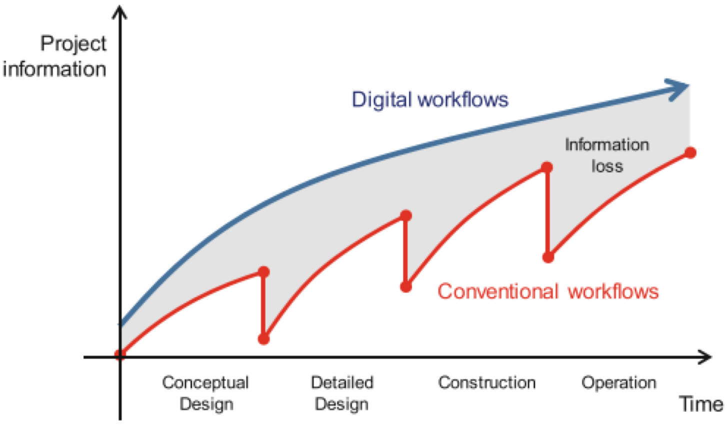


Figure 2.1: Information loss at various stages of the project lifecycle (Borrmann et al., 2018)

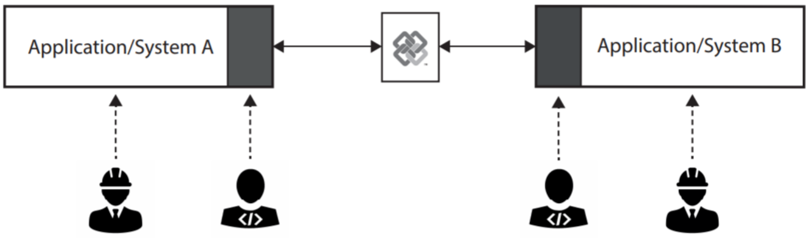


Figure 2.2: IFC exchange (which relies on end-users' modelling expertise) between two BIM software via import-export routines implemented by software developers (Zhang, 2019).

specific energy-management use cases. To satisfy specific data exchange scenarios such as energy simulations, acoustic performance and structural analysis, schema-level constraints are applied to IFC using Information Delivery Manuals (IDMs) and Model View Definitions (MVDs). The constraints determine who provides which information when and to whom for a specific use case. When the IFC schema was initially developed, its authors recognised the necessity for extensibility to accommodate the diverse use cases of the AEC/FM industry (McGlinn et al., 2016; Zhang et al., 2014) as discussed below.

1. The IFC data model is designed with a flexible structure, where many attributes are defined as `OPTIONAL` in the latest IFC specification, IFC4x3_ADD2[2]. This design allows for broad applicability across different domains and lifecycle stages, but also necessitates the use of MVDs to impose stricter requirements for specific data exchange scenarios. Furthermore, MVDs serve to specify which attributes must be populated in a given exchange context to ensure interoperability and compliance with intended workflows.

2. Secondly, IFC provides attribute extension mechanisms via *property sets* and *proxies*. As already mentioned, a syntactically correct IFC instance might miss important attributes for a specific use-case, for example, the `IfcDoor` (an entity for modelling doors in IFC) only has two mandatory attributes: `GlobalId` and `OwnerHistory`, `IfcWindow` only has `GlobalId` as a mandatory attribute. This information can only be used to identify and manage revisions of those object models. All the other information, such as width, height, fire safety class, thermal performance and price, is regarded as unnecessary for the syntactic validity of the underlying data model. This is where *property sets* come in as an extension mechanism by dynamically creating new properties to supplement the already defined static attributes within the schema. The new individual properties

[2] `https://ifc43-docs.standards.buildingsmart.org/`

are defined using IfcPropertySingleValue, a subproperty of IfcProperty, and thereafter grouped into an IfcPropertySet which can be assigned to an object via IfcRelDefinesByProperties. In addition to property sets, IfcProxy serves as a placeholder for dynamically defining semantic information not yet established by IFC (Borrmann et al., 2018).

3. A further means of extending the IFC model is provided by externally referenced properties in libraries such as bSDD (buildingSMART Data Dictionary). SWTs (see Section 2.4) suggested in Zhang et al. (2015); Debruyne et al. (2017); Werbrouck et al. (2018); Pauwels et al. (2018) are also steadily emerging as a means of providing more flexible semantic extension opportunities for the IFC schema.

The above overview is by no means exhaustive but highlights the most significant underlying concepts of IFC data modelling using the EXPRESS language in an easy-to-understand fashion to put the research problem into context.

2.4 Extending BIM with Semantic Web Technologies

The existing BIM software ecosystem is predominantly closed and specifically designed for the AEC/FM sector, which impedes the incorporation of other disciplines into the BIM framework (Werbrouck et al., 2018). Considering that optimisation problems within the industry are reliant on several domain experts who generate a lot of heterogeneous information, having explicit interdisciplinary collaboration is of paramount importance. Unlike domain-specific BIM (Pauwels et al., 2018), a methodology that allows various disciplines to interlink their knowledge on a data level already exists with principles based on the classic *WWW* (Berners-Lee et al., 2001). The common framework that allows such heterogeneous knowledge integration, sharing and re-use is called the *Semantic Web*. It aims to harmonise semantic ambiguity and discrepancies in heterogeneous data schemata by adding standardised machine-readable semantics using the Resource Description Framework (RDF) data model (Berners-Lee et al., 2001). For a building energy optimisation use case, this means that non-geometrical heterogeneous data from other domains can be used to supplement an energy analysis building model with valuable attributes. Homogeneity of this nature cannot be achieved using the BIM's native IFC-EXPRESS schema, therefore necessitating schema translations into open and extensible data structures using Semantic Web Technologies (SWT) such as RDF (Pan et al., 2004; Pauwels et al., 2010; Yang and Zhang, 2006).

The RDF[3] data model is in parallel with object-oriented modelling approaches in IFC, where notions of *entities/classes* related by *associations* are respectively represented in RDF using *concepts* related with *properties* (Pauwels and Terkaj, 2016). Anything described in the semantic web context is called a *resource* and is identified via a Uniform Resource Identifier (URI) (Studer et al., 2007). RDF provides a way of semantically describing these resources by making simple statements about them. These statements are called *triples* and syntactically take the `subject-predicate-object` format (Manola et al., 2014) as shown in Figure 2.3. Multiple statements about the same resource increase its semantic meaning and richness as shown in Figure 2.3 and Figure 2.4 to form a *knowledge graph*. URIs can be very long, making triples less human-readable and may contain prohibited characters for resource labelling. Therefore, Qualified Names (QNames) are often adopted as abbreviations for URIs. A QName has two parts: a namespace and an identifier in the form `namespace:identifier`. To store RDF triples in a compact web-publishable form, several serialisation formats can be used, i.e., Turtle (Beckett and Berners-Lee, 2011), N3 JavaScript (Berners-Lee and Connolly, 2011), RDF/XML (Gandon and Schreiber, 2014) and JSON-LD (Kellogg and Champin, 2019). When several resources related to a specific domain are organised together using formal logics (Baader, 2003; Hitzler et al., 2012; W3C OWL Working Group, 2012), they form an ontology or vocabulary. RDF alone is not expressive enough to describe ontologies, but together with Resource Description Framework Schema (RDFS) and Ontology Web Language (OWL), it is possible. The complexity and vastness of semantic web models necessitate a methodology for searching, filtering out and validating the information from them. SPARQL plays this role both locally and when dealing with federated resources (Harris and Seaborne, 2013; W3C SPARQL Working Group, 2013).

Several early efforts to embrace SWT within the industry emerged with reliance on project-specific ontologies that were hard to reuse or extend formally to other domains because of the different vocabularies and taxonomies employed. Some of these works include the e-COGNOS project from which the e-COGNOS ontology emerged (Wetherill et al., 2002),

[3] `https://www.w3.org/TR/rdf11-primer/`

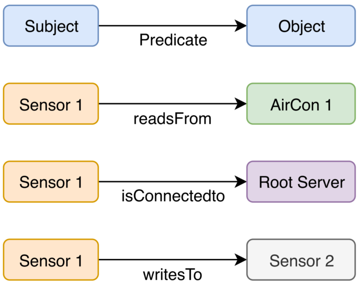


Figure 2.3: RDF triples in the form *subject-predicate-object*. The arrows imply directionality of the relationship.

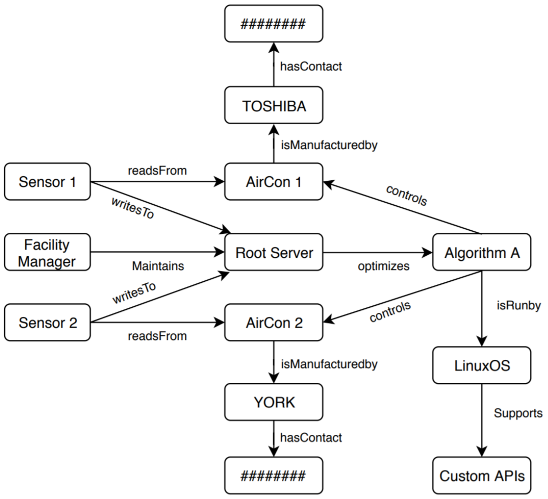


Figure 2.4: An example of an RDF graph (combination of triples) describing some information about sensors in a building connected to different air conditioning units and managed by a root server.

the inteliGrid project ontology for sharing semantics between applications (Dolenc et al., 2007), Yang and Zhang (2006)'s proposal of an early prototype to support interoperability of BIM models and project data, Elghamrawy and Boukamp (2008, 2010)'s ontologically driven model that supports management of and learning from construction problems by holistically integrating project data. Other notable research in this area can be found in Abdul-Ghafour et al. (2007); Le and David Jeong (2016); Pauwels et al. (2010); Scherer et al. (2012); Shah et al. (2011) and Venugopal et al. (2015).

In a push for standardisation, a recommendable and reusable OWL translation of IFC (ifcOWL) was proposed by Pauwels and Terkaj (2016), which was later agreed upon by the Linked Data Working Group (LDWG) (W3C, 2014). Before this, however, several efforts to convert IFC to RDF were made by Agostinho et al. (2007); Beetz et al. (2005); Krima et al. (2009); Pauwels et al. (2015); Schevers and Drogemuller (2005) and Zhao and Liu (2008), whose proposals formed the basis for Pauwels and Terkaj (2016)'s work. The ifcOWL ontology has further been modified by Pauwels et al. (2017a) for a better representation of geometric data. Terkaj and Šojić (2015) proposed an extension to ifcOWL in which EXPRESS WHERE rules were translated to OWL and included in the ifcOWL ontology. In addition, Gómez-Romero et al. (2015) proposed a fuzzy logic-based extension to the ifcOWL ontology that provides support for imprecise knowledge representation and retrieval, which is characteristic of ontologies.

The ifcOWL ontology is very large as it encapsulates the entire IFC schema, and without a doubt, can often prove to be redundant in several use cases or even hard to query. To this effect, W3C's LBDCG (W3C, 2014) has progressively developed simpler, modular and extensible ontologies with intent to cover the IFC schema in smaller and more manageable modules, with Building Topology Ontology (BOT) (Rasmussen et al., 2017b,a) proving to be the most reliable baseline module. BOT serves as the key ontology for capturing the building topology (see Figure 2.5), which is extensible to other domain ontologies like the building device automation domain (Bonino and De Russis, 2018; Schneider, 2017), sensor domain (Haller et al., 2017), geospatial domain (McGlinn et al., 2017), and FM domains.

Specific to building automation, several research efforts have emerged to embrace semantic web approaches in solving energy optimisation problems. For instance, Curry et al. (2012) combined Linked Data with scenario modelling to support interoperability during optimisation of building performance. McGlinn et al. (2016) analysed 33 EU projects that utilised BIM-based energy management plus their data requirements, to identify those that can benefit from open linked data structures. They found that projects in building design, intelligent and customer control, monitoring/visualisation, and building redesign would need a deep exploration of their exact data requirements. Anzaldi et al. (2018) proposed a holistic

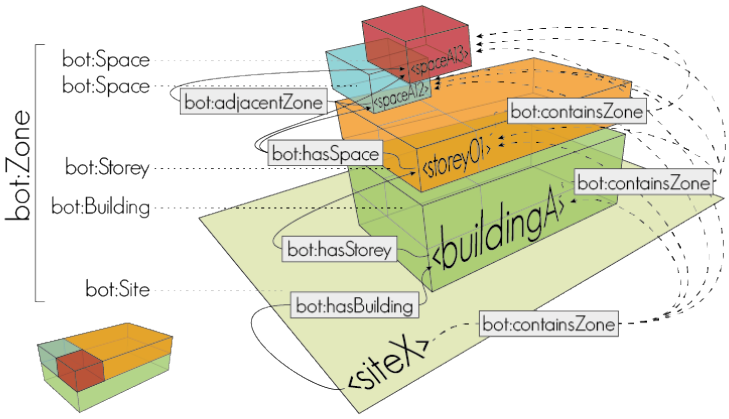


Figure 2.5: Zone connectivity as defined in the BOT ontology (Rasmussen et al., 2017a).

knowledge-based approach for intelligent building energy management using a combination of ontologies, algorithms and simulations. Radulovic et al. (2015) even went ahead to present a set of best practices and guidelines for generating and publishing Linked Data with BIM models in the context of energy consumption in buildings. Corry et al. (2015) and Scherer et al. (2012) developed a performance assessment ontology that structures heterogeneous building data into semantically enriched information, which can support the energy management of buildings. A unified energy representation for smart cities via the DogOnt ontology was proposed by Bonino and De Russis (2018) by integrating several sub-domains of energy representation, namely, electrical, thermal and city-level energy profiles. Dibley et al. (2011) and Dibley et al. (2012) coupled a multi-agent system with an ontology, 'OntoFM', to support real-time monitoring of building sensors in an automated and holistic way. Their work inherited principles from a building ontology based on IFC, a sensors ontology (OntoSensor) (Russomanno et al., 2005) and a general purpose ontology SUMO (Suggested Upper Merged Ontology) (Niles and Pease, 2001), which captures domain-independent concepts. To support interoperability and exchange of data between building energy simulation tools, ‘SimModel’, an XML-based data model, was proposed by O'Donnell et al. (2011). Pauwels et al. (2014b,a) then went ahead to avail this model as RDF graphs which can be combined with other RDF data. Tah and Abanda (2011) developed an ontology to represent information about photovoltaic systems. Reinisch et al. (2011) and Kofler et al. (2012) proposed a comprehensive 'ThinkHome system' that relies on an extensive ontological knowledge base to store all information needed to fulfil goals of energy efficiency and user comfort in future smart homes. This multi-agent system interacts with the knowledge base via SPARQL queries and Description Logic (DL) inference to autonomously control a smart home. Much of the ThinkHome Ontology is inspired by DomoML-env (Sommaruga et al., 2005), an ontology for human-home interaction aiming to connect household appliances and share information about their usage. The aforementioned ontologies can also be combined with a set of Semantic Web Rule Language (SWRL) rules that automatically apply energy management strategies through inference with the knowledge base Rossello-Busquet et al. (2011). Specifically, these rules enable the inference engine to infer if any anomalous activities are occurring (e.g. ‘air conditioners’ that are ‘working’ AND ‘windows’ that are ‘open’). A SPARQL endpoint can even be put on top of this rule engine so that the user only has to query for the results of the rules. Other systems utilising the same SWRL approach to managing smart home appliances have been proposed by Ricquebourg et al. (2007) and Tomic et al. (2010).

2.5 Augmenting BIM-Knowledge Graphs with Machine Learning (ML)

Just like the AEC/FM domain has evolved to embrace SWTs, ML is recognising the growing need to learn from disparate data sources. Wilcke et al. (2017) discusses the current shift in data science from manual feature engineering to utilising raw data, emphasising the need for models that can directly consume and learn from diverse types of information scattered across different domains. To achieve this, a data model capable of naturally expressing heterogeneous knowledge in diverse domains is required, and Wilcke et al. (2017) argues that a knowledge graph is a suitable candidate. For a specific ML task, it is possible to have good data sources with the right information, but without exposing the inherent relationships in the data and adding useful semantics to enhance context, ML models will struggle to deduce informed decisions. Furthermore, by being able to model incomplete knowledge using the Open World Assumption (OWA) (Berners-Lee et al., 2001), knowledge graphs are well suited for modelling real-world data without being concerned how the incompleteness should be dealt with, as is the case with many traditional ML methods that need to employ complex and computationally expensive data imputation techniques when faced with missing or incomplete data (Sterne et al., 2009; Zhou et al., 2024). A knowledge graph can use its intrinsic relationships to gracefully accommodate missing information by providing ML models the ability to reason over the graph structure and infer new connections based on known relationships. This means that a knowledge graph has the flexibility of representing implied facts from explicitly declared knowledge without the need to include the implied triples in the graph. This allows knowledge graphs to achieve high levels of semantic expressivity without being redundant, overly large and complex at the expense of representing many facts.

The emergence of deep learning models has paved the way for workflows that deal with extremely large raw data to automatically learn relevant features without the need for too much pre-processing. Most current models are designed for specific domains, such as image processing (LeCun et al., 1990; Krizhevsky et al., 2012; Le, 2013; Lowe, 1999), sound, or language (Graves et al., 2013; Nguyen and Grishman, 2015). However, they often struggle with heterogeneous knowledge, requiring manual pre-processing—a step where crucial learning information, including hidden relationships, can be lost (Wilcke et al., 2017). Recently, the ML community has taken a keen interest in making the knowledge graph part of the learning process (see Figure 2.6 for a high-level schematic of such learning). Some methods still require a great deal of pre-processing, while others try to work with knowledge graphs more naturally. The former first translates knowledge graphs into feature vectors, which are a more manageable form for many existing learning methods. An example is substructure counting graph kernels (Lösch et al., 2012), a type of algorithm designed to create feature vectors for individual nodes in a knowledge graph. They do this by tallying different types of substructures found near each node in a fashion similar to K-Nearest Neighbour methods in Cunningham and Delany (2007). A drawback of these substructure counting methods is that the size of the feature vector grows with the size of the data, which led to a proposal of RDF2Vec by Ristoski and Paulheim (2016) to handle large graphs more efficiently. More natural workflows of dealing with knowledge graphs include representing triples as a third-order tensor and adopting tensor decomposition methods for collective learning (Kolda and Bader, 2009; Nickel, 2013). Graph Convolutional Network (GCN) (Kipf and Welling, 2016) can also

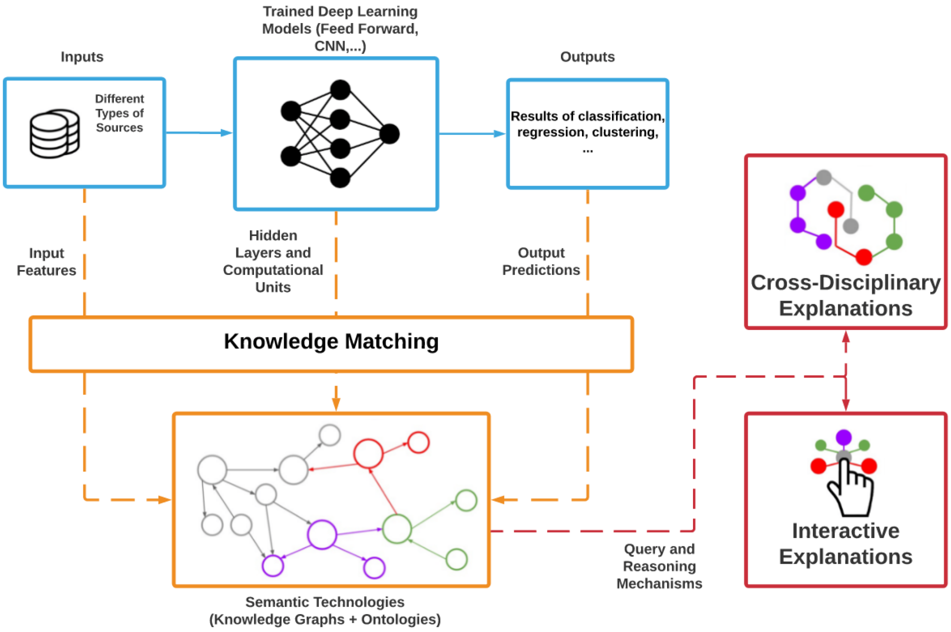

Inputs
Different Types of Sources
Trained Deep Learning Models (Feed Forward, CNN,...)
Outputs
Results of classification, regression, clustering, ...
Input Features
Hidden Layers and Computational Units
Output Predictions
Knowledge Matching
Cross-Disciplinary Explanations
Interactive Explanations
Query and Reasoning Mechanisms
Semantic Technologies (Knowledge Graphs + Ontologies)

Figure 2.6: Schematic representation of a system that integrates SWTs into deep learning (Futia and Vetrò, 2020).

be used to model and learn from relational data more naturally as described by Schlichtkrull et al. (2017). Nickel et al. (2016) provides a very comprehensive review on the use of SRL on knowledge graphs. SRL uses probabilistic models to capture the uncertainty and dependency structure of entities in a knowledge graph. Traditional SRL methods, such as Inductive Logic Programming (Muggleton and de Raedt, 1994), Rule mining (Völker and Niepert, 2011), and graphical models Wainwright and Jordan (2008), have been widely used for learning from graphs. However, these methods suffer from scalability issues as the number of statistical dependencies increases. They also require extensive prior knowledge about the learning task at hand, which can be very computationally expensive to infer if it is not available (Nickel, 2013). One of the biggest challenges of working with knowledge graphs is their lack of spatial locality, meaning their structure cannot be efficiently mapped onto a fixed grid. This is evident in Figure 2.7 and Figure 2.8, which illustrate how images and speech/text sequences naturally conform to fixed grid representations. In contrast, Figure 2.9 highlights the fundamental differences between the topological structure of knowledge graphs and the structured grids used for images, speech, and text sequences.

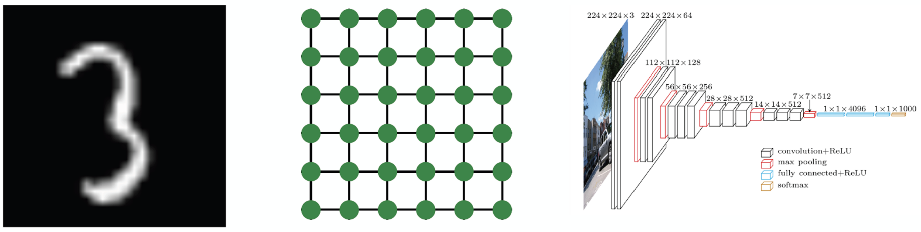


Figure 2.7: Convolutional Neural Network (CNN) models for fixed-size images/grids

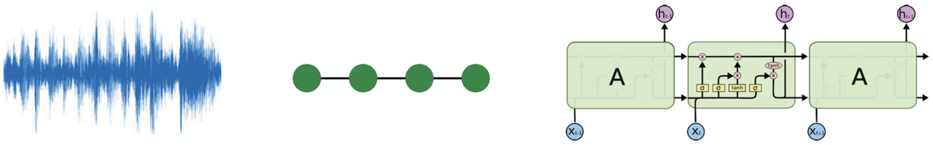


Figure 2.8: Recurrent Neural Network (RNN) models for text/sequences

Furthermore, the graph isomorphism problem, which refers to the difficulty of determining whether two graphs are structurally identical, adds to the difficulty of learning from knowledge graphs. This is a known problem that is neither NP-complete nor solvable in polynomial time, rendering it computationally difficult for traditional SRL methods that rely on manual feature engineering (Corneil and Gotlieb, 1970; Garey and Johnson, 1979; Babai,

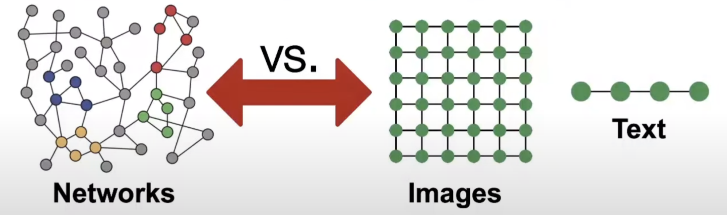


Figure 2.9: Knowledge graphs are of arbitrary size and have a complex topological structure with no spatial locality like grids

2015; An et al., 2024). Also, because knowledge graphs can represent multimodal data such as text, numbers, and timestamps, traditional SRL approaches with limited expressivity struggle to model and learn from such complex representations (Nickel et al., 2011; Nickel, 2013). To overcome these challenges, Knowledge Representation Learning (KRL) methods have gained a lot of traction (Liu et al., 2016; Hamilton et al., 2017; Zhang et al., 2018; Madjiheurem and Toni, 2019; Lin et al., 2018). These approaches aim to learn embeddings for nodes and edges within a knowledge graph without the need for manual feature engineering. Embeddings capture the essential characteristics and relationships of the graph's entities using dense vector representations. These vectors are learned in such a way that similar nodes or edges in the graph have similar embeddings. The key idea behind embeddings is to transform the knowledge graph, which is often complex and high-dimensional, into a lower-dimensional space where latent patterns can be discovered, more easily analysed and utilised in downstream tasks such as link prediction, node classification, and community detection. An important aspect of embedding techniques is the notion of score functions. These are mathematical formulations that assess how likely a triple is to be true based on the learnt embeddings, with a larger score typically implying a more plausible triple. For a triple (h, r, t), the score function $f(\mathbf{h}, \mathbf{r}, \mathbf{t})$ maps it to a scalar value $s \in \mathbb{R}$ that reflects the plausibility of the triple being true. Each entity h and t and the relation r are represented as vectors in a d-dimensional continuous vector space, with embeddings $\mathbf{h}, \mathbf{t}, \mathbf{r} \in \mathbb{R}^d$.

Certain application fields, such as social network analysis (Xu, 2021), drug discovery in bio-informatics (MacLean, 2021), and fraud detection in e-commerce (Shen et al., 2021) often deal with immensely interwoven and complex dataset structures. KRL is one aspect of ML that has made significant strides in understanding the idiosyncrasies of these datasets, however, the same cannot be said for its application in the AEC/FM domain, yet it exhibits similarly intricate datasets. To the best of the author's knowledge, no work has been done to comparatively assess the performance of KRL models when applied to BIM-Knowledge Graphs. To enhance reproducibility, trust and fair comparison of newly developed models against well-established baseline approaches, it is important to report model architectures, training steps, hyperparameters and dataset split mechanisms alongside any performance metrics.

2.6 An Intuitive Mathematical Perspective to Learning from BIM-Knowledge Graphs

For this thesis, a mathematical explanation from both set theory and first-order logic is not only deemed appropriate to define relational data but also highlights the relevance of exploiting the intrinsic relational structure of BIM-Knowledge Graphs in downstream building automation tasks. Relations, in general, define connections between entities, such as whether two rooms have a wall that connects them, whether a person has a specific indoor comfort preference, or whether a sensor is found in a particular space of a building. More precisely, in the domain of set theory and first-order logic, an $\boldsymbol{n}$-ary relation $\mathcal{R}$ over sets $\mathcal{A}_1, \cdots, \mathcal{A}_\mathrm{n}$ is defined as a set of ordered $\boldsymbol{n}$-tuples[4] $\langle a_1, \cdots, a_n \rangle$ where a_i is an element of $\mathcal{A}_\mathrm{i}$ $\forall$ i, $1 \leqslant i \leqslant n$. More intuitively, an $\boldsymbol{n}$-ary relation $\mathcal{R}$ is a subset of the Cartesian product of n sets (Halmos, 1974) (Chapter 7) $\mathcal{A}_1, \cdots, \mathcal{A}_\mathrm{n}$, formally expressed as:

$$\mathcal{R} \subseteq \mathcal{A}_1 \times \cdots \times \mathcal{A}_\mathrm{n} \tag{2.1}$$

The relation $\mathcal{R}$ is interpreted as the set of all *existing* relationships, while the Cartesian product is interpreted as the set of all *possible* relationships over the entities in the domains $\mathcal{A}_1, \cdots, \mathcal{A}_\mathrm{n}$. A single $\boldsymbol{n}$-tuple $\langle a_1, \cdots, a_n \rangle$ therefore represents a possible relationship between the entities $a_1, \cdots, a_n$, which we simply denote by $\mathcal{R}\langle a_1, \cdots, a_n \rangle$. With this background, it is evident that the RDF data modelling structure adopts *binary or dyadic relations* of the form:

$$\mathcal{R} \subseteq \mathcal{A}_1 \times \mathcal{A}_2 \tag{2.2}$$

[4] A tuple is useful for aggregating data that is needed to be considered as a single unit.

There are situations in RDF which require the modelling of $\boldsymbol{n}$-ary relations involving more than two sets of entities. These can be handled efficiently using blank nodes that intrinsically force back a dyadic relational structure. Assume that *entities* of a particular type, for instance, sensors, are encapsulated within a set, $\mathcal{E}_{\mathrm{m}}$. Similarly, let a set $\mathcal{L}_{\mathrm{n}}$ hold possible *literals* values associated with the datatype property of an entity, for instance, a sensor reading, last calibration date of a sensor, U-value of a window glass. Then, any relation $\mathcal{R} \subseteq \mathcal{E}_{\mathrm{i}} \times \mathcal{E}_{\mathrm{j}}$ is an object property while $\mathcal{R} \subseteq \mathcal{E}_{\mathrm{i}} \times \mathcal{L}_{\mathrm{j}}$ is a datatype property. Typical Non-Relational Machine Learning (NRML) settings utilize data that is literal valued and spanning over a single type of entity i.e. consisting relations that take the form $\mathcal{E} \times \mathcal{L}_{\mathrm{j}}$, with $\mathcal{E}$ denoting the set of all entities of the same type and the sets $\mathcal{L}_{\mathrm{j}}$ corresponding to the different datatype properties of these entities. Intuitively, $\mathcal{E}$ could contain all sensors in a building and the sets $\mathcal{L}_{\mathrm{j}}$ could reflect the datatype properties of those sensors like reading, calibration date, location in the building, maintenance date, accuracy etc. NRML makes an independence assumption between the literal values of different entities. For instance the accuracy of a sensor $\mathtt{s}_1 \in \mathcal{E}$ might depend on other datatype properties of this particular sensor like its calibration date, but it is assumed to be independent from the datatype properties of another sensor $\mathtt{s}_2 \in \mathcal{E}$ if $\mathtt{s}_1 \neq \mathtt{s}_2$. However, in a relational learning setting, different entity types can not only exist but also have relationships between them, taking the form, $\mathcal{E}_{\mathrm{i}} \times \mathcal{E}_{\mathrm{j}}$. To put this in context, the previous set of sensors $\mathcal{E}_{\mathrm{i}}$ together with their datatype properties could be complemented by a set of actuators $\mathcal{E}_{\mathrm{j}}$ and a relation $\mathtt{isConnectedTo} \subseteq \mathcal{E}_{\mathrm{i}} \times \mathcal{E}_{\mathrm{j}}$ which indicates which sensor is connected to which actuator. Take, for instance, a sensor observing a certain feature of interest in a building. If this sensor fails and the connected actuator starts deriving wrong control actions, one could implicitly assign credibility of the actuation error to the failed sensor using the existential relation between the two.

2.7 BIM-Knowledge Graph Patterns That are Exploitable for Building Automation

The entity-entity relationships delineated in Section 2.6 introduce rich patterns that can be exploited for collective reasoning in self-learning building automation systems.

Understanding how to extract these patterns is crucial for effective building automation, as they can provide valuable insights that inform decision-making throughout the building's FM lifecycle. In this section, we analyse three key patterns and discuss how they can be used to streamline building automation tasks.

- **Stochastic Equivalence**: Stochastic equivalence suggests that entities exhibiting similar relational patterns can be grouped in clusters (Hoff, 2007). This can be exploited for the analysis of BIM-Knowledge Graphs. For example, when predicting the relationships for a new, yet-to-be-defined component in a BIM model, one can look at the relationships of its cluster members to make an informed prediction. Another example is a Project Manager (PM) finding out that certain stakeholders on their project consistently behave in similar ways based on their cluster memberships, the PM can tailor communication and project management strategies for them accordingly.

- **Homophily**: Social networks are known to be characterised by homophily, the tendency for people from similar backgrounds to connect (Hoff, 2007). Homophilic tendencies can be leveraged to infer unknown relationships in BIM-Knowledge Graphs. For instance, a good covariate to predict the battery life of a sensor in a building might be the battery life of similar sensors in the building.

- **Global Dependencies**: The concept of global dependencies in BIM-Knowledge Graphs plays a crucial role in understanding and managing the complex interrelationships between various entities in a building. Global dependencies can be viewed through the lens of how various components, such as materials, construction methods, schedules, and costs, interact and influence each other. For instance, the success of a building construction project may depend on several factors, including the quality of the materials used, contractor expertise, and compliance with safety regulations.

The presence of these patterns in BIM-Knowledge Graphs illustrates the need for learning approaches that can fully exploit them. Section 2.5 has already highlighted how KRL models can play this role effectively. Some of the most famous KRL models are discussed in Section 2.8.

2.8 Comparative Study of KRL Models

There are several families of KRL models in literature, however, this section will only discuss the most prominent ones and analyse how they compare and contrast with each other. In addition, a synthesis of their strengths and weaknesses will be made concerning building automation.

2.8.1 Graph Neural Networks

A Graph Neural Network (GNN) is a neural model that is designed to learn from graph-structured data such as BIM-Knowledge Graphs. At its core is the concept of message-passing, which allows nodes (entities) to communicate with each other by sending and receiving messages along the edges of the graph. Each node receives messages from its neighbouring nodes, aggregates them, and combines them with its features to generate a new representation (Scarselli et al., 2009; Bronstein et al., 2016). GNNs are often used to solve three types of problems;

1. **Node-level problems**: Here, the focus is on node problems such as node classification, regression, and clustering (Zhou et al., 2020a). Node classification attempts to classify nodes into different groups, for instance, classifying sensors based on their type, location, or function. Node regression involves predicting node property values, for example, predicting the energy consumption of an HVAC system in a building. Node clustering attempts to divide nodes into distinct groups, with similar nodes placed in the same group, for example, grouping sensors that are located in the same area of the building.

2. **Edge-level problems**: GNNs can perform edge-level inferences such as edge classification and link prediction (Zhou et al., 2020a). An example is edge classification can be used to classify the type of relationship between building elements, such as the relationship between a specific sensor and a space. Similarly, link prediction can be used to predict the existence of links between building elements, such as a light switch and a lighting system.

3. **Graph-level problems**: In graph-level tasks, the goal is to classify entire graphs into different categories based on their structural properties (Zhou et al., 2020a). An example would be to determine whether a sensor network has motion sensors, temperature sensors, or air quality sensors. Graph-level tasks include graph matching, graph classification, and graph regression. These can have several applications in the building automation domain. For graph classification, take, for example, fault detection in HVAC systems: the system can be represented as a graph, where each node represents a component (e.g., compressor, evaporator, condenser) and the edges represent their interconnections. Analysing the structural properties of the graph can reveal system anomalies and categorise the graph according to the type of defect.

Graph Convolutional Network (GCN)

A common GNN variant is the GCN, which uses convolutional operations on graphs to capture structural information. GCNs apply a filter that gathers information from each node's immediate neighbours, and this process is repeated for other nodes throughout the graph (Defferrard et al., 2016). This method is computationally efficient for learning representations of local structural information, such as understanding the interactions between a thermostat and the heating units in a specific room for indoor comfort optimisation. For capturing long-range dependencies or broader structural patterns that may be present in complex graphs, GCNs often struggle. Yet, these globalised patterns are prevalent in BIM-Knowledge Graphs. Learning a long-range dependency can involve understanding how a system in one area of the building affects energy use across the entire facility. For example, if a conference room on the ground floor is in use, it could trigger adjustments in lighting and HVAC controls on other floors to optimise the energy efficiency of the entire building. Due to the vanishing gradient problem, the number of convolutional layers that can be used in GCNs is limited. As a result, most state-of-the-art GCN models are no deeper than 3 or 4 layers (Pascanu et al., 2012). Li et al. (2019) presented a proposal for training very deep GCNs by adapting CNN concepts such as residual/dense connections and dilated convolution to GCN architectures. Due to computational constraints, the authors did not explore their proposals in detail. Perhaps another limitation of the vanilla GCN architecture is its inability to handle different edge types in a graph. It assumes a single type of edge and treats all edges equally during the message passing and aggregation process. In a graph with multiple edge types, a vanilla GCN would not be able to distinguish between different types of relationships which are intrinsic to BIM-Knowledge Graphs. Relational Graph Convolutional Network s (R-GCNs) extend GCNs to heterogenous graphs with multiple edge types (Schlichtkrull et al., 2017).

Graph Attention Network (GAT)

Another GNN variant is the GAT. It uses attention mechanisms to learn node representations from a graph. Introduced by Veličković et al. (2017), a GAT weights each node's neighbours based on their significance to the node and aggregates their representations to generate the node's new representation. This notion of attention allows the model to focus on the most important relationships and components, making the predictions more accurate and interpretable. In GCNs, a node updates its features by averaging all the features of its neighbours, treating all neighbour contributions equally. GATs can be computationally expensive due to the need for extra computations to determine the significance of each node or edge. Sparse attention mechanisms have been proposed to reduce the redundancy among edges, allowing GATs to focus only on task-relevant edges for attention calculations (Ye and Ji, 2019). Unlike GCNs, GATs are effective at learning representations that capture both local and global structural information.

Graph Sampling and Aggregation (GraphSAGE)

GCNs and GATs are designed to work with a specific, fixed graph, meaning they create embeddings (representations) for the nodes in that graph only. These frameworks are transductive, meaning that the embeddings they generate are specific to the nodes present in the training graph and cannot be easily extended to new or unseen nodes. As a result, any changes to the graph, such as the addition of new nodes, require retraining or re-computation of the embeddings, which limits their adaptability in dynamic or evolving environments. They also fail to generalise their knowledge across different graphs. Conversely, GraphSAGE is an inductive approach that can generate embeddings for new, unseen nodes or graphs (Hamilton et al., 2017). It utilises available node attribute information to create representations for new data points. In the context of BIM-Knowledge Graphs, inductive capabilities allow the incorporation of new data into the graph as a building's lifecycle changes. For instance, as new materials are introduced, these need to be updated within the BIM-Knowledge Graph, and inductive reasoning can help assess how the new materials might integrate with existing materials, predict their performance, or suggest optimal usages. Similarly, in a construction project management scenario, new nodes for additional stakeholders such as contractors or suppliers can be continuously added to the BIM-Knowledge Graph as the project evolves, and inductive reasoning can use prior knowledge about similar existing stakeholders to predict the influence of the new stakeholders on the project timeline or cost.

2.8.2 Translation Embedding Models

Translational distance models use distance-based scoring functions to assess the plausibility of a fact by measuring the distance between two entities, typically after a translation by the relation (Wang et al., 2017).

TransE and Some of Its Extensions

TransE, proposed by Bordes et al. (2013), is a simple yet effective model for KRL. It represents entities and relationships as vectors in a low-dimensional embedding (vector) space. The key idea of TransE is to interpret relationships as translations between entities. The scoring function of TransE measures the plausibility of a triple (h, r, t) by computing the distance between the head entity h, the relationship r, and the tail entity t in the embedding space. Mathematically, the scoring function for TransE is defined as:

$$f(h, r, t) = ||h + r - t||_2 \tag{2.3}$$

TransE is efficient and fairly easy to implement, making it a popular choice for many KRL tasks. However, it has limitations in dealing with 1-to-N, N-to-1, and N-to-N relations (Wang et al., 2014; Lin et al., 2015). This makes it less suitable for capturing the complexity and heterogeneity of relationships in BIM-based knowledge graphs. Take, for example, a 1-to-N relation, `SensorOf`, meant to represent the existence of a sensor in a specific space. TransE might learn very similar embeddings for `ConferenceRoom`, `Lobby` and `PrayerRoom`, which are all spaces connected to the same type of `TemperatureSensor`, even though they are all different spaces. The same happens for N-to-1 and N-to-N relations. TransH (Wang et al.,

2014) addresses these limitations by allowing an entity to have distinct representations when involved in different relations. This means that even if the embeddings of `ConferenceRoom`, `Lobby`, and `PrayerRoom` might be very similar given the relation `SensorOf`, they could still be far away from each other given other relations. TransH does this by introducing relationship-specific hyperplanes to capture the different transformations associated with different relationships. It models the interaction between entities and relationships on these hyperplanes. The scoring function for TransH is defined as:

$$f(h, r, t) = ||h_{\perp r} + r - t_{\perp r}||_2 \tag{2.4}$$

Where, $h_{\perp r}$ and $t_{\perp r}$ denote the projected representations of the head and tail entities onto the hyperplane associated with the relationship r. Another translational embedding model is TransR (Zhang et al., 2021), which employs relation-specific spaces instead of the hyperplanes used by TransH. While this allows TransR to model complex relations effectively, computational efficiency is sacrificed because a projection matrix is produced for each relation, whereas TransE and TransH rely on vector representations for relations. The scoring function for TransR is defined as:

$$f(h, r, t) = ||M_r h + R_r t - M_r t||_2 \tag{2.5}$$

In this equation, M_r and R_r are the relationship-specific mapping matrices, and $M_r t$ represents the projected representation of the tail entity under relationship r.

2.9 Summary of Research Gaps Identified

The literature review has shown that in recent years, SWTs have been a driving force in advancing the field of BIM, leading to a significant development of BIM-Knowledge Graphs and domain-specific ontologies (data modelling vocabularies). Concurrently, KRL, a promising approach for learning from knowledge graphs, has seen significant development in other domains such as bioinformatics, where it has been used to understand complex biological relationships and processes to deduce new drug discoveries. Despite KRL's success in other domains, its application to BIM-Knowledge Graphs has remained largely unexplored, presenting the research gaps delineated below.

1. **Review of KRL methods within the AEC/FM domain**: The discussion of KRL models in Section 2.8 is not exhaustive; it highlights only a few notable models while offering intuitive context from the BIM and AEC/FM domains. The aim is to inspire AEC/FM researchers to further explore foundational KRL models and their applications in the AEC/FM domain. Therefore, there is a need to thoroughly review the architectures of KRL models while identifying possible entry points and roadblocks into the AEC/FM field.

2. **Developing a Methodology for Applying KRL to BIM-Knowledge Graphs**: There is a need to establish some foundational baselines for the training of KRL models on BIM-Knowledge Graphs to enhance reproducibility, fair comparison of newly developed domain-specific KRL models and their evaluation by future researchers.

3. **Exploring the deployment options of KRL models in the AEC/FM industry**: There is a need to explore and test how best to deploy KRL models for different downstream tasks in the AEC/FM domain. Any scalability or performance issues should also be reported.

4. **Investigating the privacy and security issues arising from the application of KRL to AEC/FM data**: KRL's message-passing formalism could propagate sensitive node information if any to several other parts of a knowledge graph. Further research is needed to investigate mitigation strategies that won't affect model performance in any way, such as data anonymisation to obfuscate sensitive information, differential privacy to add carefully calibrated noise to data or the model's outputs, encryption and role-based access control.

5. **Investigate if KRL's usual performance metrics are applicable to AEC/FM's data in their vanilla form**:

   KRL and AEC/FM being a nascent integration, it requires careful evaluation and validation strategies. Typical KRL evaluation metrics such as Mean Rank (MR), Mean Reciprocal Rank (MRR), Hits@N, Receiver Operating Characteristic (ROC), and Area under the ROC Curve (AUC) may not work ”out of the box” when it comes to AEC/FM evaluations.

Much as this review has identified several gaps, this thesis focuses on gaps 2 and 3 while providing necessary recommendations for closing other gaps.

Chapter 3

Methodology

Taking into account the research questions posed in Section 1.3 and the research gaps identified in Section 2.9, this chapter presents a detailed narrative of the steps taken to develop and implement the Knowledge Representation Learning-based Building Control Framework (KRL-based BCF). For better understanding, this methodology is broken down into 2 core steps (see Figure 3.1).

1. **Linked Building Data (LBD) modelling**: This section walks through a prototypical data modelling example to delineate the technical aspects and key considerations for building an effective BIM-KG for training KRL models. Although a building automation use case is used, the same steps can presumably be used for other domains such as heritage, quantity-takeoff and energy analysis.

2. **Performance analysis of KRL on LBD**: This section explores the integration of KRL with LBD (BIM-KGs), focusing on the use of performance analysis experiments whose goal is not to identify the best KRL model configurations, but rather examine more closely how model performance can be affected by modifications to the training step, selection of hyperparameters, their optimization and initialization approaches.

The experimental results from step 2 are used to define the prerequisites for integrating KRL with BIM-KGs in a domain-independent framework. Again, although a building automation use case is used to formulate the framework, it is extensible to other AEC/FM domains, and a prototype will be presented to illustrate such extensions while assessing the feasibility and applicability of the framework.

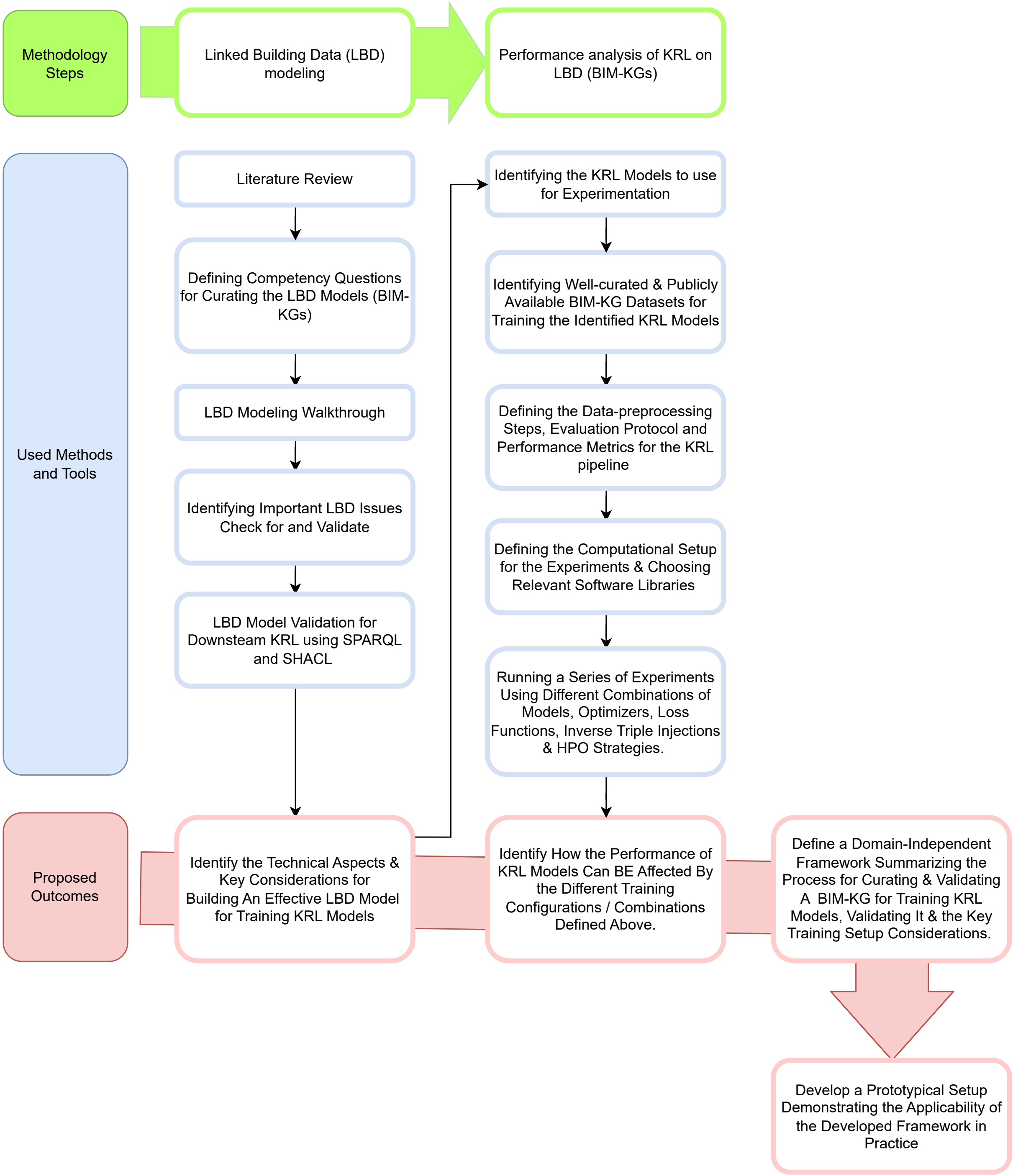

Methodology Steps
Linked Building Data (LBD) modeling
Performance analysis of KRL on LBD (BIM-KGs)
Used Methods and Tools
Literature Review
Defining Competency Questions for Curating the LBD Models (BIM-KGs)
LBD Modeling Walkthrough
Identifying Important LBD Issues Check for and Validate
LBD Model Validation for Downsteam KRL using SPARQL and SHACL
Identifying the KRL Models to use for Experimentation
Identifying Well-curated & Publicly Available BIM-KG Datasets for Training the Identified KRL Models
Defining the Data-preprocessing Steps, Evaluation Protocol and Performance Metrics for the KRL pipeline
Defining the Computational Setup for the Experiments & Choosing Relevant Software Libraries
Running a Series of Experiments Using Different Combinations of Models, Optimizers, Loss Functions, Inverse Triple Injections & HPO Strategies.
Proposed Outcomes
Identify the Technical Aspects & Key Considerations for Building An Effective LBD Model for Training KRL Models
Identify How the Performance of KRL Models Can BE Affected By the Different Training Configurations / Combinations Defined Above.
Define a Domain-Independent Framework Summarizing the Process for Curating & Validating A BIM-KG for Training KRL Models, Validating It & the Key Training Setup Considerations.
Develop a Prototypical Setup Demonstrating the Applicability of the Developed Framework in Practice

Figure 3.1: The research methodology overview

3.1 Linked Building Data (LBD) Modelling

This work adopts domain-agnostic SWTs to demonstrate the process of developing a BIM-KG within the building automation domain while using carefully crafted competency questions to scope the specific objectives that the BIM-KG needs to satisfy.

3.1.1 Competency Questions as an LBD Modeling Guide

Rather than develop new data modelling vocabularies (ontologies), this work adopts existing vocabularies from the LBDCG (W3C-Linked Data Community Group, 2018). A specific ontology of interest is the ifcOWL ontology (Pauwels and Terkaj, 2016). As per the IFC4 Add2 release[1], ifcOWL has about 770 classes, approximately 1190 object properties, and around 60 datatype properties. This granularity is one of ifcOWL's main strengths, offering a level of detail that surpasses many other BIM ontologies. It caters to a wide array of applications from architectural design to facility management. However, this granularity can pose challenges, as the numerous classes and properties, along with their complex relationships, can become unwieldy for certain smaller use cases. Additionally, there is a risk of higher computational demands for querying and inference, potentially hindering performance in real-time building automation tasks applications. Given this, smaller and more focused ontologies, some branching off ifcOWL need to be adopted. Throughout this section, precise competency questions are crafted to guide the choice of modular ontologies and to help maintain focus on the objectives that the curated LBD model (BIM-KG) has to fulfil. In standard ontology development methodologies, *competency questions* usually denote very specific user-oriented queries that an ontology must be capable of answering. In this work, however, the term has been used more broadly to define overarching semantic objectives rather than narrowly scoped user requirements.

1. **Competency Question 1 (CQ1)**: *How to semantically describe the high-level concepts of a building in a way that formulates a semantic extension baseline for describing low-level building information relevant for indoor environment monitoring and control?*

   To answer this question, the Building Topology Ontology (BOT)[2] (Rasmussen et al., 2017a) is employed as it is deemed appropriate for encoding relationships between the main components of a building (site, building, storey and space) using a highly modular and simplistic set of semantic blocks. In BOT, a building consists of zones in a hierarchy. The subclass of a zone is a site which contains a building(s), storey(s), and a space(s). A zone can be adjacent to another zone or even contain other zones. It can also be bounded

[1] `https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2_TC1/OWL/index.html`

[2] https://w3c-lbd-cg.github.io/bot/

by physical building elements or even contain them. Building elements can also host other elements (a wall hosting a sensor). Table 3.1 summarizes the classes adopted from BOT while Figure 3.2 provides an intuitive description of how BOT is used to describe a site, `<UNM>`, having a building, `<Block_B>`, containing a storey, `<Floor_3>` with a certain space, `<Office_204>`. The respective entity connections are made via the object properties; `bot:hasBuilding`, `bot:hasStorey` and `bot:hasSpace`. Explicitly asserted relationships/properties are shown by the solid line arrows, while those that are automatically inferred are shown by the dotted line arrows. The back-end inference rules at play here are defined in BOT via the ranges and domains summarised in Table 3.1. A corresponding machine-readable serialisation (Turtle format) of the data model (BIM-KG) is shown in Listing 3.1. With a BOT foundation, semantic extensions can be made to describe low-level details about specific features of interest contained in the space, `<Office_204>`, such as sensors and walls, via another object property, `bot:containsElement` as the extension point.

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix bot: <https://w3id.org/bot#> .
```

Table 3.1: BOT classes and properties to be adopted for this work's data modelling example

Classes (domain)	Properties	Classes(range)
bot:Zone	bot:containsZone	bot:Zone
	bot:adjacentZone	bot:Zone
bot:Site	bot:hasBuilding	bot:Building
bot:Building	bot:hasStorey	bot:Storey
bot:Storey	bot:hasSpace	bot:Space
bot:Space	bot:containsElement	bot:Element
bot:Element	bot:hostsElement	bot:Element

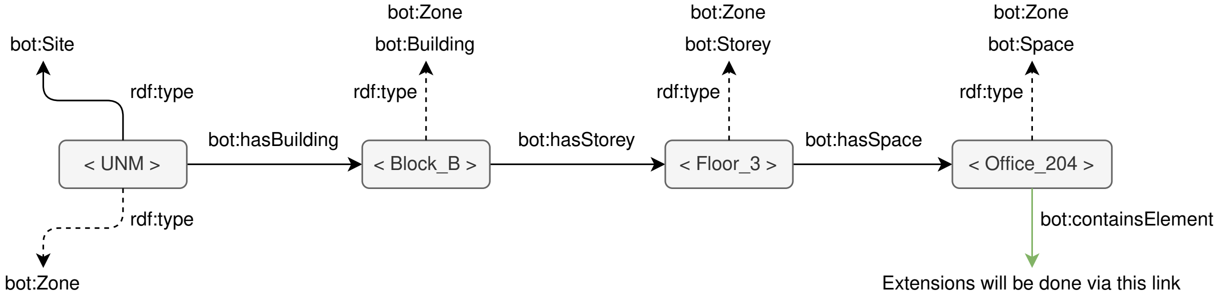


Figure 3.2: Using BOT classes and properties to describe the high-level semantic topological details of a building

```

# Site location (UNM) for some building of interest (Block_B)
<UNM> a bot:Site ;
  bot:hasBuilding <Block_B> .

# Block_B has some storey of interest Floor_3
<Block_B> bot:hasStorey <Floor_3> .

# Floor_3 has some space(zone) of interest Office_204
<Floor_3> bot:hasSpace <Office_204> .
```

Listing 3.1: Turtle serialization of the information modelled in Figure 3.2 above

2. **Competency Question 2 (CQ2)**: *How to semantically describe a feature of interest within an indoor space while encoding its measurable properties, corresponding property values and property states in a way that allows tracking changes, deletions and revisions?*

   For this question, the Semantic Sensor Network (SSN)[3] (Haller et al., 2017) ontology is adopted. At its core exists a lightweight but self-contained ontology, Sensor, Observation, Sampling and Actuator (SOSA), encapsulating elementary classes necessary for the semantic description of features of interest and their properties, sensor observations, and feature sampling procedures to describe tractable sensor observations and actuation behaviour. Furthermore, the Smart Energy Aware Systems (SEAS) (Lefrançois et al., 2016) ontology is used to avail semantic extensions to SSN. SEAS is an ecosystem of modules that, together, provide, semantic vocabulary to describe physical systems and their interrelations. Among these is the `seas:FeatureOfInterestOntology`[4] for describing features of interest and their properties, and the `seas:EvaluationOntology`[5] for describing evaluations of these properties. However, these natively have no semantics to encode property states in a way that can be tracked over time. For this, the Ontology for Property Management (OPM)[6] (Rasmussen et al., 2018) is deemed relevant. The specific classes and properties it provides for this work are summarised in Table 3.3.

[3] https://www.w3.org/TR/vocab-ssn/

[4] https://w3id.org/seas/FeatureOfInterestOntology

[5] https://w3id.org/seas/EvaluationOntology

[6] https://w3c-lbd-cg.github.io/opm/

To illustrate the usage of these ontologies to answer competency question 2, a direct semantic extension is made to the bot:Space, <Office_204>. First, it is defined as a sosa:FeatureOfInterest having two measurable properties; <Office_204#temperature> and <Office_204#humidity>, both defined via the relation ssn:hasProperty. To stay compliant with OPM and satisfy competency question 2, both properties are required to have at least one opm:hasPropertyState relation for assigning states to the properties. A property state in OPM is an evaluation that contains the value and metadata of a property deemed true at a specific point in time. OPM also specifies that, as a minimum, a property state should have a value and preferably, a generation time, an assignment that can respectively be done via the properties; schema:value, from schema.org[7] and prov:generatedAtTime, from the Provenance Ontology[8]. <Office_204> is further defined to have two elements, which are both walls. One wall, <Office_204/east>, is located on the eastern side of the room, and the other wall, <Office_204/south>, is on the southern side. Each of them is described as both a sosa:FeatureOfInterest and a sosa:Platform. The latter simply means that each of the walls hosts another entity, in this case, a <NodeMCU> Printed Circuit Board (PCB) that is also a sosa:Platform hosting a DHT22 temperature and humidity sensor. Because the temperature and humidity properties are defined on the <Office_204> entity, but are implicitly being measured from the interior walls <Office_204/east> and <Office_204/south> via the embedded sensors, it is necessary to describe each wall as a sosa:Sample[9]. A more intuitive graphical description of this data modelling process is provided in Figure 3.3 together with a Turtle serialisation in Listing 3.2. Another ontology that can be adopted for the explicit definition of complex functionality of smart appliances and their controllability is Smart Appliances REFerence (SAREF)[10] (Daniele et al., 2015). The starting point of the SAREF vocabulary is a device. Currently, much of the semantic vocabulary for describing device controllability has been availed by the SEAS, SSN SOSA and BOT however, should the need arise for more explicit descriptions of systems and their energy consumption

[7] https://schema.org/value

[8] https://www.w3.org/TR/prov-o/

[9] https://www.w3.org/TR/vocab-ssn/#SOSASample

[10] https://saref.etsi.org/

behaviour, SAREF extensions can be adopted.

Table 3.2: SEAS classes and properties adopted for this work's data modelling example

Classes	Properties
seas:ElectricPowerSystem	seas:isPoweredBy
seas:TemperatureEvaluation	seas:optimizes
seas:AgentComfortEvaluation	seas:thermalTransmittance
seas:MaximumComfortableEvaluation	seas:relativeToAgent
seas:MinimumComfortableEvaluation	seas:evaluatedSimpleValue
seas:Battery	seas:hasTemporalContext

Table 3.3: OPM classes and properties adopted for this work's data modelling example

Classes	Properties
opm:Assumed	opm:hasPropertyState
opm:CurrentPropertyState	
opm:PropertyState	
opm:Confirmed	
opm:OutdatedPropertyState	
opm:Deleted	

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix bot: <https://w3id.org/bot#> .
@prefix xsd:  <http://www.w3.org/2001/XMLSchema#> .
@prefix cdt:   <http://w3id.org/lindt/custom_datatypes#> .
@prefix schema: <http://schema.org/>.
@prefix sosa: <http://www.w3.org/ns/sosa/> .
@prefix ssn: <http://www.w3.org/ns/ssn/> .
@prefix seas: <https://w3id.org/seas/> .
@prefix opm: <https://w3id.org/opm#> .
@prefix prov:  <http://www.w3.org/ns/prov#> .

# Office_204 (FOI) hosts some 2 walls at the east and south that will host
# some sensors. The space also has two properties temperature and humidity
<Office_204> a sosa:FeatureOfInterest ;
  bot:containsElement <Office_204/east>, <Office_204/south> ;
  ssn:hasProperty <Office_204#temperature> , <Office_204#humidity> .

```

```
# Office_204 east side wall to host a NodeMCU board with
# a DHT22 temp and  hum sensor.
<Office_204/east> a sosa:FeatureOfInterest , sosa:Sample , sosa:Platform ;
  sosa:hosts <NodeMCU_1> .

# Office_204 south side wall to host a NodeMCU board with a DHT22 temp and
# hum sensor.
<Office_204/south> a sosa:FeatureOfInterest , sosa:Sample , sosa:Platform ;
  sosa:hosts <NodeMCU_2> ;

# DESCRIPTION OF PCB BOARDS HOSTING THE SENSORS
###############################################################

# NodeMCU 1 board hosted by the office_204 east side wall.
<NodeMCU_1> a ssn:System , sosa:Platform ;
  sosa:hosts <DHT22/01> ;
  ssn:hasSubSystem <DHT22/01> .

# NodeMCU 2 board hosted by the office_204 south side wall.
<NodeMCU_2> a ssn:System , sosa:Platform ;
  sosa:hosts <DHT22/02> ;
  ssn:hasSubSystem <DHT22/02> .

# Assigning a state to the temperature property of Office #204
<Office_204#temperature>
  opm:hasPropertyState <Office_204#temperature_state_48906948_er8t78> .

# Assigning semantics to Office_204#temperature_state_48906948_er8t78 state
<Office_204#temperature_state_48906948_er8t78>
  a opm:Confirmed ,
    opm:CurrentPropertyState ;
  schema:value "30.5 Cel"^^cdt:temperature ;
  prov:generatedAtTime "2020-07-28T16:41:17.711+02:00"^^xsd:dateTime.

# Assigning a state to the humidity property of Office #204
<Office_204#humidity>
  opm:hasPropertyState <Office_204#humidity_state_40039548_gktiy8> .
```

```

# Assigning semantics to Office_204#humidity_state_40039548_gktiy8 state
<Office_204#humidity_state_40039548_gktiy8>
  a opm:Confirmed ,
    opm:CurrentPropertyState ;
  schema:value "85.0 %"^^cdt:ucum ;
  prov:generatedAtTime "2020-07-28T16:41:17.711+02:00"^^xsd:dateTime.
```

Listing 3.2: Turtle serialization of the information modelled in Figure 3.3.

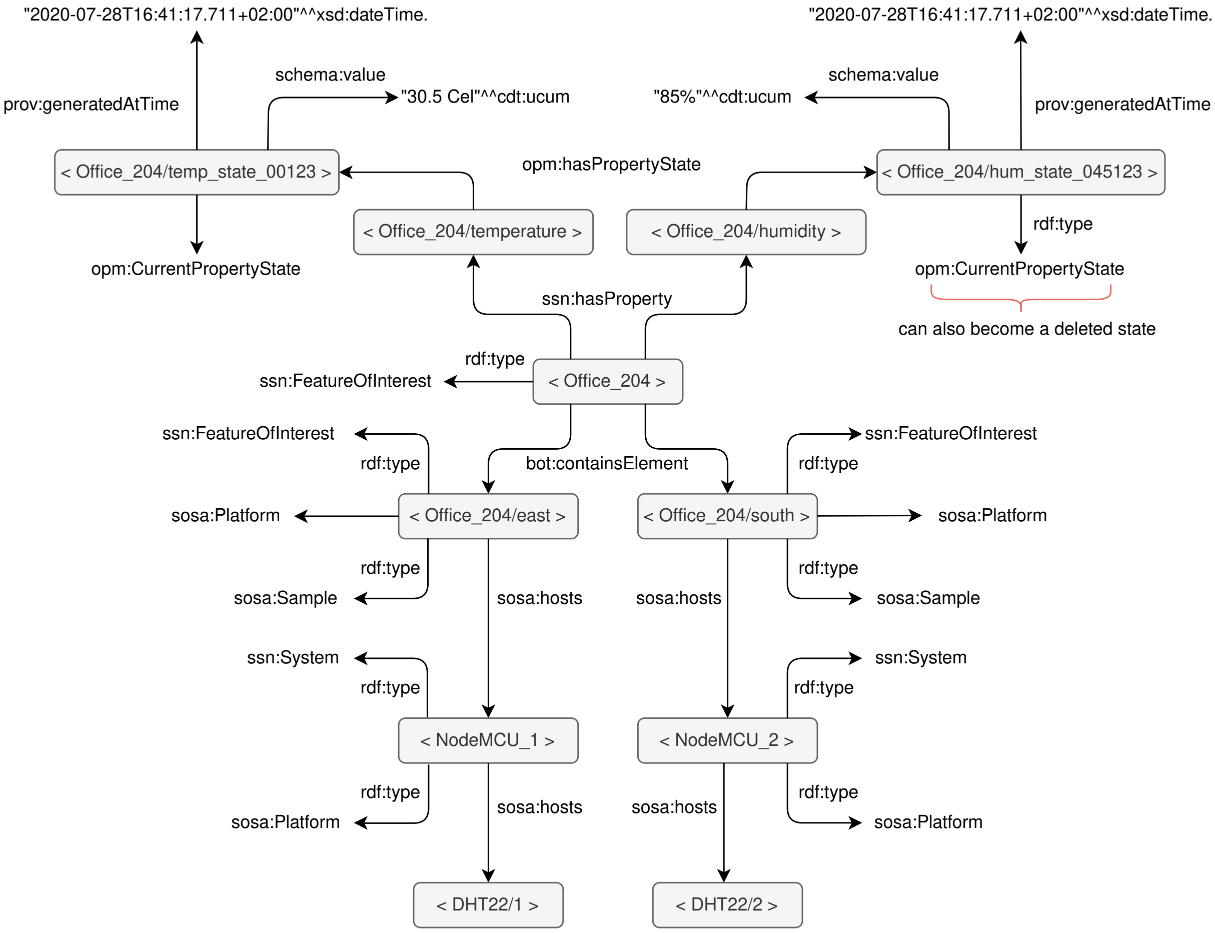


Figure 3.3: Using SSN/SOSA, OPM and SEAS ontologies to extend the semantic details of a building to encapsulate indoor zone information about sensors, elements contained within, their properties and state management (simplistic view).

Below is a summary of the modular ontologies that have been adopted to solve the competency questions above;

1. **Building Topology Ontology (BOT)**[11]: BOT (Rasmussen et al., 2017a) provides the foundation vocabulary necessary to model the core topological aspects of a building

[11] https://w3c-lbd-cg.github.io/bot/

such as a site, a building, a storey, a space, and an element in a space. BOT is chosen for simplicity, modularity and extensibility reasons.

2. **Semantic Sensor Network (SSN)**[12] **and Sensor, Observation, Sampling and Actuator (SOSA)**: SSN and SOSA (Haller et al., 2017) offer the vocabulary necessary to describe sensors and their observations.

3. **SEAS**: SEAS (Lefrançois et al., 2016) offers an ontology to describe smart appliances and their communication with the grid.

4. **OPM**: OPM (Rasmussen et al., 2018) provides a schema for describing temporal properties that are subject to changes as the building design evolves.

5. **SAREF (Smart Appliances REFerence)**: The SAREF (Daniele et al., 2015) suite of ontologies is a set of standardised frameworks designed to ensure that different IoT solutions from various providers can work together seamlessly.

3.1.2 LBD Model Validation for Downstream KRL

The continued standardisation of Semantic Web approaches in the AEC/FM industry has resulted in an unprecedented volume of building data being included on the web as Linked Data. Although gathering and publishing such massive amounts of data is certainly a step in the right direction for the industry, the effectiveness of this data hinges on its quality. In other words, simply having access to an extensive web of building data does not guarantee valuable insights or improved decision-making. Within the context of KRL, the potency of a model is tightly bound to the quality of the input data (knowledge graph). Factors such as data integrity, consistency, density, noise level, completeness, redundancy, and structural regularities in a knowledge graph can significantly influence the quality of what the model learns (the resultant embeddings) (Zaveri et al., 2016). In the case of FM, this can arguably affect a building's operational behaviour. From a data engineering perspective, data quality is often defined as fitness for use within a certain domain, use case, or application (Juran et al., 1979; Wang and Strong, 1996; Knight and Burn, 2005). It is important to note that even datasets with quality issues can hold value if they meet the standards required for particular applications.

[12] https://www.w3.org/TR/vocab-ssn/

For instance, the web contains content of varying quality from DBPedia[13] (Zaveri et al., 2013) but is still widely regarded as highly useful. In the AEC/FM industry, most data-intensive BIM applications rely on some form of data-fitness guidelines (ISO 29481-1, 2016; ISO 16739:2024, 2024) that specify the fundamental requirements data must satisfy to be deemed useful for a specific use case. However, there is little consensus on what constitutes effective data-fitness guidelines in the industry, which hampers the development of comprehensive tool-based compliance checkers. These tools are crucial for reviewing large volumes of data points (triples) in large enterprise BIM-KGs. To be effective, this thesis argues that the set of rules in each data-fitness guideline has to be small in the beginning. Such a small set, of course, cannot be all-encompassing, but it can give industry stakeholders a foothold to achieve measurable effects on data reliability and verifiability. In this research, focusing on structural consistency, data completeness, and redundancy serves as a pragmatic starting point to tackle the most immediate threats to data reliability and minimise cascading errors further along the KRL pipeline. *Structural consistency* guarantees that the BIM-KG adheres to a predefined schema and semantic constraints, ensuring that the underlying representation is reliable for parsing, processing and analysis. *Data completeness* verifies that all essential information for a specific use case is present. *Redundancy checking* helps prevent duplication and conflicting data, which can otherwise obscure important insights or lead to erroneous conclusions. Although these three issues do not cover every possible aspect of data quality, they lay a foundation for the development of more nuanced or domain-specific data-fitness guidelines in the industry.

This research uses SHACL[14] to illustrate the procedures for validating a BIM-KG against specific quality criteria. Simultaneously, SPARQL[15] is employed for its data extraction and manipulation capabilities, facilitating cleanup and transformation tasks on the BIM-KG whenever issues are identified. It is important to note that SPARQL can also be used for validation as a result of its high expressivity. Before detailing the validation process, a brief narrative of the chosen BIM-KG issues to check for is provided below.

1. **Structural Consistency**: Because KRL relies on message-passing, any structural irregularities in the BIM-KG can disrupt the learning process and impact the quality of the resulting embeddings. Irregularities like the one shown in Figure 3.4 can

[13] `https://www.dbpedia.org/resources/ontology/`

[14] https://www.w3.org/TR/shacl/

[15] https://www.w3.org/TR/sparql11-query/

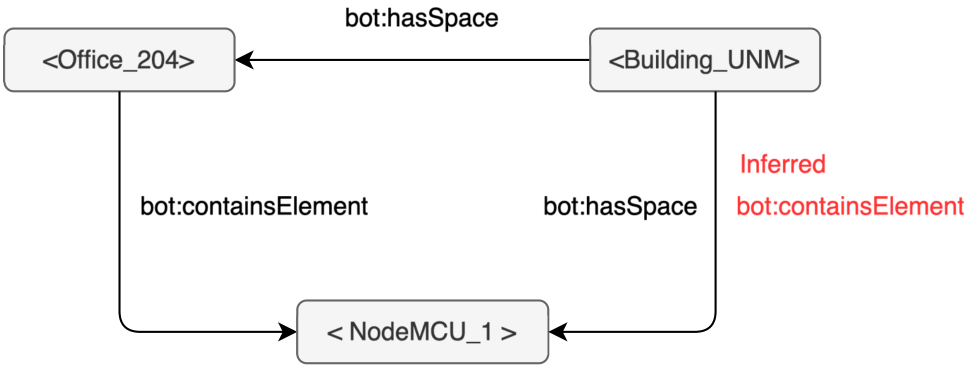


Figure 3.4: Example of inconsistent knowledge in a BIM-KG

arise from inconsistent or incomplete data entry in the underlying BIM model. For example, if a BIM-KG contains the following triples: `<Office_204>` $\xrightarrow{\text{bot:containsElement}}$ `<NodeMCU_1>`, `<Building_UNM>` $\xrightarrow{\text{bot:hasSpace}}$ `<Office_204>`, `<Building_UNM>` $\xrightarrow{\text{bot:hasSpace}}$ `<NodeMCU_1>`, the inferred knowledge ‘`<Building_UNM>` $\xrightarrow{\text{bot:containsElement}}$ `<NodeMCU_1>`’ contradicts the logical expression of the third triple. Using SHACL, specific constraints can be defined to dictate the acceptable configurations for nodes and relationships in the BIM-KG (Werbrouck et al., 2019; Stolk and McGlinn, 2020; Guo et al., 2021; Pauwels et al., 2024). In addition, SPARQL can be leveraged to query the BIM-KG and identify any deviations from the enforced SHACL shapes and corrective actions are taken using SPARQL’s Create, Read, Update and Delete (CRUD) operations (Yurchyshyna et al., 2007; Yurchyshyna and Zarli, 2009; Zhang et al., 2017).

2. **Data Completeness**: In the context of this thesis, data completeness refers to how well a BIM-KG covers the relevant domain knowledge, i.e., the proportion of existing data to the total data required for a specific problem. Data completeness is a multi-faceted problem encompassing several dimensions such as accuracy, timeliness, relevancy, objectivity, and believability. What constitutes ‘complete’ data can vary depending on the application or use case, meaning that different situations may have different requirements for what makes data complete. For an indoor environment control task, the instance `<Office_204>` might suffer from a data completeness problem if its measurable properties, `<Office_204#temperature>` and `<Office_204#humidity>`, are absent in the BIM-KG, whereas this would not be of concern for a cost estimation use case. When a KRL model is trained on such incomplete data, the resultant embeddings miss out on potentially important contextual information, leading to

inaccurate downstream inferences and analyses. Zaveri et al. (2016) reviewed several quality assessment approaches for Linked Data and categorised knowledge graph completeness into four classes: schema completeness, property completeness, population completeness and inter-linking completeness. This thesis does not go into the details of these classes; however, Zaveri et al. (2016)'s work is a starting point for other researchers in the AEC/FM domain to investigate data completeness issues that are specific to BIM-KGs.

3. **Redundancy**: Redundancy occurs when multiple nodes in a BIM-KG hold the same type of information but are labelled with different names or identifiers. A hypothetical example with intentional redundancy would be a particular space, say `<Office_204>`, being represented by two different properties in the same dataset, such as `http://example.org/spaceID` and `http://example.org/name`. This redundancy (`spaceID` and `name`) can ideally be solved by merging the two properties and keeping only one unique identifier. Property or entity duplication in a BIM-KG can misguide a KRL model's message-passing process, resulting in malformed embeddings.

To illustrate the validation process for the BIM-KG discussed in Section 3.1, the requirements below are utilized. It is important to note that these requirements are typically defined by a domain expert for a specific use case.

1. Every `sosa:FeatureOfInterest` has to host at least one `ssn:System`.

2. Every `ssn:System` has to host at least one `sosa:Sensor`.

3. Every `ssn:Property` has to have an associated `opm:PropertyState`.

4. Every `opm:CurrentPropertyState` has a defined value (`schema:value`) and a timestamp (`prov:generatedAtTime`).

To check these conditions, validations that are rooted in SPARQL and SHACL are adopted. First, SPARQL `ASK` queries are provided for each condition, which return either true if the condition is violated, or false if the data meets the condition (see Listing 3.3 - Listing 3.6):

1. `sosa:FeatureOfInterest` hosts at least one `ssn:System`:

```
PREFIX ssn: <http://www.w3.org/ns/ssn/>.
PREFIX sosa: <http://www.w3.org/ns/sosa/>.

ASK WHERE {
  ?foi a sosa:FeatureOfInterest .
  FILTER NOT EXISTS {
      ?foi sosa:hosts ?system .
      ?system a ssn:System .
  }
}
```

Listing 3.3: This query returns true if there exists at least one sosa:FeatureOfInterest that does not host any ssn:System.

2. Every ssn:System hosts at least one sensor:

```
PREFIX ssn: <http://www.w3.org/ns/ssn/>.
PREFIX sosa: <http://www.w3.org/ns/sosa/>.

ASK WHERE {
  ?system a ssn:System .
  FILTER NOT EXISTS {
    ?system sosa:hosts ?sensor
    ?sensor a sosa:Sensor
  }
}
```

Listing 3.4: This query returns true if there exists at least one ssn:System that does not host any sosa:sensor

3. Every ssn:Property has an associated state:

```
PREFIX ssn: <http://www.w3.org/ns/ssn/>.
PREFIX sosa: <http://www.w3.org/ns/sosa/>.
PREFIX opm: <https://w3id.org/opm#>.

ASK WHERE {
  ?property a ssn:Property .
  FILTER NOT EXISTS {
    ?property opm:hasPropertyState ?state
  }
}
```

Listing 3.5: This query returns `true` if there exists at least one `ssn:Property` that has no property state

4. Every `opm:CurrentPropertyState` has a defined value and a timestamp:

```
PREFIX opm: <https://w3id.org/opm#>.
PREFIX schema: <http://schema.org/>.
PREFIX prov:  <http://www.w3.org/ns/prov#>.

ASK WHERE {
  ?state a opm:CurrentPropertyState .
  FILTER NOT EXISTS { ?state schema:value ?value }
  FILTER NOT EXISTS { ?state prov:generatedAtTime ?time }
}
```

Listing 3.6: This query returns `true` if there exists at least one `opm:CurrentPropertyState` that does not have either a `schema:value` or `prov:generatedAtTime` property.

SHACL is more expressive and capable of defining more complex constraints than SPARQL alone, as shown in Listing 3.7 below.

```
@prefix sh: <http://www.w3.org/ns/shacl#> .
@prefix xsd:  <http://www.w3.org/2001/XMLSchema#> .
@prefix schema: <http://schema.org/>.
@prefix sosa: <http://www.w3.org/ns/sosa/> .
@prefix ssn: <http://www.w3.org/ns/ssn/> .
@prefix opm: <https://w3id.org/opm#> .

# Shape for FeatureOfInterest
:FeatureOfInterestShape
    a sh:NodeShape ;
    sh:targetClass sosa:FeatureOfInterest ;
    sh:property [
        sh:path sosa:hosts ;
        sh:class ssn:System ;
        sh:minCount 1 ;
        sh:message "A FeatureOfInterest must host at least one System."
    ] .

# Shape for System
:SystemShape
    a sh:NodeShape ;
    sh:targetClass ssn:System ;
    sh:property [
        sh:path sosa:hosts ;
        sh:minCount 1 ;
        sh:message "A System must host at least one sensor."
    ] .

# Shape for Property
:PropertyShape
    a sh:NodeShape ;
    sh:targetClass ssn:Property ;
    sh:property [
        sh:path opm:hasPropertyState ;
        sh:minCount 1 ;
        sh:message "A Property must have an associated state."
    ] .

# Shape for CurrentPropertyState
:CurrentPropertyStateShape
    a sh:NodeShape ;
    sh:targetClass opm:CurrentPropertyState ;
    sh:property [
        sh:path schema:value ;
        sh:minCount 1 ;
        sh:message "A CurrentPropertyState must have a defined value."
    ] ;
    sh:property [
        sh:path prov:generatedAtTime ;
        sh:datatype xsd:dateTime ;
        sh:minCount 1 ;
        sh:message "A CurrentPropertyState must have a timestamp."
    ] .
```

Listing 3.7: SHACL shapes to express the same constraints defined in Listings 3.3 - 3.6

In the above SHACL shapes graph, each `sh:NodeShape` defines the shape for a specific class of nodes, e.g., `sosa:FeatureOfInterest`, `ssn:System`, `ssn:Property`, and `opm:CurrentPropertyState`. For each shape, `sh:property` is used to specify constraints for the properties of the nodes that conform to the shape. For instance, in `:FeatureOfInterestShape`, the `sh:property` construct requires that each `sosa:FeatureOfInterest` must host (`sosa:hosts`) at least one (`sh:minCount 1`) `ssn:System`. Similarly, `:SystemShape` specifies that each `ssn:System` must host at least one sensor. The `sh:message` constructs are used to provide human-readable error messages that are displayed when a constraint is violated. In practice, SHACL validation is performed using pySHACL[16], a Python library developed by `RDFlib`[17]. The high-level implementation details for this are shown in Listing 3.8.

```python
import rdflib
from pyshacl import validate

# Load RDF Data
data_graph = rdflib.Graph()
data_graph.parse("path_to_the_kg_data", format='turtle')

# Load SHACL Shapes
shapes_graph = rdflib.Graph()
shapes_graph.parse("path_to_your_shacl_shapes", format='turtle')

# Run the validation
val = validate(data_graph, shacl_graph=shapes_graph)
conforms, results_graph, results_text = val

# Check if the data passed the SHACL validation
if conforms:
    print("The data graph passed SHACL validation!")
```

[16] https://github.com/RDFLib/pySHACL

[17] https://github.com/RDFLib/rdflib

```
else:
    print("The data graph failed SHACL validation.")
    print(results_text)
```

Listing 3.8: Python script for loading RDF data and SHACL shapes for validating a knowledge graph

3.1.3 Conclusion

This section provided a prototypical example for building a BIM-KG for training KRL models. Several existing AEC/FM vocabularies (ontologies) were used instead of introducing new ones—a workflow that is strongly encouraged because it promotes interoperability, leverages established domain expertise, and prevents unnecessary fragmentation in data standards. Two approaches were presented to maintain the focus of the BIM-KG's scope to the task at hand: crafting competency questions as a proactive approach and using `SHACL` and SPARQL for BIM-KG validation as a reactive approach. To ensure that the modelled BIM-KG is sufficient in the KRL phase, a pragmatic starting point is taken by considering three issues as important to check for: structural consistency, data completeness and redundancy. Arguably, these tackle the most immediate threats to data reliability and minimise cascading errors further along the KRL pipeline while laying a strong foundation for other researchers to develop more nuanced or domain-specific data-fitness guidelines in the industry.

3.2 Performance Analysis of Knowledge Representation Learning (KRL) on Linked Building Data (LBD)

The section develops and formalises a methodology for applying KRL to BIM-KGs using performance analysis experiments. An overview of the models, datasets, and evaluation protocol used for the experiments is discussed herein.

3.2.1 Models

Several KRL models have been introduced in Section 2.8, differentiated primarily by their score functions. For this research's experiments, five KRL models were chosen: ComplEx, DistMult,

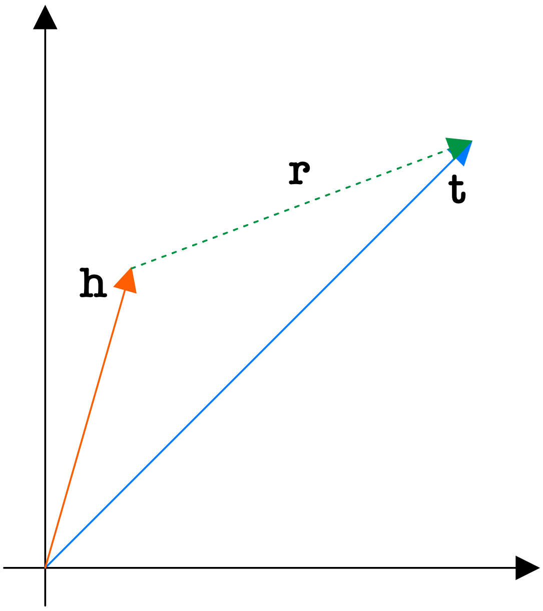


Figure 3.5: A simple illustration of TransE

RotatE, TransE and TransH. In this work, all models consume triples sampled from the 2 publicly available BIM-KGs datasets[18] described in Subsection 3.2.2. The internal structure and outputs of each model are discussed below.

1. **TransE** (Bordes et al., 2013) is one of the earliest and most influential KRL models for link prediction. It embeds both entities and relations from a knowledge graph into a low-dimensional vector space, typically $\mathbb{R}^d$, and represents relationships as translation vectors. As illustrated in Figure 3.5, the core intuition is that for a triple (h, r, t), where h is the *head* entity, r is the *relation*, and t is the *tail* entity, the embedding of h plus the embedding of r should be close to the embedding of t (Equation 3.1):

$$\mathbf{h} + \mathbf{r} \approx \mathbf{t} \tag{3.1}$$

where $\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$ are the entity embeddings, and $\mathbf{r} \in \mathbb{R}^d$ is the relation embedding. During training, each triple is presented along with a negatively corrupted counterpart (h', r, t') (where the head or the tail entity is replaced with a random entity as described in Subsection 3.2.3) to teach the model how to distinguish correct (positive) facts from incorrect (negative) ones. TransE initialises a real-valued vector $\mathbf{h} \in \mathbb{R}^d$ for each

[18] `https://github.com/BIM-and-Automation-Laboratory/phd-source/tree/main/hpo-study/datasets`

unique entity h in the BIM-KG. For instance, entities `RoomA`, `Temperature`, and `Door1` each have their own d-dimensional embedding, whose vectors are updated during training to capture the semantic relationships inherent in the BIM-KG. Each relation r (e.g., `hasProperty`, `isConnectedTo`) is also represented by a vector $\mathbf{r} \in \mathbb{R}^d$. In the translational framework, relations shift the head embedding vector towards the tail embedding vector, in other words, if you take the embedding of the head entity (**h**) and add the relation's vector (**r**) to it, you should end up close to the tail entity's embedding (**t**). To measure how well a triple (h, r, t) fits the translation criterion, the scoring function below is used: This work adopts the Euclidean distance between $\mathbf{h} + \mathbf{r}$ and **t**.

$$f(h, r, t) = ||\mathbf{h} + \mathbf{r} - \mathbf{t}||_2 \tag{3.2}$$

A *lower* score indicates a better fit (a more plausible triple). For positive training triples, the objective is to make $f(h, r, t)$ *small*, whereas for negative/corrupted triples, the objective is to push $f(h', r, t')$ *higher*. During training, TransE typically uses a Margin Ranking Loss (MRL) to teach the model which triples are correct and which are incorrect. For each true triple (h, r, t) and its corrupted counterpart (h', r, t'), the MRL is expressed as:

$$\mathcal{L} = \sum_{\substack{(h,r,t)\in\mathcal{P} \\ (h',r,t')\in\mathcal{C}}} \max\Big(0,\ \gamma\ +\ f\big(h', r, t'\big)\ -\ f\big(h, r, t\big)\Big) \tag{3.3}$$

where:

- $\mathcal{P}$ is the set of positive (true) triples (sampled from the BIM-KG).
- $\mathcal{C}$ is the set of negative (corrupted) triples.
- $f(\cdot)$ is the scoring function defined in Equation 3.2.
- γ is a margin hyperparameter enforcing separation between positive and negative triples.

After training, each entity and relation in the BIM-KG has a learned embedding in $\mathbb{R}^d$. This is typically a *lookup table* of d-dimensional vectors for all entities and relations. Together with the scoring function $f(h, r, t)$ in Equation 3.2, link prediction tasks can be carried out. For instance, if you have a partial triple `RoomA` $\xrightarrow{\text{hasProperty}}$?, TransE can rank all possible tail entities by computing ||`RoomA + hasProperty - candidate` ∈{`Temperature, LightLevel`}|| and choosing the closest match, `Temperature`. Another concrete example in practice is building automation stakeholders using TransE to evaluate the plausibility scores of various hypotheses, potentially aiding in *context-aware decision making*. TransE is computationally efficient and relatively easy to implement, making it a popular choice for many KRL tasks. Although it has limitations in dealing with 1-to-N, N-to-1, and N-to-N relations, limiting the ability to capture the complexity and heterogeneity of relationships in BIM-KGs, its behaviour is investigated nonetheless. Again, the goal of these experiments is not to find the best KRL model configuration but to expose and better understand the idiosyncrasies arising from integrating different KRL models with BIM-KGs.

2. **TransH** (Wang et al., 2014) extends TransE by allowing each relation to define its own hyperplane in the embedding space as shown in Figure 3.6. While TransE represents every relation as a single vector $\mathbf{r}$ in a shared space, TransH introduces two main components for each relation r: a normal vector $\mathbf{w}_r$ that determines the orientation of its hyperplane, and a translation vector $\mathbf{d}_r$ that represents how relation $\mathbf{r}$ translates head entities to tail entities once they are projected onto the hyperplane. This modification addresses the issue of entities participating in multiple, semantically distinct relations (for example, a room that might be adjacent to another room (via the relation `isAdjacentTo`) and has a property such as `Temperature`) (via another relation `hasProperty`). By projecting entities onto a relation-specific hyperplane, TransH provides a more flexible, context-sensitive representation. Each entity h or t is assigned an embedding vector in $\mathbb{R}^d$ while for each relation r, the model maintains a normal vector $\mathbf{w}_r$ and a translation vector $\mathbf{d}_r$. Before applying a translation, TransH projects an entity embedding $\mathbf{h}$ onto the hyperplane defined by $\mathbf{w}_r$:

$$\mathbf{h}_{\perp} = \mathbf{h} \; - \; (\mathbf{w}_r^{\top}\mathbf{h})\, \mathbf{w}_r, \tag{3.4}$$

and similarly for the tail embedding $\mathbf{t}$.

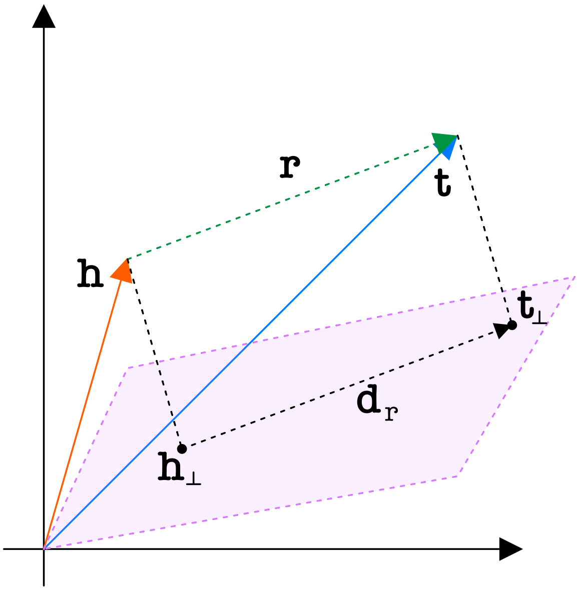


Figure 3.6: A simple illustration of TransH

$$\mathbf{t}_{\perp} = \mathbf{t} - \left(\mathbf{w}_r^{\top}\mathbf{t}\right)\mathbf{w}_r \tag{3.5}$$

The translation itself then takes place in this hyperplane:

$$\mathbf{h}_{\perp} + \mathbf{d}_r \approx \mathbf{t}_{\perp}. \tag{3.6}$$

This design allows different relations to transform entities in relation-specific ways, thus capturing more nuanced contextual behaviour than TransE. During training, TransH typically also applies a MRL approach similar to TransE. However, in this case the scoring function $f(h, r, t)$ is the distance between $\mathbf{h}_{\perp} + \mathbf{d}_r$ and $\mathbf{t}_{\perp}$ (this work adopts the Euclidean distance or L_2 norm),

$$f(h, r, t) = \left\|\left(\mathbf{h}_{\perp} + \mathbf{d}_r\right) - \mathbf{t}_{\perp}\right\|_2, \tag{3.7}$$

A lower value of $f(h, r, t)$ indicates a more plausible triple, since it implies that $\mathbf{h}_{\perp} + \mathbf{d}_r$ is geometrically close to $\mathbf{t}_{\perp}$ in the embedding space. By the end of training, TransH provides each entity in the BIM-KG with a learned embedding, along with a normal vector and translation vector for each relation. These learned parameters enable a variety of link prediction tasks. For example, given `RoomA` $\xrightarrow{\text{isAdjacentTo}}$?, the model can project `RoomA` onto the hyperplane for `isAdjacentTo`, apply the translation $\mathbf{d}_{\texttt{isAdjacentTo}}$, and measure distances to all other entity embeddings to identify plausible neighbours. The main advantage of TransH over TransE in a BIM-KG setting is its capacity to handle varied semantics for the same entity depending on which relation it is involved in. Such flexibility is especially valuable if a single entity (such as a specific room or sensor) participates in multiple relationships that do not necessarily share the same geometric properties. However, it should be noted that TransH introduces additional parameters in the form of relation-specific normal vectors, which increases both the model's expressiveness and its computational overhead. As a result, practitioners must weigh these factors against the potential gains in predictive accuracy and the need to handle more complex relation-specific behaviour within BIM-KGs.

3. **RotatE** (Sun et al., 2019) represents entities and relations using complex-valued embeddings, where each dimension is treated as a point in the complex plane. Concretely, for a triple (h, r, t), the head and tail entities h and t have embeddings $\mathbf{h}, \mathbf{t} \in \mathbb{C}^d$, while the relation r is associated with a vector $\mathbf{r} \in \mathbb{C}^d$. As illustrated in Figure 3.7a, the key idea is to interpret each relation as a rotation in the complex plane, so that $\mathbf{t}$ is obtained by rotating $\mathbf{h}$ according to $\mathbf{r}$. In practice, this is realised via element-wise (Hadamard) multiplication:

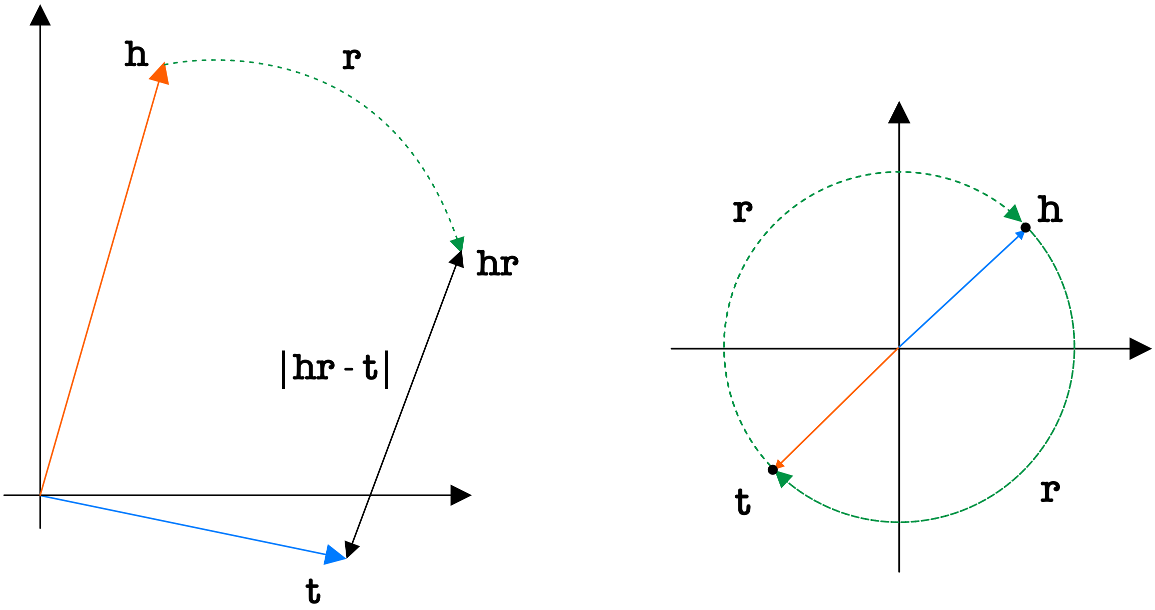
$$\mathbf{t} \approx \mathbf{h} \odot \mathbf{r}, \tag{3.8}$$

where every element of $\mathbf{r}$ is constrained to have absolute value 1. This constraint makes $\mathbf{r}$ a pure phase vector that *rotates* $\mathbf{h}$ within each dimension of $\mathbb{C}^d$. By formulating relations as rotations, RotatE can naturally capture a variety of relational patterns, including symmetry, antisymmetry, and inversion. For instance, a symmetric relation like `isAdjacentTo` would imply that rotating $\mathbf{h}$ by $\mathbf{r}$ to arrive at $\mathbf{t}$ also means rotating $\mathbf{t}$ by the same $\mathbf{r}$ recovers $\mathbf{h}$. This is illustrated in Figure 3.7b. An antisymmetric relation would instead prohibit such a bidirectional mapping, as the same rotation cannot be applied in reverse without changing the embedding position. In the context of a BIM-KG, RotatE's capacity to capture both symmetrical and asymmetrical relationships is particularly useful for encoding connections such as `isConnectedTo`, `isPartOf`, or even hierarchical dependencies among building components (e.g., floors, rooms, and sensors). Since BIM-KGs often mix functional, spatial, and compositional relationships, the flexibility offered by complex embeddings can lead to more accurate link predictions. Nonetheless, these potential benefits come at the expense of managing complex-valued vectors and additional hyperparameter tuning.

4. **DistMult** (Yang et al., 2014) is a bilinear embedding model that a simplification of RESCAL (Nickel, 2013). In this model, both entities and relations are mapped to vectors $\mathbf{e}_i, \mathbf{r}_j \in \mathbb{R}^d$. Given a triple (h, r, t), DistMult employs a scoring function based on element-wise (Hadamard) multiplication:

$$f(h, r, t) = \mathbf{h}^\top (\operatorname{diag} \mathbf{r})\, \mathbf{t} = \sum_{k=1}^{d} h_k\, r_k\, t_k, \tag{3.9}$$

where $\mathbf{h}, \mathbf{t}, \mathbf{r}$ are the embeddings for the head entity h, tail entity t, and relation r, respectively, and $\operatorname{diag} \mathbf{r}$ denotes a diagonal matrix with $\mathbf{r}$ on its diagonal. Intuitively, DistMult captures relational influence by scaling each dimension of $\mathbf{h}$ by the corresponding dimension in $\mathbf{r}$, which is then combined with $\mathbf{t}$. A notable property of DistMult is that it is inherently *symmetric* with respect to relations. That is,



(a) A simple illustration of RotatE modeling **r** as a rotation in complex space

(b) An illustration of RotatE modeling symmetric relations **r**

Figure 3.7: Simple illustration of RotatE's mechanics

reversing the roles of head and tail does not change the score, since the scoring function is commutative. This can be beneficial for knowledge graphs containing primarily undirected or bidirectional relationships (such as `isAdjacentTo`), but it becomes a limitation in scenarios where antisymmetric or more complex relation patterns, such as hierarchical relationships (such as `hasProperty`) are prevalent. Still, for large-scale building data requiring rapid inference, DistMult offers a favourable trade-off between speed and performance, especially when combined with optimisations such as negative sampling and regularisation.

5. **ComplEx** builds upon bilinear approaches such as DistMult by allowing entity and relation embeddings to be complex-valued instead of being restricted to real numbers. Specifically, each entity and relation e, r is mapped to a vector $\mathbf{e}, \mathbf{r} \in \mathbb{C}^d$. To score a triple $\big(h, r, t\big)$, ComplEx applies a bilinear form in the complex domain:

$$f(h, r, t) \;=\; \mathrm{Re}\Big(\big\langle \mathbf{h},\, \mathbf{r},\, \bar{\mathbf{t}} \big\rangle\Big) \;=\; \sum_{k=1}^{d} \big(h_k \;\cdot\; r_k \;\cdot\; \overline{t_k}\big),$$

where $\overline{t_k}$ denotes the complex conjugate of t_k, and $\mathrm{Re}(\cdot)$ takes the real part of the resulting sum. By extending the entity and relation representations into the complex plane, ComplEx can model both symmetric and asymmetric relations, thus overcoming the inherent symmetry limitation of DistMult. For instance, when a relation is symmetric (such as `isAdjacentTo`), the imaginary parts of the embeddings cancel out in the scoring function, similarly to DistMult. Conversely, when a relation is asymmetric, phase components in the complex embeddings can effectively capture directionality. The flexibility of handling both symmetric and asymmetric relations may lead to improved link prediction performance across a mix of relational patterns. However, as with other complex-valued models like RotatE, ComplEx demands careful hyperparameter tuning (embedding dimension, learning rate, regularisation) to ensure numerical stability and avoid overfitting. Its additional computational cost compared to real-valued approaches must be weighed against the potential accuracy gains when working with large-scale BIM-KG datasets.

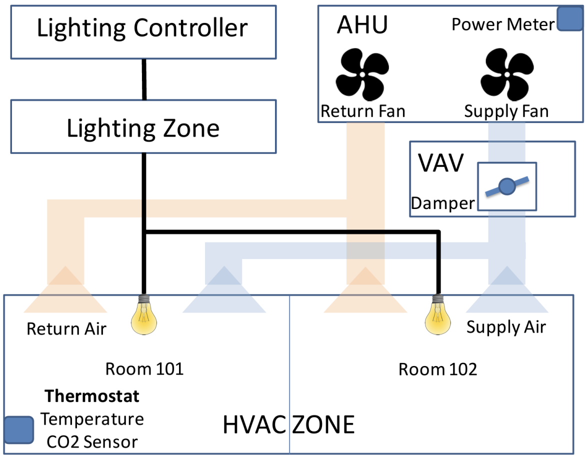


Figure 3.8: A simple example highlighting the components of a building modelled using the Brick schema. Source (Balaji et al., 2016)

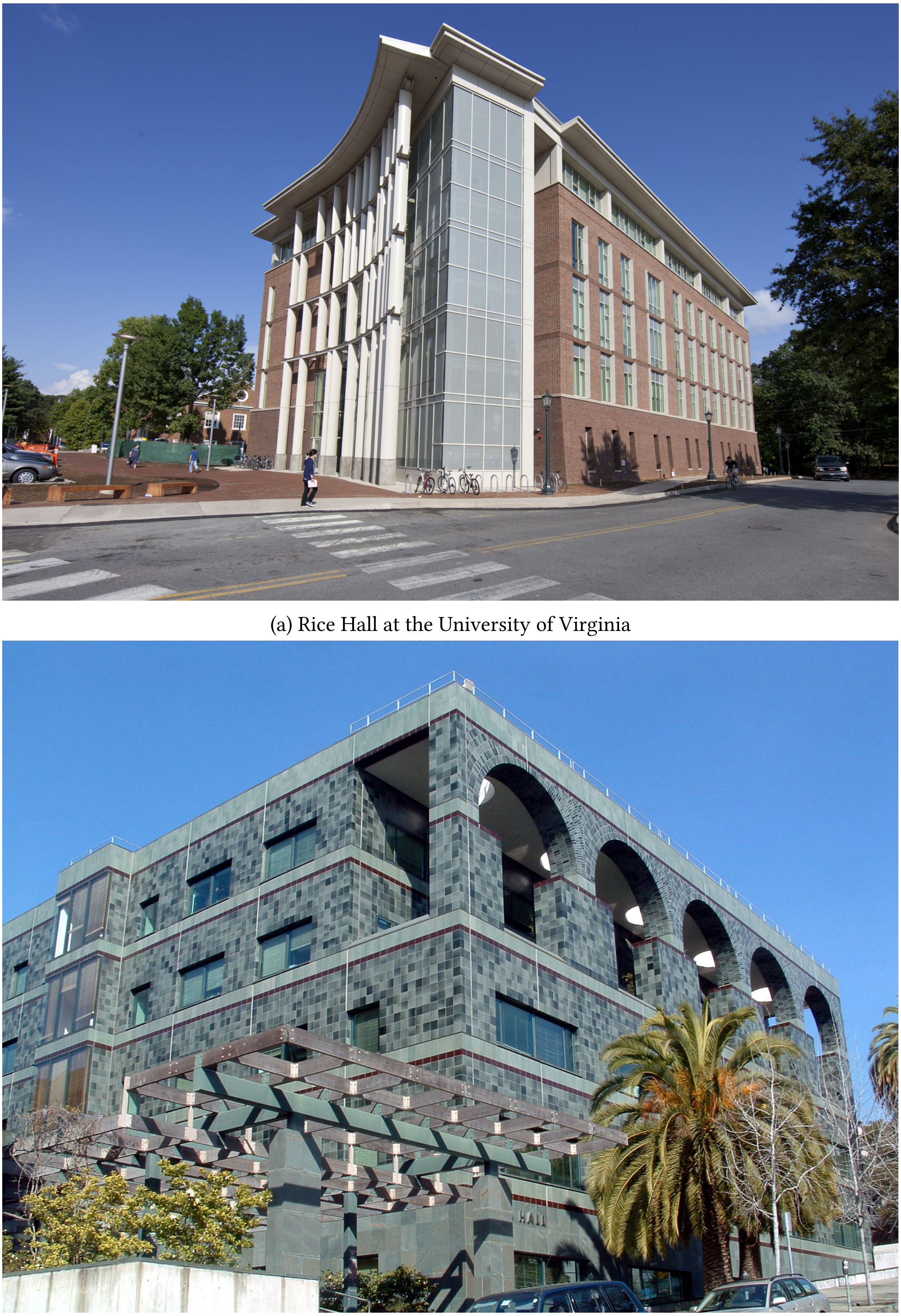
3.2.2 Datasets

In these experiments, two publicly available BIM-KGs were used. These datasets are representative examples of how the Brick ontology[19] can be used to model real buildings - Rice Hall at the University of Virginia (Balaji et al., 2016) and Soda Hall at the University of California, Berkeley (Balaji et al., 2016), as detailed in Table 3.4 - Table 3.7 and shown in Figure 3.9. Brick is an open-source initiative aimed at standardising the semantic descriptions of physical, logical, and virtual assets in buildings and their inter-relationships. An illustrative example of a building modelled using Brick is shown in Figure 3.8. In this example, an Air Handling Unit (AHU) supplies conditioned air to a Variable Air Volume (VAV) Box, which adjusts the airflow to an HVAC zone that has two rooms. The HVAC zone has a thermostat equipped with a temperature and carbon dioxide sensor. Additionally, these two rooms are part of a lighting zone, and the lights in this zone are controlled by the building's lighting controller. The two datasets used in this thesis' experiments closely mirror this example building's setup, but on a larger scale. For an in-depth discussion on the creation and evaluation of these datasets, please refer to Balaji et al. (2016) and Balaji et al. (2018).

Table 3.4: BIM-KG datasets used in the performance analysis experiments

BIM-KG Dataset	$\|\mathcal{E}\|$	$\|\mathcal{K}\|$	$\mathcal{E}$ Types	$\mathcal{R}$ Types
Rice Hall at University of Virginia	810	1665	65	6
Soda Hall at University of California, Berkeley	1738	3774	36	9

[19] https://brickschema.org/



(a) Rice Hall at the University of Virginia

(b) Soda Hall at the University of California, Berkeley

Figure 3.9: The two buildings modelled by BRICK and adopted for this thesis' KRL performance analysis experiments

Table 3.5: Training dataset properties and structural patterns

Dataset	Density	Entity Heterogeneity	Average Degree
Rice Hall	0.002	65	3.664
Soda Hall	0.001	36	4.342

3.2.3 Evaluation Protocol

The overarching goal of KRL models is to learn entity and relation embeddings that capture the underlying structure and semantics of a knowledge graph. To learn reliable embeddings, a KRL model needs to have the ability to distinguish between true facts and false facts in a knowledge graph. In reality, knowledge graphs are curated only with true facts, yet both true and false facts are required for successful KRL model training. When trained exclusively on true facts, a KRL model never encounters invalid examples and thus risks learning a trivial strategy—namely, predicting every possible fact as true. By introducing corruption sets (i.e., negative samples generated by substituting entities in valid triples), the model is encouraged to discriminate between correct and incorrect facts. This process prevents the embeddings from collapsing into a single "*always-true*" mode and enables more robust inference, particularly when dealing with missing or newly introduced facts in a BIM-KG. As mentioned earlier, in the OWA, a fact cannot be considered false just because it does not exist in a knowledge graph. However, when considering the Local Closed World Assumption (LCWA), a constrained variation of the Closed World Assumption (CWA), a knowledge graph is only locally complete, i.e., if a fact `Room_A` $\xrightarrow{\text{ssn:hasProperty}}$ `Temperature` is missing from a BIM-KG, LCWA interprets `Room A` as not having a Temperature property, but only within the context of this specific BIM-KG. This is also known as *(Scoped) Negation as Failure.* With this assumption, a set of negative triples $\mathcal{C}$ can be generated by altering either the head or tail entity of each triple (h, r, t) in a knowledge graph as shown in Equation 3.10 below;

$$\mathcal{C} = \left\{(\hat{h}, r, t) \mid \hat{h} \in \mathcal{E}\right\} \cup \left\{(h, r, \hat{t}) \mid \hat{t} \in \mathcal{E}\right\} \tag{3.10}$$

where $\mathcal{E}$ is the set of all entities in the knowledge graph, while $\hat{h}$ and $\hat{t}$ are the corrupted head and tail entities, respectively. For any given triple (h, r, t) in the test set $\mathcal{K}_{\text{test}} \subseteq \mathcal{K}$, the left hand side of Equation 3.10, $\left\{(\hat{h}, r, t) \mid \hat{h} \in \mathcal{E}\right\}$ corrupts the head entity h by replacing it with any other entity $\hat{h}$ from the set $\mathcal{E}$, creating triples of the form $(\hat{h}, r, t)$

Table 3.6: Key Facts About Rice Hall, University of Virginia

Category	**Details**
Building	Rice Hall, University of Virginia
Function	Hosts the Computer Science Department
Size	100,000+ square feet
Floors	6
Construction Year	2011
Rooms	120+ (faculty offices, teaching/research labs, study areas, conference rooms)
Building Management System (BMS)	Contracted with Trane[20]
HVAC System	- 4 AHUs - 30+ Fan Coil Unit s (FCUs) - 120 VAVs - Low-temperature chilled beams - Ice tank-based chilling towers - Enthalpy wheel heat recovery system - Thermal storage system
Lighting System	- Motorized shades - Abundant daylight sensors - Motion sensors

Table 3.7: Key Facts About Soda Hall, UC Berkeley

Category	**Details**
Building	Soda Hall, UC Berkeley
Function	Houses the Computer Science Department
Size	110,565+ square feet
Floors	5
Construction Year	1994
Rooms	200+ (small to medium-sized closed offices for faculty and graduate student groups)
Building Management System (BMS)	Provided by the now-defunct Barrington Systems; exposes only HVAC sensors
HVAC System	- Pneumatic controls with 232 thermal zones. - Periphery zones have VAVs with reheat - Other zones without reheat. - VAVs with reheat: Single control setpoint for both reheat and airflow using proprietary value mapping mechanism. - Contains redundant chillers, condensers, and cooling towers

where $\hat{h}$ is not the original head entity. Similarly $\{(h, r, \hat{t}) \mid \hat{t} \in \mathcal{E}\}$, corrupts the tail entity by replacing t with any other entity $\hat{t} \in \mathcal{E}$, resulting in triples of the form $(h, r, \hat{t})$ where $\hat{t}$ is not the original tail entity. To visualise this process better, consider a set of 5 entities $\mathcal{E} = \{\texttt{Space}, \texttt{Sensor}, \texttt{Site}, \texttt{Storey}, \texttt{Temperature}\}$ and a set of 2 relations $\mathcal{R} = \{\text{bot:hasElement}, \texttt{bot:adjacentElement}\}$. Given a triple, $\texttt{Space} \xrightarrow{\text{bot:hasElement}} \texttt{Sensor}$, the corruption set $\mathcal{C}$ below can be generated by replacing either the head or tail with entities from $\mathcal{E}$. The first two rows hold corruptions where the tail has been replaced with `Site` and `Temperature`, respectively, while the last two rows hold corruptions where the head has been replaced with `Site` and `Storey`, respectively.

$$\mathcal{C} = \left[\begin{array}{ccc} \textbf{Subject} & \textbf{Relation} & \textbf{Object} \\ \hline \texttt{Space} & \texttt{bot:hasElement} & \texttt{Site} \\ \texttt{Space} & \texttt{bot:hasElement} & \texttt{Temperature} \\ \texttt{Site} & \texttt{bot:hasElement} & \texttt{Sensor} \\ \texttt{Storey} & \texttt{bot:hasElement} & \texttt{Sensor} \end{array}\right]$$

It is possible for some of the generated synthetic negatives to actually be true positives (already existing in $\mathcal{K}$). Whenever these are encountered in this work, they are removed from $\mathcal{K}$ using the filtered evaluation setting (Bordes et al., 2013; Ali et al., 2020a), as their presence can skew the training results.

KRL treats link prediction as a *learning-to-rank* problem. In this paradigm and for the rest of this thesis, a *query* Q takes the form of a partially known triple such as $(h, r, ?)$, where the head entity (h) and the relation (r) are given, but the *tail* $(?)$ is unknown. The model must then assign scores to a set of candidate tails, ranking the correct entity *higher* than incorrect (or corrupted) alternatives. For instance, given a true triple $\texttt{RoomA} \xrightarrow{\text{ssn:hasProperty}} \texttt{Temperature}$ in a BIM-KG, the query $\texttt{RoomA} \xrightarrow{\text{ssn:hasProperty}} ?$ is answered by scoring every possible tail in the knowledge graph and ranking them. A well-trained model places `Temperature` near the top, while all corrupted versions—such as $\texttt{RoomA} \xrightarrow{\text{ssn:hasProperty}} \texttt{RoomB}$—receive lower scores. The *loss function* typically enforces this ranking objective. A common approach is a *margin-based ranking loss*, where the model is penalised if a true triple's score is not sufficiently higher than its corresponding corrupted triple. Formally, for a true triple (h, r, t) and its corrupted counterpart (h', r, t'), the margin-based loss might look like:

$$\mathcal{L} = \sum_{\substack{(h,r,t)\in\mathcal{P} \\ (h',r,t')\in\mathcal{C}}} \max\Big(0,\ \gamma\ +\ f\big(h', r, t'\big)\ -\ f\big(h, r, t\big)\Big),$$

where $f(\cdot)$ is the model's scoring function, γ is a margin, $\mathcal{P}$ denotes the set of positive (true) triples, and $\mathcal{C}$ the corrupted (negative) triples. This loss drives the model to *rank* the true triple higher than any false candidate by a margin of γ. Consequently, when evaluating on a holdout set $\mathcal{K}_{\text{test}}$, a well-performing model will *score* (and thereby rank) the correct triple higher than any corrupted variation—leading to meaningful Mean Rank (MR) or Mean Reciprocal Rank (MRR) metrics that reflect the model's ability to discern true facts from false ones. When multiple triples in the test set, $\mathcal{K}_{\text{test}}$, receive the same score, this work resolves such ties by computing the *mean* of the optimistic and pessimistic rankings. Concretely, in the *optimistic* scenario, the true triple is assumed to occupy the *top* position among all those with equal scores, whereas in the *pessimistic* scenario, it is ranked *last* among them. The final rank is the arithmetic mean of these two extremes, thereby providing a fair tie-break mechanism that avoids inflating or deflating the model's performance metrics. Regarding performance metrics, as a starting point, this work adopts three commonly used ones, namely: Mean Reciprocal Rank (MRR), Hits@K and Adjusted Mean Rank (AMR). These are briefly defined below; however, for a more detailed narrative and discussion, reference is made to Ali et al. (2020a)

1. **MRR**: MRR evaluates the effectiveness of information retrieval systems by calculating the average of the reciprocal ranks of the first relevant result for each query in a set of queries Q. In simpler terms, for each query, the rank position of the first correct answer is identified, the reciprocal of that rank is taken, and then those values are averaged across all queries. This provides a measure of how quickly the system finds the correct answer for queries. Mathematically, MRR is defined as:

$$\text{MRR} = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{\text{rank}_i} \tag{3.11}$$

   where rank_i is the rank of the true triple for the i-th query. For better intuition, imagine a BMS that tries to answer some arbitrary building automation queries Q1, Q2 and Q3.

Table 3.8: Ficticious responses of an illustrative BMS to 3 arbitrary queries and their reciprocal rank values. The correct response is marked with a ✓

Query	Proposed Results	Rank	Reciprocal Rank
Q1	Room101 $\xrightarrow{\text{hasSensor}}$ TempSensorA, Room101 $\xrightarrow{\text{isPartOf}}$ BuildingB, **Room101** $\xrightarrow{\textbf{isPartOf}}$ **BuildingA** ✓	3	$\frac{1}{3}$
Q2	HVAC1 $\xrightarrow{\text{connectedTo}}$ HVAC2, **HVAC1** $\xrightarrow{\textbf{serves}}$ **ZoneA** ✓, HVAC1 $\xrightarrow{\text{hasPart}}$ FanA	2	$\frac{1}{2}$
Q3	Light1 $\xrightarrow{\text{installedIn}}$ Room101, **Light1** $\xrightarrow{\textbf{controlledBy}}$ **ControlSystemA** ✓, Light1 $\xrightarrow{\text{hasSwitch}}$ SwitchB	2	$\frac{1}{2}$

(a) Q1 = Which zone is Room101 part of i.e., Room101 $\xrightarrow{\text{isPartOf}}$? ?

(b) Q2 = Which zone does HVAC1 serve i.e., HVAC1 $\xrightarrow{\text{serves}}$? ?

(c) Q3 = Which system controls Light1 i.e., Light1 $\xrightarrow{\text{controlledBy}}$? ?

For each query, the BMS makes three guesses with the first one being the one it thinks is most likely correct, as shown in Table 3.8 (the actual correct triple is marked with a ✓). MRR ranges from 0 to 1, with 1 indicating that all true triples are ranked first, while 0 shows that none of the proposed results are correct. The MRR for the BMS is therefore calculated as $\frac{\frac{1}{3}+\frac{1}{2}+\frac{1}{2}}{3} \approx 0.44$ using Equation 3.11

2. **Hits@K**: This metric measures the accuracy of a retrieval system by checking whether the correct answer is within the top K ranked results. Mathematically, this is defined using the formula below

$$\text{Hits@K} = \frac{1}{|Q|}\sum_{i=1}^{|Q|} \mathbf{1}(\text{rank}_i \leq \texttt{K}) \tag{3.12}$$

where $\mathbf{1}(\text{rank}_i \leq K)$ is an indicator function that equals 1 if the true triple's rank is within the top K, and 0 otherwise. Hits@1, Hits@3, and Hits@10 are commonly used variations of this metric. Using the example in Table 3.8, hits@1 = $\frac{0}{3} = 0$, hits@3 = $\frac{3}{3} = 1$, hits@10 = $\frac{3}{3} = 1$ (the denominator for all being the 3 true existing facts, while the numerator is the number of correct facts appearing within the top K.

3. **AMR**: This metric complements MRR and Hits@K metrics and has been shown to provide more fair comparisons between datasets of different sizes (Berrendorf et al., 2020). Mathematically, AMR is defined in Equation 3.13 as the ratio of the mean rank to the expected value;

$$\text{AMR} = \frac{\text{Mean Rank (MR)}}{\text{Expected Mean Rank (EMR)}} \tag{3.13}$$

3.2.4 Implementation Details

All experiments were performed on a single MacBook Pro with an Apple M1 Pro chip and 16GB RAM using PyKEEN (Ali et al., 2020b), a Python-based library for KRL built on top of PyTorch, while hyperparameter optimisations were handled using Optuna (Akiba et al., 2019). The exact software versions are kept consistent and delineated in the source code attached to this thesis.

Chapter 4

Results and Discussion

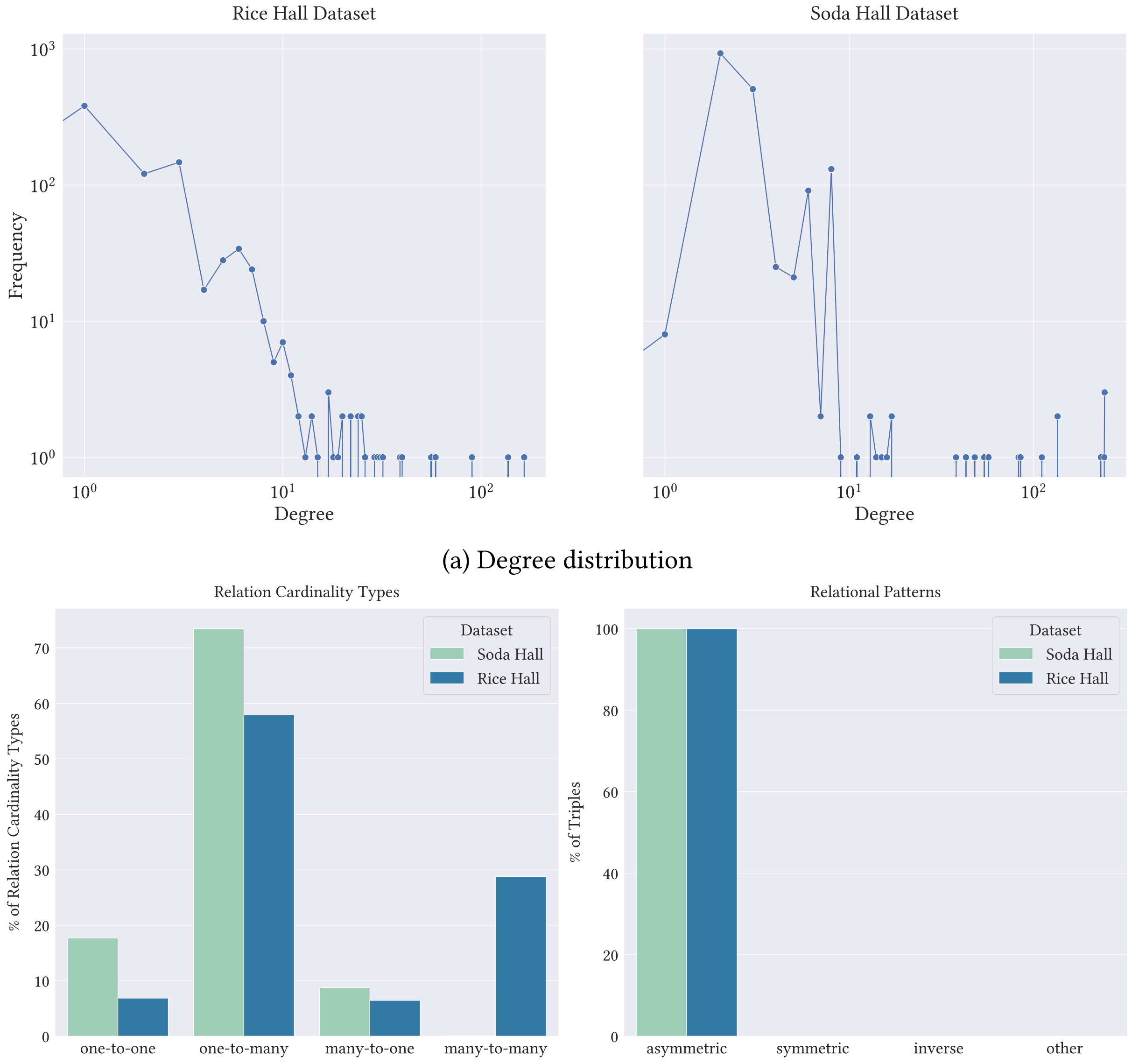
This chapter presents the experimental results obtained from the performance analysis that is delineated in Chapter 3. For an impartial assessment of all models tested during training, a random 10% holdout set of test triples was used. The holdout set is not seen by the models during training or validation. This strategy ensures fair evaluation of each model's generalisation capabilities to new, unseen data. To lay a solid framework for repeatability in future research experiments, throughout the analysis, a consistent set of training setup choices and hyperparameters was maintained, as detailed in Table 4.1.

4.1 A Study On Training Setup Choices

The initial series of experiments aims to evaluate the influence of different categorical decisions related to the training configuration of KRL models. In particular, this method varies the optimizer (selecting from a choice of Adam (Kingma and Ba, 2014), AdaGrad (Duchi et al.,

Table 4.1: Default Training Setup Choices and Hyperparameters

Parameter	**Value By Approach**				
	ComplEx	DistMult	RotatE	TransE	TransH
Embedding Dim	50	50	200	50	50
Num Epochs	500				
Learning Rate	0.02				
Num Negatives	1				
Optimizer	Adagrad				
Inverse Relations	False				
Loss Function	Margin Ranking Loss (Margin 1.0)				



(a) Degree distribution

(b) Relation cardinality types and relation patterns

Figure 4.1: Training dataset degree distributions, relation cardinality types and relation patterns

2011) and Stochastic Gradient Descent (SGD) (Eon Bottou, 1998)), training objective function (selecting from a choice of Binary Cross-Entropy Loss (BCEL) (Dettmers et al., 2017), Softplus Loss (SPL) (Glorot et al., 2011), MRL (Bordes et al., 2013), and the self-adversarial loss (NSSA) (Sun et al., 2019)), and finally considering the exclusion or inclusion of inverse relationships in the BIM-KGs, a process that involves adding a copy of each triple during training but with an inverse relation. A summary of the above categorical choices is presented in Table 4.2.

Figure 4.2 depicts the Hits@10 scores distribution on the test set for different training setup options for all models and datasets. It offers detailed insight into how the models react to variations in the training setup. In the case of the Rice Hall dataset, all models

Table 4.2: Training Setup Choice Matrix. Each setup represents one of 120 unique combinations (5 models × 3 optimisers × 4 loss functions × 2 inverse relationship settings), with each trial trained for 500 epochs as specified in Table 4.1.

Model	Optimizer	Loss Function	Inverse Relationship
All models	Adagrad	Binary Cross Entropy Loss (BCEL)	False
			True
		Softplus Loss (SPL)	False
			True
		Margin Ranking Loss (MRL)	False
			True
		NSSA Loss	False
			True
	Adam	Binary Cross Entropy Loss (BCEL)	False
			True
		Softplus Loss (SPL)	False
			True
		Margin Ranking Loss (MRL)	False
			True
		NSSA Loss	False
			True
	SGD	Binary Cross Entropy Loss (BCEL)	False
			True
		Softplus Loss (SPL)	False
			True
		Margin Ranking Loss (MRL)	False
			True
		NSSA Loss	False
			True

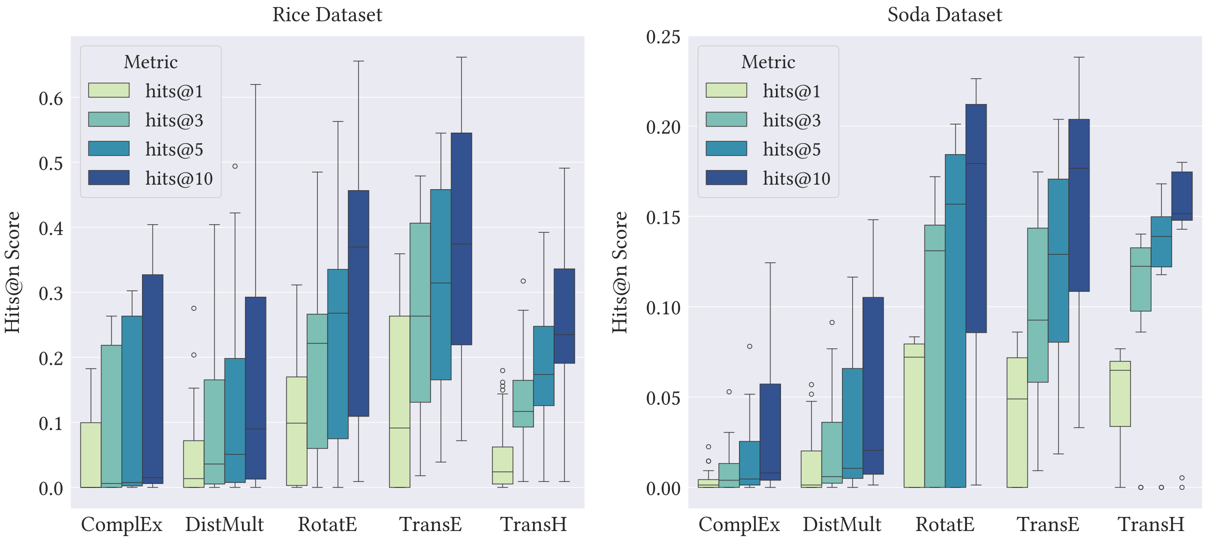
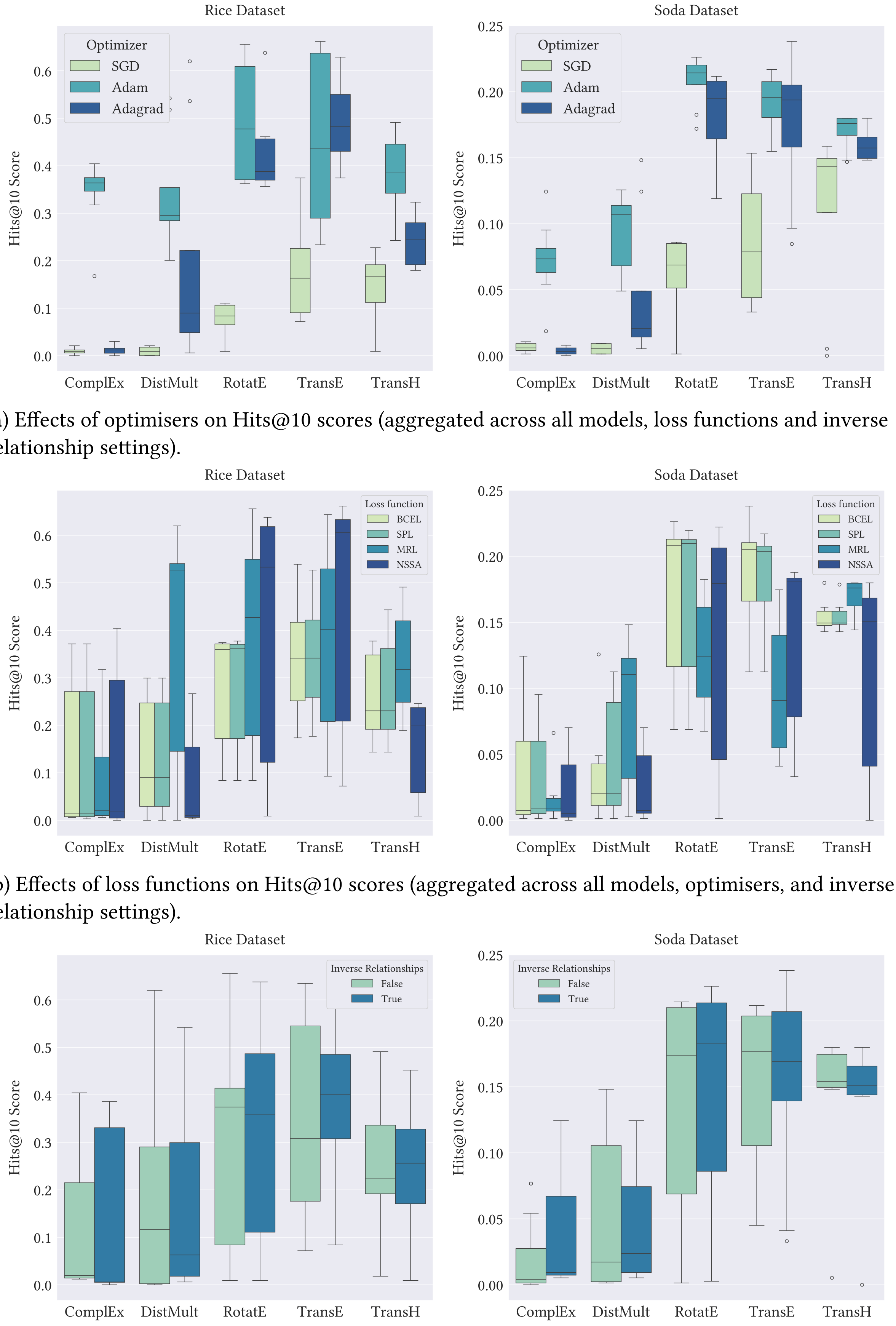


Figure 4.2: Distribution of Hits@n (where $n \in \{1, 3, 5, 10\}$) scores across all categorial training setup choices.



(a) Effects of optimisers on Hits@10 scores (aggregated across all models, loss functions and inverse relationship settings).

(b) Effects of loss functions on Hits@10 scores (aggregated across all models, optimisers, and inverse relationship settings).

(c) Effects of having or not having inverse relationships on Hits@10 scores (aggregated across all models, optimisers, loss functions).

Figure 4.3: The effect of different training setup choices across all models and both datasets.

show a comparable performance range, with TransE and RotatE standing out as the highest performers, whereas DistMult and ComplEx consistently show subpar performance no matter the training setup, a trend that can also be seen on the Soda Hall dataset, albeit happening more aggressively. DistMult inherently assumes symmetric relations due to its bilinear scoring function. However, many relationships in both datasets, such as `feeds` or `isPartOf`, are inherently directional (asymmetric). ComplEx attempts to address this by extending DistMult to complex numbers, allowing it to capture both symmetric and asymmetric relations. Despite this, the real challenge lies in the nuanced, hierarchical, and interdependent relationships typical of BIM-KGs, which ComplEx may not fully capture due to its inability to infer composition patterns. Composition patterns allow a building's multi-faceted relationships to be represented in knowledge graph embeddings. For example, the temperature in a room might be influenced by the operation of HVAC systems, the number of occupants, time of day, and even external weather conditions. Expressive capture of such patterns can enable an automation agent to predict the impact of adjusting the configurations of one system (like the HVAC settings) on various related metrics (such as energy consumption or occupant comfort). Also, buildings often have a hierarchical structure: composed of floors, floors are composed of rooms, and rooms can contain various elements, sensors or actuators. Composition patterns in embeddings can reflect this hierarchy, allowing automation agents to aggregate or disaggregate information at different levels. For instance, understanding the aggregated energy use at the overall building level while also being able to drill down into specific floors or rooms. It is also important to note that BIM-KGs are often characterised by sparse data, with many potential but unobserved relationships between entities, which sparsity challenges the generalisation capacity of these models. TransE and RotatE are less susceptible to overfitting in sparse environments because they embody lower complexity through their respective translational and geometric operations, i.e., TransE needs a single vector to represent each relation as a translation in the embedding space, while RotatE requires a single complex number to represent each relation as a rotation. The lower number of parameters reduces the models' capacity to fit noise, a common pitfall in sparse datasets where the signal-to-noise ratio [1] can be low. Perhaps the most striking observation is that RotatE generally demonstrates superior performance across both datasets; however, as seen in the Rice Hall dataset, older

[1] Signal-to-noise ratio is defined as the ratio of meaningful input to meaningless or unwanted input

methods, such as TransE, can outperform it if given an optimised training setup.

To further explore the effects of various training configurations, other distributions of model performance (Hits@10) for both datasets are presented in Figure 4.3a through Figure 4.3c, revealing some intriguing patterns based on the different choices made. Figure 4.3a indicates that the setups that utilise the Adam optimiser consistently outperform those that use Adagrad and SGD. Adam's adaptive learning rate mechanism and momentum updates likely contribute to its ability to converge faster and escape local minima more effectively. Adagrad also adopts adaptive learning rates, performing smaller updates for parameters associated with frequently occurring features and larger updates for parameters associated with infrequent features. Adam and Adagrad's adaptive learning rate mechanisms make them particularly well suited for tasks with sparse data, where some features frequently occur while others remain rare. However, the monotonic decreasing learning rate of Adagrad can pose challenges in certain scenarios. As the Adagrad algorithm accumulates squared gradients over time, the learning rates for all parameters continuously decrease. While this ensures stable and well-scaled updates, it may also cause the algorithm to prematurely and excessively reduce the learning rate. The poor performance of SGD indicates that its simplistic updating mechanism faces challenges in effectively exploring the complex parameter space of KRL models. Also, SGD does not incorporate adaptive learning rates, which means that it treats all parameters equally, applying the same update magnitude across the board. This uniform approach does not account for the importance of different features within the data. It is important to note that, due to the *no-free-lunch*[2] theorem, there is no one-size-fits-all optimizer; in reality, an optimizer's efficiency is highly reliant on the training setup and unique characteristics of the underlying dataset. This is evident in Figure 4.3a (Soda Hall dataset), where for ComplEx, Adagrad performs worse than SGD.

Looking at Figure 4.3b reveals similar performance for the BCEL and the SPL across both datasets. This is because SPL is equivalent to BCEL though numerically more stable. Even though (Ali et al., 2020a) claims that BCEL is not well-suited for translational distance models, it exhibits competitive performance for TransE on the Soda Hall dataset and even surpasses the numerically more stable SPL. These peculiarities highlight that identifying appropriate training configurations can produce results that deviate from what was previously known.

[2] There is no single optimizer to that will always do better than any other optimizer

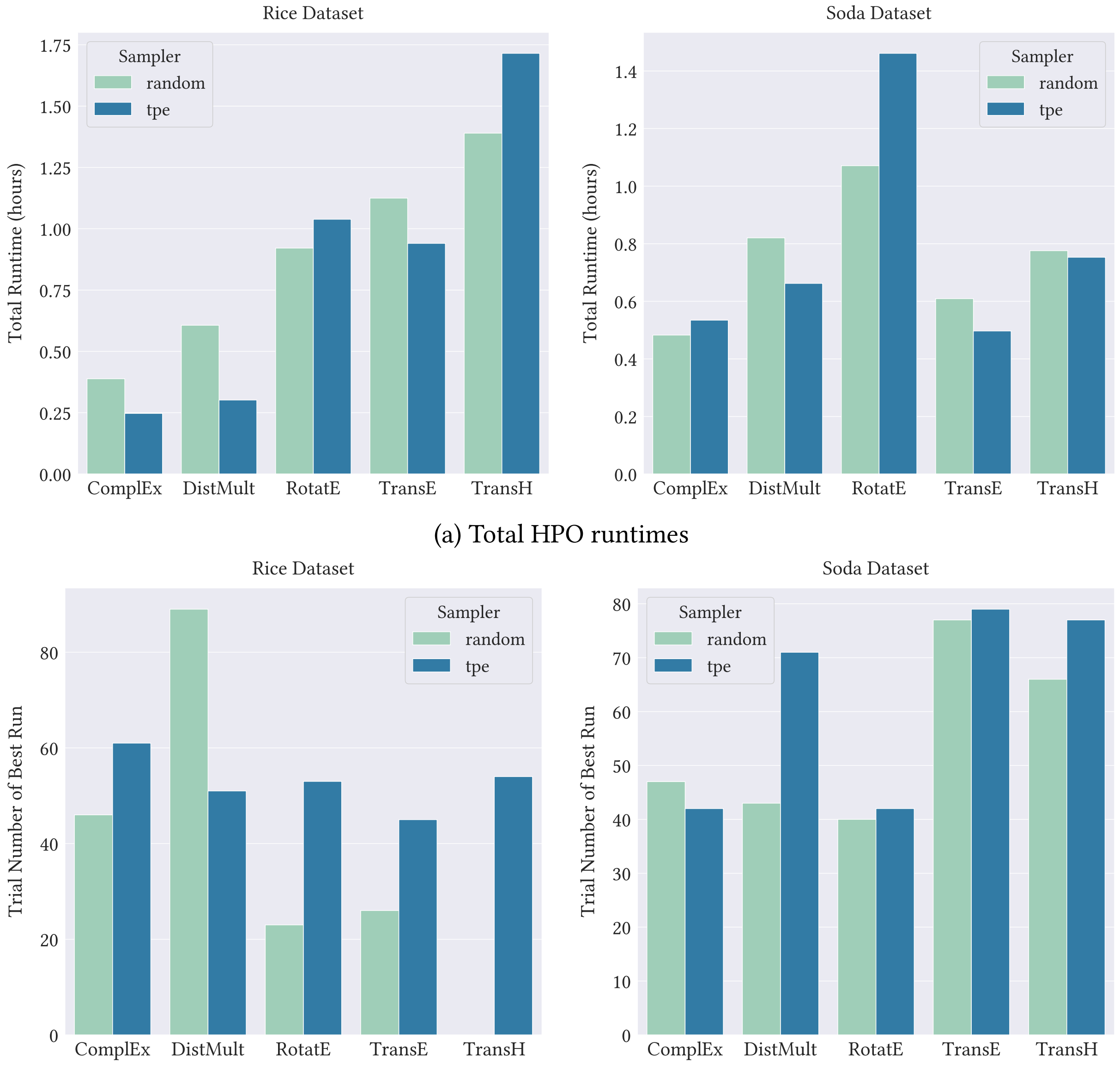
Experimental evidence from the biological domain has shown that adding inverse relationships to the training knowledge graph performs worse than not including them; however, the results herein are contrasting, as shown in Figure 4.3c, where adding inverse relationships to the dataset configurations for ComplEx and RotatE yields better performance. In contrast, the same configuration leads to poor performance for TransH and DistMult when trained on the Soda Hall Dataset. However, when trained on the Rice Hall dataset, Distmult and TransE perform similarly on average. Overall, Figure 4.3 has revealed how combinatorial the problem of configuring the training setup is. Another interesting observation is that improving training setups has proven to enhance performance more significantly than improving model architectures. For instance, the older TransE model has been observed to outperform the newer RotatE model if it is configured suboptimally, as is the case in Figure 4.2. In Figure 4.5c, showing the distribution of Hits@10 scores across all 100 trials using the Tree-structured Parzen Estimator (TPE) method, it is notable that all models are sensitive to the hyperparameters, which means that the best-performing model could easily be outperformed if not carefully tuned.

4.2 A Study On HPO Choices

Even with a robust training configuration, the choice of hyperparameters can significantly affect the model performance. In this study's experiments, two HPO search strategies are employed, i.e., Bayesian TPE (Bergstra et al., 2011) and random search (Bergstra and Bengio, 2012). In the Bayesian TPE, a posterior distribution of the objective function is modelled and used to predict the performance of different hyperparameter configurations based on historical evaluations, while random search selects hyperparameter configurations uniformly at random from a predefined range. For each strategy, 100 experiments are conducted without

Parameter	**Value range**
Embedding Dim	$[16 \ldots 512, 16]$
Num Epochs	$[10 \ldots 50, 10]$
Learning Rate	$[0.001 \ldots 0.1, \log]$
Num Negatives	$[1 \ldots 100, 10]$

Table 4.3: Range of search for HPO values in the form of minimum, maximum and step.

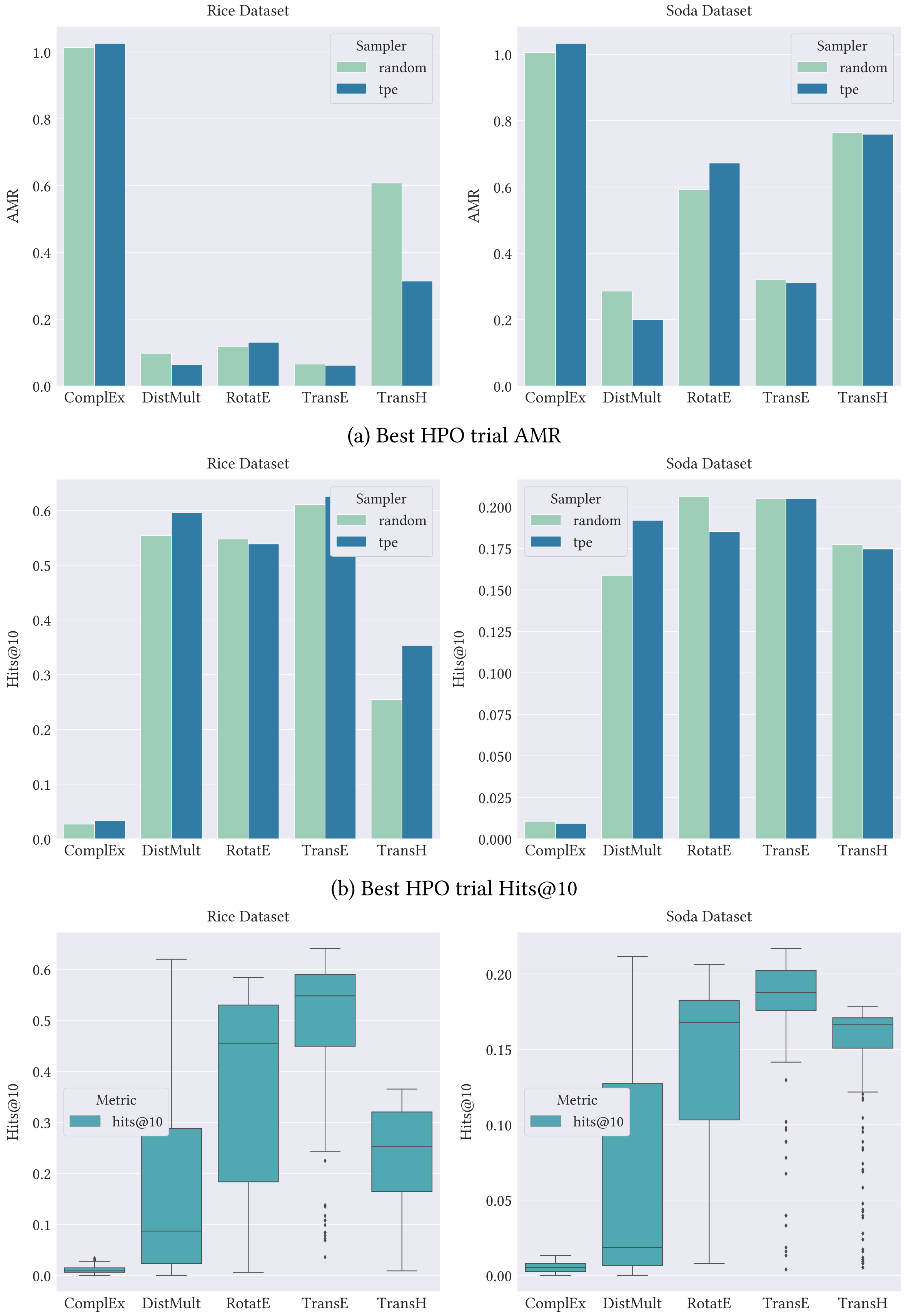


(a) Total HPO runtimes

(b) Best HPO trial number

Figure 4.4: The effect of 2 different HPO strategies across all models and both datasets (Part 1).

time constraints, using a fixed model seed, training setup (see Table 4.1), and hyperparameter optimisation search ranges (see Table 4.3). The hyperparameters were evaluated using the AMR metric for both HPO search strategies on a 10% holdout set of triples that is randomly selected but fixed across all trials. Observing Figure 4.4a, TPE trials took a shorter time than random search on average. Even though TPE's parameter tuning strategies can increase runtime, it managed to achieve shorter trial runs than random search. Figure 4.4b shows that on average, TPE achieves its best performance closer to the maximum number of trials (100). Again, this is likely due to the fact that TPE can tune parameters that can increase run time significantly. In Figure 4.5a and Figure 4.5b, it is interesting to see how close the performance



(a) Best HPO trial AMR

(b) Best HPO trial Hits@10

(c) Distribution of Hits@10 Scores Across all 100 HPO runs using the training setup in Table 4.1, and hyperparameter optimisation search ranges in Table 4.3)

Figure 4.5: The effect of 2 different HPO strategies across all models and both datasets (Part 2).

of TPE is to random search, with TPE yielding only slightly better-performing models. While often considered more naive, random search can occasionally outperform sophisticated methods by randomly finding optimal parameters, potentially uncovering high-performing configurations that systematic searches may miss. Close observation of Figure 4.5a and Figure 4.5b shows a negative correlation between AMR and Hits@10, which is nice to see as the HPO was solely focused on optimising for AMR.

4.3 KRL-BIM-KG Applicability System Architecture

As delineated in Subsection 1.6.4 of the research scope, the system architecture presented here is at a high-level and is positioned in this thesis as a *scaffolding layer* that demonstrates how the combination of KRL and BIM-KGs can be applied in real-life building automation workflows/systems without claiming a fully industrialized deployment. The three guiding assumptions are;

1. No authoritative recipe exists for integrating KRL with BIM-KGs, making a top-down, reference architecture more valuable than a narrowly tailored low-level build.
2. Real buildings mix communication protocols such as BACnet [3], Modbus[4], Message Queuing Telemetry Transport (MQTT)[5], and proprietary APIs; the architecture presented here therefore separates concerns into loosely coupled modules (KRL configurators, RDF triple stores, IoT, LLM-RAG interface) so practitioners can swap implementations to match specific problem constraints.
3. Domain experts—from facility managers to energy consultants—tend to prefer conversational exploration over writing SPARQL. RAG pipelines backed by LLMs are recommended in the stack to translate natural-language questions into SPARQL queries.

Within these boundaries, the system architecture in this section, as summarised in Figure 4.7, delivers the following:

1. **A conceptual pathway**: A clear mapping from BIM/COBie import ⟶ BIM-KG graph + IoT ⟶ KRL embeddings ⟶ vector search ⟶ LLM reasoning.

[3] `https://bacnet.org/`

[4] `https://www.modbus.org/`

[5] `https://mqtt.org/`

Table 4.4: Scalability-oriented design principles in the COBie handler.

Principle	Design choice in code	Why it matters for large projects
Ontology-first mapping	Lines 32–53 bind BRICK, SOSA/SSN, & SAREF4BLDG ontology namespaces	Aligns every triple with public schemas, enabling standard reasoners and avoiding vendor lock-in.
Sheet-level streaming	Each workbook sheet (*Facility*, *Floor*, *Space*, ...) processed in an isolated loop	Allows chunked ingestion; this means that a 30 000-row *Space* sheet stays responsive and memory-safe.
Incremental enrichment	For every BRICK entity created, companion SOSA/SSN property triples (e.g. temperature, humidity) are added immediately.	Data loggers can later stream observations directly to those properties without touching the core topology of the curated BIM-KG.

2. **Interoperable endpoints**: Graph Query Language (GraphQL) and Representational State Transfer (REST) API entry points to abstract low-level framework implementations and also allow external services such as Microsoft Dynamics 365[6] to to contribute business context without being deeply grounded by RDF or KRL principles.

The remainder of this section details each module's workflow, illustrates two representative user interactions, and presents pilot metrics that can be used in future work to validate the architectural soundness.

4.3.1 COBie Handler — Curating BIM-KGs at Scale

The COBie handler is the *front-door data service* that converts raw COBie workbooks into a semantically rich BIM-KG ready for downstream KRL and other modules. COBie is an ideal entry format because it already aggregates most of the asset, spatial, and operational metadata required for lifecycle analytics (*facility*, *floor*, *space*, *component*, ...) in a semi-structured `.xlsx` container. Domain experts are also comfortable with spreadsheet tooling, and the widespread availability of Python, Java, and .NET libraries for spreadsheet parsing means that even large estate owners can ingest decades of legacy FM data without first migrating to a proprietary interface; the handler simply builds on this ecosystem to generate RDF triples. The handler's logic embodies 3 design principles summarised in Table 4.4.

[6] `https://www.microsoft.com/en-us/dynamics-365`

The COBie handler module [7] presented in this implementation architecture follows the workflow below.

1. **Initialize an RDF graph** by creating a new `rdflib.Graph` and binding standard prefixes (`rdf`, `owl`, `xsd`, `brick`, `sosa`, `ssn`, `saref4bldg`) for the different BIM-KG domain ontologies.

2. **Process Facility sheet**: extract site and building identifiers and labels; add `brick:Site`, `brick:Building`, and a `brick:hasPart` link from site to building.

3. **Process Floor sheet**: for each row, instantiate a `brick:Floor` entity, assign an `rdfs:label`, and link it to its building via `brick:hasPart`.

4. **Process Space sheet**: for each valid space, create a `brick:Space` that is also a `sosa:FeatureOfInterest`; add temperature/humidity `ssn:Property` nodes; link spaces to floors with `brick:isPartOf`.

5. **Process Zone & Component sheets**: map zones to `brick:Zone` and establish `brick:hasPart` links to spaces; detect component types (e.g., AHU, VAV), create corresponding BRICK entities with `rdfs:label`, and link them to their containing spaces.

In large-scale applications and deployments of the COBie handler, several operational and performance challenges can emerge. A non-exhaustive list is provided below as a starting point for consideration, together with some proposals.

1. **Write-throughput saturation**

   When a large COBie workbook is parsed, the handler may overwhelm a single Central Processing Unit (CPU) core and the triplestore's insert queue. *Proposed solution:* partition the workload by sheet or by configurable row-chunks and dispatch them to multiple worker processes.

2. **In-memory footprint**

   The default `rdflib.Graph` keeps all triples in-memory, which can exhaust Random

[7] `https://github.com/BIM-and-Automation-Laboratory/coolopt/blob/development/lbd/research_modules/lbd.py`

Access Memory (RAM) on large-scale COBie imports. *Proposed solution:* swap to an `rdflib.ConjunctiveGraph`[8] backed by a SQLAlchemy store[9], or stream triples directly into an external triplestore via batched SPARQL `INSERT/UPDATE` operations, thereby shifting the memory burden on the database layer.

3. **Better rule checking**

   Hard-coded string checks for component types (e.g., "VAV", "Chiller") risk failure when organisations use different naming schemes. *Proposed solution:* externalise the string-to-ontology mapping as a JavaScript Object Notation (JSON) ruleset or SHACL shapes file that the handler loads at runtime; this avoids code changes when new component classes appear.

The COBie handler *anchors semantic consistency* for the entire framework: defects here can propagate into malformed KRL embeddings and misleading LLM answers. Its modular approach allows injection of various BIM-KG validation strategies such as those highlighted in Subsection 3.1.2.

4.3.2 IoT Handler for Real-Time Sensor Integration

The IoT handler is responsible for ingesting live building-automation data and streaming it into a document store such as MongoDB[10]. Each observation can then be linked to its spatial context of the building using the BIM-KG produced by the COBie handler. This work adopts the IoT schematic in Figure 4.6, which comprises a DHT22 sensor connected to an ESP32 microcontroller programmed[11] to connect to a remote MQTT broker and perform the downstream tasks summarised below.

1. **MQTT Subscription:** Connect to the broker, subscribe to topic patterns (such as `building/+/floor/+/space/+/sensor/+/data`).

2. **Message Parsing:** Deserialise each JSON payload to extract timestamp, sensor ID, measurement type, and value.

[8] `https://rdflib.readthedocs.io/en/stable/apidocs/rdflib.html`

[9] `https://www.sqlalchemy.org/`

[10] `https://www.mongodb.com/`

[11] `https://github.com/BIM-and-Automation-Laboratory/esp32-dht22-infrared-setup/blob/main/program.ino`

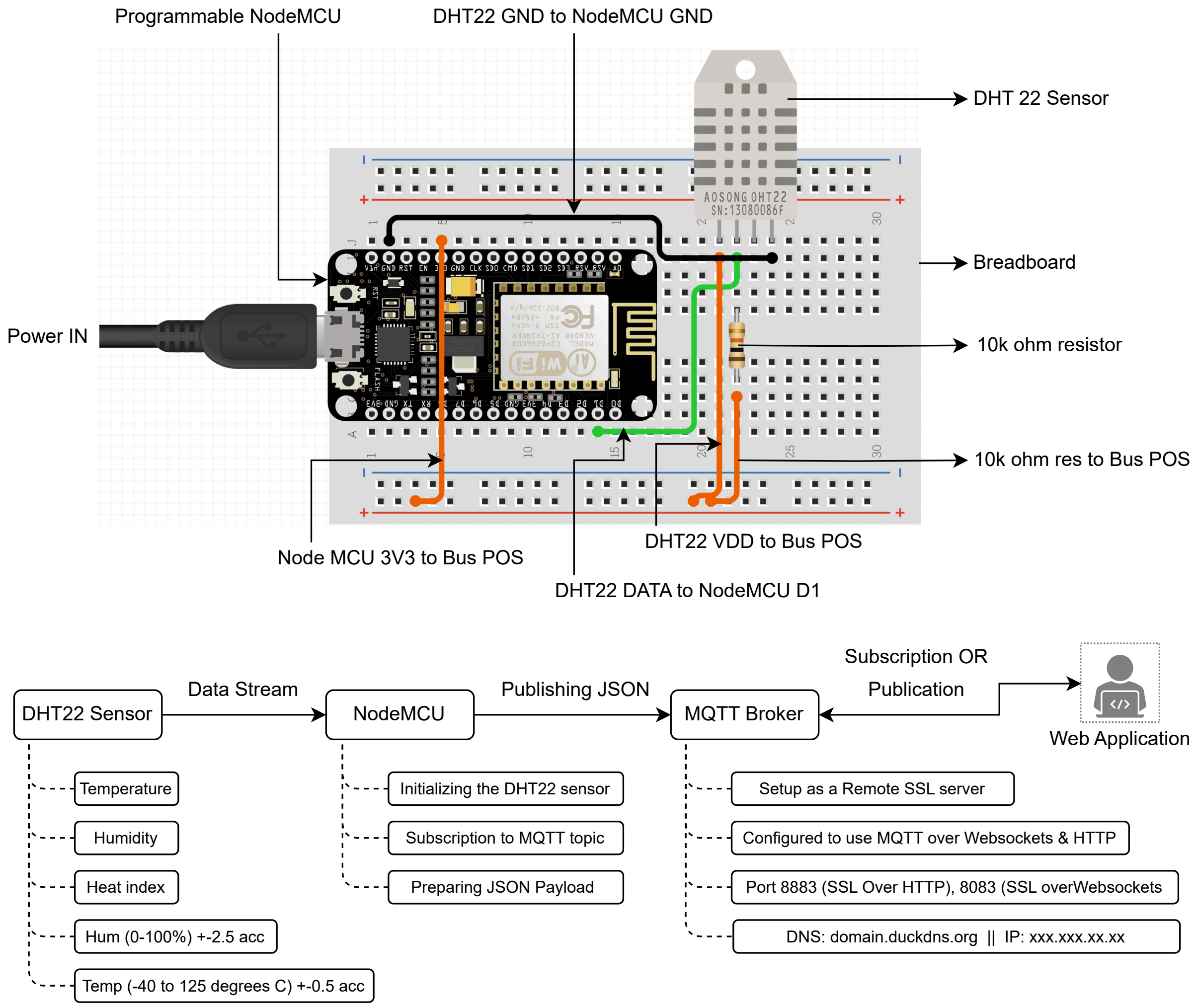


Figure 4.6: DHT22 Setup with ESP32 Micro controller

3. **RDF Triple Generation:** Map sensor IDs to the property nodes created by the COBie handler (e.g. `fdg:GUID-temp`), then emit triples using SOSA/SSN: `sosa:Observation`, `sosa:hasFeatureOfInterest`, `sosa:observedProperty`, `sosa:resultTime`, `sosa:hasSimpleResult`.

4. **Graph Insertion:** Batch append observations to both the MongoDB and triplestore in real time.

In large buildings, the IoT handler must sustain continuous, high-frequency ingestion of measurements while ensuring each observation is semantically linked to the BIM-KG. As device counts and message rates increase, several bottlenecks can degrade performance. The following list summarises some key challenges and proposes strategies to mitigate them.

- **High-Frequency Streams:** Hundreds of sensors publishing every 10–30 seconds can overwhelm a single consumer. *Proposed solution:* shard subscriptions across multiple asynchronous workers or use a Kafka[12] layer to buffer and partition topics before RDF mapping.

- **Batch vs. Single-Message Inserts:** Per-message SPARQL updates can incur network and transaction overhead. It is better to accumulate observations into micro-batches (for example, 100 messages) and issue grouped `INSERT DATA` calls.
- **Back-pressure Management:** Store throttling under load can cause client-side memory growth. *Proposal:* implement a bounded queue with retry/back-off logic or integrate MQTT Quality of Service (QoS) levels to prevent message loss while respecting store capacity.
- **Fault Tolerance and Replayability:** Lost messages during network outages can break the observation history. *Proposal:* enable persistent MQTT queues and log raw payloads to durable storage (e.g. Amazon S3[13]) so that a replay service can re-inject missing data into the BIM-KG.

4.3.3 KRL Configurator Module for Generating Embeddings

The KRL Configurator transforms the static BIM-KG into numeric embeddings for downstream link prediction tasks, similarity search and RAG services. It cleanly separates BIM-KG data ingestion, data preparation, model training, evaluation and exporting of learned embeddings to support profiling, scalability, and modular swaps. The KRL configurator module can be adopted from the experimentation pipelines presented in Section 4.1[14] and Section 4.2[15] using the workflow summarised below.

1. **Import RDF graph**

   Load triples from a SPARQL endpoint or parse a Turtle/TTL file into an in-memory `rdflib.Graph`, ensuring all namespaces (BRICK, SOSA/SSN) are bound.

[12] https://kafka.apache.org/20/documentation/streams/architecture

[13] https://aws.amazon.com/pm/serv-s3/

[14] https://github.com/BIM-and-Automation-Laboratory/phd-source/blob/main/training-setup-study/experiment.ipynb

[15] https://github.com/BIM-and-Automation-Laboratory/phd-source/blob/main/hpo-study/experiment.ipynb

2. **Map URIs to IDs**

   Enumerate every unique subject, predicate, and object URI; assign each a contiguous integer ID; and extract a list of (head_id, relation_id, tail_id) facts.

3. **Prepare training dataset**

   Split the positive facts into training, validation, and test sets; generate negative samples on the fly (corrupting head or tail); and batch the data into tensors.

4. **Dispatch training job**

   Instantiate the selected embedding model (e.g. TransE, ComplEx, TransH) with hyperparameters from a JSON/YAML configuration[16]; and execute training on or Graphics Processing Unit (GPU), streaming mini-batches and logging metrics[17] (loss, MRR, Hits@k).

5. **Export embeddings**

   Upon convergence, serialize entity and relation vectors (e.g. as `.npy` files or directly into a vector database such as Faiss[18] or PGVector[19]); and emit training metadata (model type, parameters, performance) for cataloguing and downstream retrieval.

Beyond the limitations discussed in Subsection 1.6.4, this work does not evaluate the correctness of the generated embeddings. This is because the primary research objective was to expose the mechanics of applying KRL to BIM-KGs—an area where baseline workflows, agreed-upon benchmark datasets, and metrics are still absent. However, both embedding *correctness* and *explainability* highlight a necessary next phase, especially considering that some high-performing models can still violate logical constraints. Classic KRL evaluation—hits@k, MRR—optimises for rank agreement on withheld triples, yet *ignores domain axioms*—`rdfs:domain`, `rdfs:range` or `owl:disjointWith`. A TransE model can score highly while mapping `brick:Space` → `brick:hasPart` → `brick:Chiller`, which contradicts both ontology and building logic. In FM, such contradictions are fatal because

[16] `https://github.com/BIM-and-Automation-Laboratory/phd-source/blob/main/training-setup-study/config.yaml`

[17] `https://github.com/BIM-and-Automation-Laboratory/phd-source/tree/main/training-setup-study/results`

[18] `https://github.com/facebookresearch/faiss`

[19] `https://github.com/pgvector/pgvector`

the LLM-RAG layer may surface them as actionable advice. A typical LLM-RAG workflow would look like this;

1. **Query intake** – Using an LLM such as ChatGPT[20], a facility manager asks: *"Which AHU on Level 2 is closest to failure?"*

2. **Sparse retrieval** – A SPARQL template performs a candidate restriction to `brick:Air_Handling_Unit` instances located in Level 2 spaces.

3. **Dense retrieval** – Candidate URIs are converted to their TransE vectors; a k-NN search pulls the most similar embeddings to historically failed AHUs.

4. **Prompt assembly** – The top-k triples, embedding scores, and maintenance logs form the CONTEXT block of an LLM prompt.

5. **LLM generation** – ChatGPT (via LangChain[21]) produces a ranked list with free-text rationales and confidence scalars.

This RAG pattern exploits the learned embeddings as a *semantic prior*: they bias the LLM towards more factual responses. For this to work, there is a need for ontology-aware regularisation or post-training SHACL and SPARQL audits before embeddings are adopted in operational building-automation workflows. Below are some recommended correctness-oriented extensions.

1. **Constraint-aware negative sampling:** Standard KRL training randomly *corrupts* a positive triple (h, r, t) by replacing either the head or tail with a random entity, yielding (h', r, t) or (h, r, t'); the model then learns to score such negatives lower than the positives. This procedure, however, does not distinguish between ontology-consistent and ontology-violating negatives. By first encoding domain rules as SHACL shapes, it is possible to bias the sampling process toward *only those negatives that break the rules*, thereby teaching the model to respect ontological boundaries. For BRICK, a simple shape might assert that the object of `brick:isPartOf` *must* be a `brick:EquipmentRoot` and *must-not* be a `brick:Space`.

[20] `https://chatgpt.com/`

[21] LangChain is a software framework that helps facilitate the integration of LLMs into applications`https://www.langchain.com/`

During mini-batch construction, negatives are generated, such as texttbrick:Chiller $\rightarrow$ `brick:isPartOf` $\rightarrow$ `brick:Space_2` rather than a semantically neutral corruption like replacing the *Chiller* with an unrelated *Pump*. Because the corrupted triple violates the SHACL shape, the optimiser is forced to push the `Chiller` and `Space` embeddings *apart*, preventing the phenomenon in which disjoint classes occupy the same region of the embedding space. As training iterates over thousands of such rule-breaking negatives, the model internalises the logical constraints, improving the *correctness* of high-ranked predictions without requiring a separate post-processing filter.

2. **Logical-consistency score** — After model convergence, a scoring function $f(h, r, t) \in \mathbb{R}$ assigns plausibility to any triple (h, r, t). The following procedure evaluates whether the model's highest-confidence predictions respect ontology rules.

   (a) **Candidate generation** — For each rule, enumerate all triples that could satisfy it. Example: the rule "every `brick:Zone` `hasPart` at least one `brick:Space`" yields the set $\{(z, \texttt{hasPart}, s) \mid z{:}\texttt{brick:Zone}, s{:}\texttt{brick:Space}\}$.

   (b) **Scoring and ranking** — Apply f to every candidate and sort in descending order; retain the top-N triples (e.g. N=100).

   (c) **Compliance check** — Evaluate each of the N triples against the rule's SHACL shape. A triple linking a Zone to a Space passes; a link to a Chiller or an undefined entity fails.

   (d) **Metric computation** — The logical-consistency score is

$$\text{LCS} = \frac{\text{number of rule-compliant triples in top-}N}{N}.$$

   (e) **Interpretation** — An LCS below 0.9 signals systematic ontology violations, indicating that the embedding space places a significant fraction of disallowed triples among its most confident predictions.

3. **Qualitative RAG audit** – Inject deliberately inconsistent triples into the graph and verify that the embedding-guided RAG output either ignores or explicitly rejects them, providing a user-facing safeguard.

4.3.4 Interoperability Layer — GraphQL and REST Endpoints

This module exposes the framework's core functions through two complementary interfaces:

- **GraphQL API** — a single `/graphql` endpoint that supports typed queries and mutations against the BIM-KG and the vector store. Clients can request precisely the fields they need (e.g. *zone name, latest CO$_2$ reading, top-3 similar zones*) without over-fetching.
- **REST API** — conventional `HTTP GET/POST` routes for common operations such as `/spaces/{id}`, `/zones`, `/embeddings/search`. These routes suit tools or enterprise systems that are not GraphQL-aware.

Exposing these interfaces abstracts away the underlying RDF triples and embedding details so external services can retrieve BIM data without specialist knowledge. There is also added flexibility from pairing GraphQL's fine-grained, exploratory queries with REST's stable, cache-friendly routes for integrating platforms like Microsoft Dynamics 365. These interfaces also align well with widely adopted authentication and DevOps practices—OAuth2, API gateways, monitoring systems—making the framework slot smoothly into enterprise environments without too much custom tooling.

Example use-case : A facilities-management add-in for Microsoft Dynamics 365 calls the REST route `/embeddings/search?iri={ahu_id}&k=5`. The gateway runs a k-nearest-neighbour query on the vector store, retrieves the five most similar AHUs (based on past fault patterns), enriches each result with human-readable labels from the BIM-KG via SPARQL, and returns a JSON array ready for display inside the Dynamics user interface.

4.4 Knowledge Representation Learning-based Building Control Framework (KRL-based BCF)

Insights gained from the experimental results presented in Section 4.1 and Section 4.2, together with the applicability configuration above, were combined to define the key prerequisites for integrating KRL with BIM-KGs in a framework that is arguably domain-agnostic.

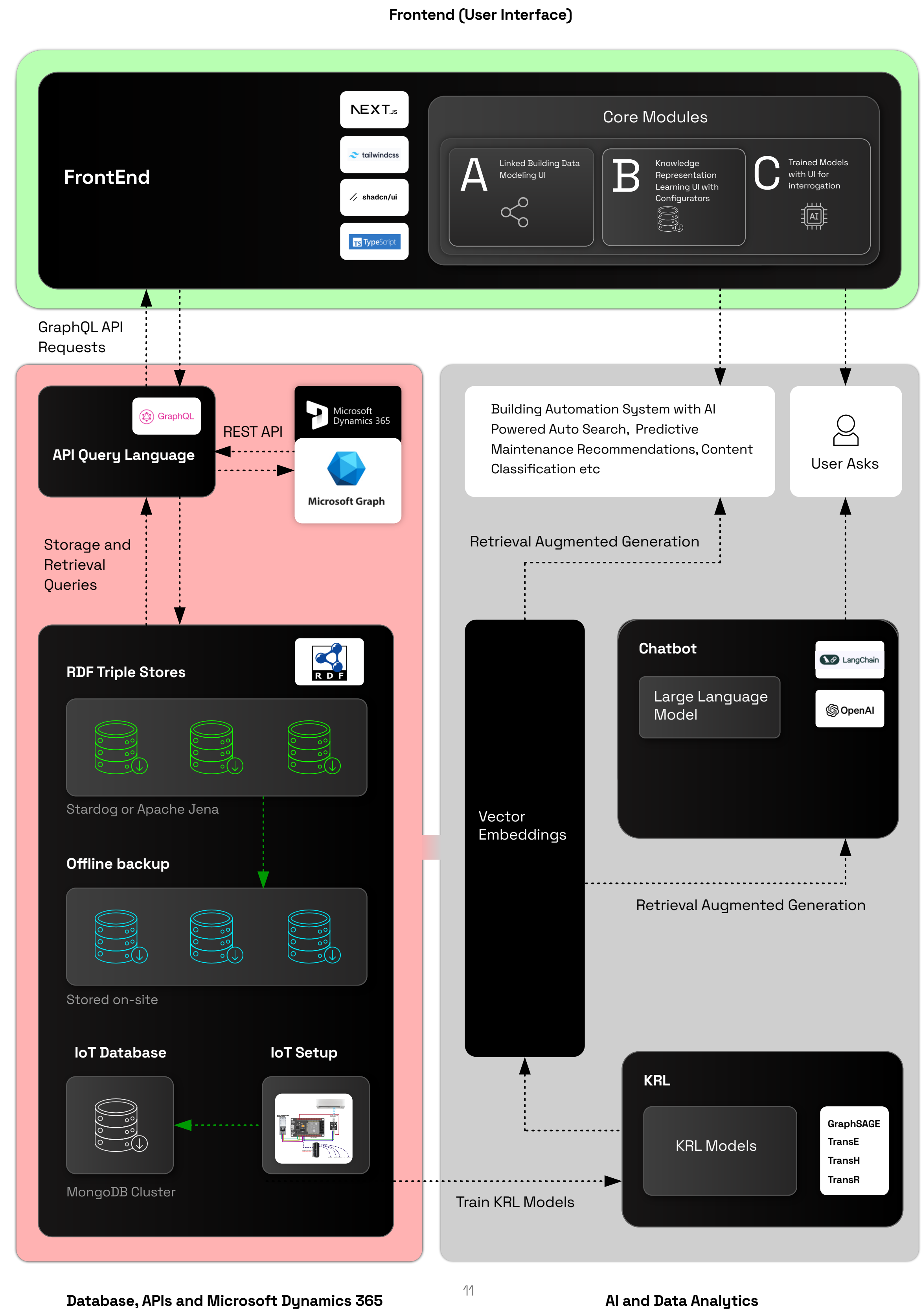


Figure 4.7: Framework Applicability System Architecture

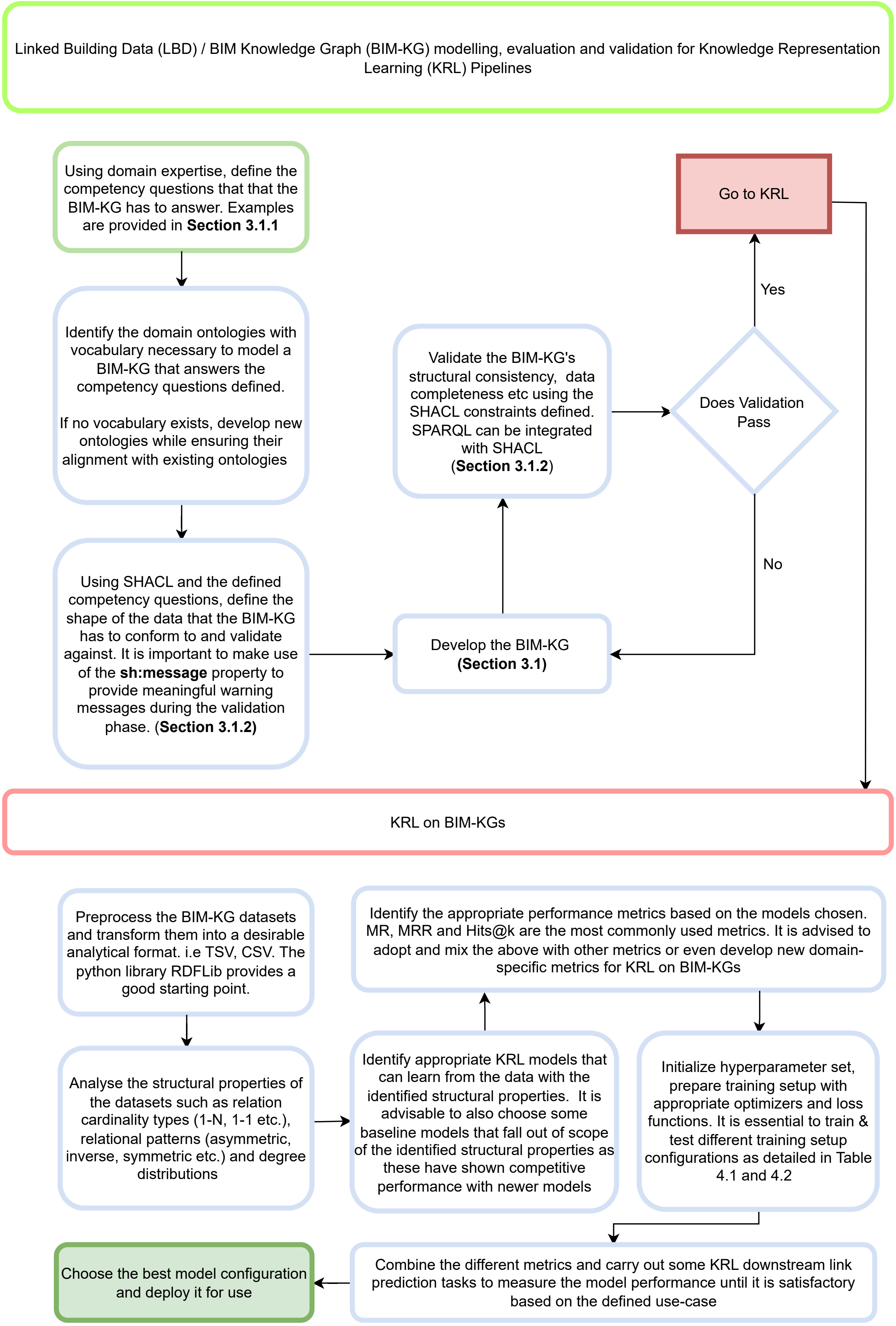

Linked Building Data (LBD) / BIM Knowledge Graph (BIM-KG) modelling, evaluation and validation for Knowledge Representation Learning (KRL) Pipelines
Using domain expertise, define the competency questions that that the BIM-KG has to answer. Examples are provided in **Section 3.1.1**
Identify the domain ontologies with vocabulary necessary to model a BIM-KG that answers the competency questions defined.
If no vocabulary exists, develop new ontologies while ensuring their alignment with existing ontologies
Using SHACL and the defined competency questions, define the shape of the data that the BIM-KG has to conform to and validate against. It is important to make use of the **sh:message** property to provide meaningful warning messages during the validation phase. (**Section 3.1.2**)
Develop the BIM-KG **(Section 3.1)**
Validate the BIM-KG's structural consistency, data completeness etc using the SHACL constraints defined. SPARQL can be integrated with SHACL (**Section 3.1.2**)
Does Validation Pass
Yes
No
Go to KRL
KRL on BIM-KGs
Preprocess the BIM-KG datasets and transform them into a desirable analytical format. i.e TSV, CSV. The python library RDFLib provides a good starting point.
Analyse the structural properties of the datasets such as relation cardinality types (1-N, 1-1 etc.), relational patterns (asymmetric, inverse, symmetric etc.) and degree distributions
Identify appropriate KRL models that can learn from the data with the identified structural properties. It is advisable to also choose some baseline models that fall out of scope of the identified structural properties as these have shown competitive performance with newer models
Identify the appropriate performance metrics based on the models chosen. MR, MRR and Hits@k are the most commonly used metrics. It is advised to adopt and mix the above with other metrics or even develop new domain-specific metrics for KRL on BIM-KGs
Initialize hyperparameter set, prepare training setup with appropriate optimizers and loss functions. It is essential to train & test different training setup configurations as detailed in Table 4.1 and 4.2
Combine the different metrics and carry out some KRL downstream link prediction tasks to measure the model performance until it is satisfactory based on the defined use-case
Choose the best model configuration and deploy it for use

Figure 4.8: Knowledge Representation Learning-based Building Control Framework (KRL-based BCF)

Chapter 5

Conclusions and Recommendations For Future Research

This thesis has examined several aspects of KRL with respect to BIM-KGs, which will be summarized in this chapter while referencing the research questions. This section will conclude with a discussion of interesting directions for future research.

5.1 Summary

BIM-KGs are increasingly being adopted in the AEC/FM field for semantic interoperability and logical inference. Learning from these knowledge graphs using KRL is still in its infancy. This thesis has identified that the efforts to integrate KRL with BIM-KGs are still very slow, primarily due to the absence of standardised procedures for training and evaluating KRL models in a reproducible and fair manner. For KRL to impact the AEC/FM domain, this work has emphasized the critical importance of comprehensively reporting model architectures, training setups and hyperparameters to enhance trust and understanding of KRL-based methods among AEC/FM stakeholders and researchers. This research has addressed the following research questions.

1. **Research Question 1**: *How can knowledge graphs be used to represent the semantic relationships between different building components and systems using domain-agnostic technologies for efficient utilisation in downstream KRL tasks?*

   This thesis has discussed how SWTs can be used to develop BIM-KGs in a data-agnostic fashion. An exploratory walkthrough was made to highlight the technical aspects and key considerations for building an effective BIM-KG for training KRL models. Notably, the need to identify small and modular ontologies using domain-expert competency questions that can be validated against using mechanisms such as SHACL and SPARQL. Furthermore, because the potency of a KRL model is tightly bound to the quality of the input knowledge graph, it is important to check for BIM-KG issues that affect KRL message-passing. By answering this question, a foundation was laid for AEC/FM researchers to explore other important BIM-KG issues to check for before performing downstream KRL tasks.

2. **Research Question 2**: *How can KRL be used to learn the relationships formulated in Research Question 1 for building automation?*

   From the outset, this thesis hypothesised that KRL can be used to learn the hidden patterns within a BIM-KG by leveraging message-passing to propagate learnt information throughout all nodes in the graph. The perception is that imbuing building automation agents with holistic information about the buildings they control can support context-aware decision-making during downstream automation tasks. To answer research question 2, this work used performance analysis experiments to examine how model performance can be affected by modifications to the training step, selection of hyperparameters and their optimisation. This research identified models RotatE and TransE, NSSA loss and Adam optimiser as robust baselines when integrating KRL with BIM-KGs. Throughout the experiments, it was observed that older models like TransE can still be competitive with optimised training and HPO configurations. Furthermore, despite extensive hyperparameter searches, there was considerable variance among top-performing model configurations, indicating the need for nuanced parameter combinations. This complexity suggests that manual tuning may not yield optimal results, advocating for the adoption of HPO strategies. Furthermore, the disparity in hyperparameters between the two datasets underscores the influence of dataset-specific parameters. Finally, random search methods, when repeated sufficiently, yielded configurations comparable to more systematic approaches, albeit in less time.

3. **Research Question 3**: *How can the prerequisites for integrating KRL with BIM-KGs be formalised in a practical framework to enhance trust, reproducibility and understanding of KRL-based methods among AEC/FM stakeholders and researchers?*

   To answer this question, the experimental results from research question 2 were used to deduce the prerequisites for integrating KRL with BIM-KGs, which prerequisites are then used to define a step-by-step framework. To illustrate its implementation, a practical setup is devised consisting of an IoT device and a prototype program of the framework wrapped inside an API. Although a building automation use case is used to formulate the framework, the setup serves as a reference point for extensibility to other AEC/FM domains.

5.2 Recommendations for Future Work

This section briefly outlines interesting directions for future research that can improve the framework's capabilities, explainability, and computational efficiency within the building automation domain.

5.2.1 Enforcing Onset SHACL Validations and Schema Conformity

Instead of attempting to address duplicates and inconsistencies later on in the KRL pipeline, using SHACL restrictions at the outset of BIM-KG curation might be a proactive method to ensure consistency from the beginning. In addition to onset SHACL validations, starting with a clear and consistent schema that specifies the kinds of entities and relationships that will be included in the knowledge graph can improve the quality of the training data used to learn representations, resulting in more accurate and effective models. In its present form, the research's framework does not explicitly account for the erroneous nature of already existing BIM-KGs, which can potentially diminish the accuracy of the learnt KRL embeddings.

5.2.2 Learning From Multi-modal BIM-KGs

Multi-modal BIM-KGs have the capacity to represent different types of building information which are usually of different formats and frequently maintained in separate data silos.

Integrating these modalities into a single knowledge graph can provide a more comprehensive understanding of a building, allowing for more sophisticated reasoning by building control agents. Learning from multi-modal knowledge graphs poses several challenges for KRL. First, the embeddings must be capable of capturing the interactions between multiple modalities, which may necessitate the development of new embedding models. Second, different modalities may have varying degrees of sparsity or noise, necessitating the use of specialised or fine-tuned optimisers, loss functions and performance metrics.

5.2.3 Explainability Improvements

It may not always be obvious how the learnt KRL embeddings were derived or which exact factors influenced them. This research has already shown how combinatorial the problem of choosing a training setup is. This complexity inherently translates to poor model explainability. To alleviate this, future research can focus on including rule-based systems that make the decision-making process more transparent and interpretable, as the reasoning behind the decisions can be traced back to the specific rules being used. This would help building automation specialists better understand and trust the decisions being made in instances where the KRL model's choices directly affect the physical environment and the occupants in it.

5.2.4 A Need for Agreed-upon Fair Evaluation Protocols and Novel Datasets

A major obstacle to the development and assessment of KRL-BIM-KG pipelines is the absence of agreed-upon evaluation protocols and benchmark data sets. To address this issue, it is essential to develop fair and reproducible evaluation protocols for comparing the performance of various KRL-BIM-KG pipelines. Similarly, creating new benchmark datasets that are open to the research community, just like in the biological field, can aid in fairer evaluation of KRL-BIM-KG pipelines.

5.2.5 Security Issues

KRL embeddings capture the holistic context of a knowledge graph, which makes them susceptible to a variety of security vulnerabilities such as adversarial alterations. In a building's context, these can have serious implications for building safety and efficiency. To address these security concerns, several defence mechanisms can be developed, such as encryption and training the KRL model using a mix of clean and adversarial data. Also, an IoT device may encrypt data before delivering it to the KRL model, preventing attackers from intercepting and altering the data to improve its resistance to attacks.

Bibliography

Abdul-Ghafour, S., Ghodous, P., Shariat, B., and Perna, E. (2007). A Common Design-Features Ontology for Product Data Semantics Interoperability. In *IEEE/WIC/ACM International Conference on Web Intelligence*, pages 443–446. IEEE.

Agostinho, C., Dutra, M., Jardim-Gonçalves, R., Ghodous, P., and Steiger-Garção, A. (2007). EXPRESS to OWL morphism: making possible to enrich ISO10303 Modules. In *14th ISPE International Conference on Concurrent Engineering*, pages 391–402, London. Springer.

Akiba, T., Sano, S., Yanase, T., Ohta, T., and Koyama, M. (2019). Optuna: A Next-generation Hyperparameter Optimization Framework. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 2623–2631.

Akompab, D. A., Bi, P., Williams, S., Grant, J., Walker, I. A., and Augoustinos, M. (2013). Awareness of and attitudes towards heat waves within the context of climate change among a cohort of residents in adelaide, australia. *International Journal of Environmental Research and Public Health*, 10(1).

Alam, M., Sanjayan, J., and Zou, P. X. (2019). Balancing energy efficiency and heat wave resilience in building design. In *Climate Adaptation Engineering: Risks and Economics for Infrastructure Decision-Making*.

Ali, M., Berrendorf, M., Hoyt, C. T., Vermue, L., Galkin, M., Sharifzadeh, S., Fischer, A., Tresp, V., and Lehmann, J. (2020a). Bringing Light Into the Dark: A Large-scale Evaluation of Knowledge Graph Embedding Models Under a Unified Framework. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(12):8825–8845.

Ali, M., Berrendorf, M., Hoyt, C. T., Vermue, L., Sharifzadeh, S., Tresp, V., and Lehmann, J. (2020b). PyKEEN 1.0: A Python Library for Training and Evaluating Knowledge Graph Embeddings. *Journal of Machine Learning Research*, 22.

An, X., Li, L. F., Yang, X., and Luo, M. X. (2024). Portable network resolving huge-graph isomorphism problem. *Machine Learning: Science and Technology*, 5(3).

Anderson, A., Marsters, A., Dossick, C. S., and Neff, G. (2012). Construction to operations exchange: Challenges of implementing COBie and BIM in a large owner organization. In *Construction Research Congress 2012: Construction Challenges in a Flat World, Proceedings of the 2012 Construction Research Congress.*

Anzaldi, G., Corchero, A., Wicaksono, H., McGlinn, K., Gerdelan, A., and Dibley, M. J. (2018). Knoholem: Knowledge-Based Energy Management for Public Buildings Through Holistic Information Modeling and 3D Visualization. *International Technology Robotics Applications*, 70:47–56.

Asadi, E., Da Silva, M. G., Antunes, C. H., and Dias, L. (2012). Multi-objective optimization for building retrofit strategies: A model and an application. *Energy and Buildings*, 44(1).

Baader, F. (2003). *The Description Logic Handbook – Theory, Implementation and Applications.* Cambridge University Press, Cambridge, MA, USA.

Babai, L. (2015). Graph Isomorphism in Quasipolynomial Time. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, volume 7443327, pages 684–97. Arxiv.

Balaji, B., Bhattacharya, A., Fierro, G., Gao, J., Gluck, J., Hong, D., Johansen, A., Koh, J., Ploennigs, J., Agarwal, Y., Berges, M., Culler, D., Gupta, R., Kjærgaard, M. B., Srivastava, M., and Whitehouse, K. (2016). Brick: Towards a unified metadata schema for buildings. *Proceedings of the 3rd ACM Conference on Systems for Energy-Efficient Built Environments, BuildSys 2016*, pages 41–50.

Balaji, B., Bhattacharya, A., Fierro, G., Gao, J., Gluck, J., Hong, D., Johansen, A., Koh, J., Ploennigs, J., Agarwal, Y., Bergés, M., Culler, D., Gupta, R. K., Kjærgaard, M. B., Srivastava, M., and Whitehouse, K. (2018). Brick : Metadata schema for portable smart building applications. *Applied Energy*, 226:1273–1292.

Baniassadi, A., Heusinger, J., and Sailor, D. J. (2018). Energy efficiency vs resiliency to extreme heat and power outages: The role of evolving building energy codes. *Building and Environment*, 139:86–94.

Barbau, R., Krima, S., Rachuri, S., Narayanan, A., Fiorentini, X., Foufou, S., and Sriram, R. D. (2012). OntoSTEP: Enriching product model data using ontologies. *CAD Computer Aided Design*, 44(6):575–590.

Barriopedro, D., García-Herrera, R., Ordóñez, C., Miralles, D. G., and Salcedo-Sanz, S. (2023). Heat Waves: Physical Understanding and Scientific Challenges. *Reviews of Geophysics*, 61(2).

Bayoudh, K., Knani, R., Hamdaoui, F., and Mtibaa, A. (2021). A survey on deep multimodal learning for computer vision: advances, trends, applications, and datasets. *The Visual Computer*, 38(8):2939–2970.

Beckett, D. and Berners-Lee, T. (2011). Turtle - Terse RDF Triple Language-W3C Team Submission 28 March 2011.

Beetz, J., van Leeuwen, J., and de Vries, B. (2005). An Ontology Web Language Notation of the Industry Foundation Classes. In *22nd CIB W78 Conference on Information Technology in Construction*, pages 193–198. Technische Universität Dresden.

Beetz, J., van Leeuwen, J., and de Vries, B. (2009). IfcOWL: A case of transforming EXPRESS schemas into ontologies. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing: AIEDAM*, 23(1):89–101.

Bergstra, J., Bardenet, R., Bengio, Y., and Kégl, B. (2011). Algorithms for Hyper-Parameter Optimization. *Advances in Neural Information Processing Systems*, 24.

Bergstra, J. and Bengio, Y. (2012). Random Search for Hyper-Parameter Optimization. *Journal of machine learning research*, 13(1):281–305.

Berners-Lee, T. (1996). WWW Past & Future. *Computer*, 29(10):69–77.

Berners-Lee, T. (2006). Linked Data - Design Issues.

Berners-Lee, T. and Connolly, D. (2011). Notation3 (N3): a readable RDF syntax. W3C Team Submission. *World Wide Web Consortium (W3C)*.

Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The Semantic Web: A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities. *Linking the World's Information: Essays on Tim Berners-Lee's Invention of the World Wide Web*, pages 91–103.

Berrendorf, M., Faerman, E., Vermue, L., and Tresp, V. (2020). On the Ambiguity of Rank-Based Evaluation of Entity Alignment or Link Prediction Methods. *arXiv preprint arXiv:2002.06914*.

Bonino, D. and De Russis, L. (2018). DogOnt as a viable seed for semantic modeling of AEC/FM. *Semantic Web*, 9(6):763–780.

Bonner, S., Barrett, I. P., Ye, C., Swiers, R., Engkvist, O., Hoyt, C. T., and Hamilton, W. L. (2022). Understanding the performance of knowledge graph embeddings in drug discovery. *Artificial Intelligence in the Life Sciences*, 2:100036.

Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., and Yakhnenko, O. (2013). Translating Embeddings for Modeling Multi-relational Data. *Advances in neural information processing systems*, 26:2787–2795.

Borrmann, A., König, M., Koch, C., and Beetz, J. (2018). *Building Information Modeling Technology Foundations and Industry Practice.* Springer International Publishing.

Bronstein, M. M., Bruna, J., Lecun, Y., Szlam, A., and Vandergheynst, P. (2016). Geometric deep learning: going beyond Euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42.

Chen, J., Lin, H., Han, X., and Sun, L. (2024). Benchmarking Large Language Models in Retrieval-Augmented Generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17754–17762. Association for the Advancement of Artificial Intelligence.

Chen, K. W., Janssen, P., and Schlueter, A. (2018a). Multi-objective optimisation of building form, envelope and cooling system for improved building energy performance. *Automation in Construction*, 94:449–457.

Chen, Y., Norford, L. K., Samuelson, H. W., and Malkawi, A. (2018b). Optimal control of HVAC and window systems for natural ventilation through reinforcement learning. *Energy and Buildings*, 169:195–205.

Corneil, D. G. and Gotlieb, C. C. (1970). An Efficient Algorithm for Graph Isomorphism. *Journal of the ACM (JACM)*, 17(1):51–64.

Corry, E., Pauwels, P., Hu, S., Keane, M., and O'Donnell, J. (2015). A performance assessment ontology for the environmental and energy management of buildings. *Automation in Construction*, 57:249–259.

Cunningham, P. and Delany, S. J. (2007). K -Nearest Neighbour Classifiers. *Multiple Classifier Systems*, pages 1–17.

Curry, E., O'Donnell, J., and Corry, E. (2012). Building Optimisation using Scenario Modeling and Linked Data. In *First International Workshop on Linked Data in Architecture and Construction.*

Dai, Y., Wang, S., Xiong, N. N., and Guo, W. (2020). A Survey on Knowledge Graph Embedding: Approaches, Applications and Benchmarks. *Electronics 2020, Vol. 9, Page 750*, 9(5):750.

Daniele, L., den Hartog, F., and Roes, J. (2015). Created in Close Interaction with the Industry: The Smart Appliances REFerence (SAREF) Ontology. *7th International Workshop, FOMI 2015 Berlin, Germany, August 5, 2015 Proceedings*, (August):102–112.

Debruyne, C., McGlinn, K., McNerney, L., and O'Sullivan, D. (2017). A lightweight approach to explore, enrich and use data with a geospatial dimension with semantic web technologies. *ACM*, (May):1–6.

Defferrard, M., Bresson, X., and Vandergheynst, P. (2016). Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 3844–3852.

Delgarm, N., Sajadi, B., Kowsary, F., and Delgarm, S. (2016). Multi-objective optimization of the building energy performance: A simulation-based approach by means of particle swarm optimization (PSO). *Applied Energy*, 170:293–303.

Deng, M., Fu, B., Menassa, C. C., and Kamat, V. R. (2023). Learning-Based personal models for joint optimization of thermal comfort and energy consumption in flexible workplaces. *Energy and Buildings*, 298.

Dettmers, T., Minervini, P., Stenetorp, P., and Riedel, S. (2017). Convolutional 2D Knowledge Graph Embeddings. *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, pages 1811–1818.

Dibley, M., Li, H., Miles, J., and Rezgui, Y. (2011). Towards intelligent agent based software for building related decision support. *Advanced Engineering Informatics*, 25(2):311–329.

Dibley, M., Li, H., Rezgui, Y., and Miles, J. (2012). An ontology framework for intelligent sensor-based building monitoring. *Automation in Construction*, 28:1–14.

Dolenc, M., Katranuschkov, P., Gehre, A., Kurowski, K., and Turk, Z. (2007). The inteligrid platform for virtual organisations Interoperability. *Electronic Journal of Information Technology in Construction*, 12:459–477.

Dong, Y., Coleman, M., and Miller, S. A. (2021). Greenhouse Gas Emissions from Air Conditioning and Refrigeration Service Expansion in Developing Countries. *Annual Review of Environment and Resources*, 46(1):59–83.

Duchi, J., Hazan, E., and Singer, Y. (2011). Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *The Journal of Machine Learning Research*, 12(7).

East, E. W., Nisbet, N., and Liebich, T. (2013). Facility Management Handover Model View. *Journal of Computing in Civil Engineering*, 27(1):61–67.

El-Mekawy, M. (2010). *Integrating BIM and GIS for 3D City Modelling: The Case of IFC and CityGML.* PhD thesis, Royal Institute of Technology (KTH).

Elghamrawy, T. and Boukamp, F. (2008). A vision for a framework to support management and learning from construction problems. In *Proceedings of the 25th International Conference on Formation Technology in Construction: Improving the management of Construction Projects through IT adoption*, number 1517, pages 1–10.

Elghamrawy, T. and Boukamp, F. (2010). Managing construction information using RFID-based semantic contexts. *Automation in Construction*, 19(8):1056–1066.

Eon Bottou, L. (1998). Online Learning and Stochastic Approximations. *Online learning in neural networks*, 17(9):142.

Futia, G. and Vetrò, A. (2020). On the Integration of Knowledge Graphs into Deep Learning Models for a More Comprehensible AI—Three Challenges for Future Research. *Information (Switzerland)*, 11(2):122.

Gandon, F. and Schreiber, G. (2014). RDF 1.1 XML Syntax W3C Recommendation 25 February 2014 (2014).

Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Wang, M., and Wang, H. (2023). Retrieval-Augmented Generation for Large Language Models: A Survey. *arXiv preprint arXiv:2312.10997*, 2.

Garey, M. R. and Johnson, D. S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness (Series of Books in the Mathematical Sciences).* W. H. Freeman.

Ge, X., Wang, Y.-C., Wang, B., and Kuo, C. C. J. (2023). Knowledge Graph Embedding: An Overview. *APSIPA Transactions on Signal and Information Processing*, 13(1).

Ginestet, C. (2010). Introduction to Statistical Relational Learning. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 173(4).

Glorot, X., Bordes, A., and Bengio, Y. (2011). Deep Sparse Rectifier Neural Networks. In *International Conference on Artificial Intelligence and Statistics*, pages 315–323. JMLR Workshop and Conference Proceedings.

Gómez-Romero, J., Bobillo, F., Ros, M., Molina-Solana, M., Ruiz, M., and Martín-Bautista, M. (2015). A fuzzy extension of the semantic Building Information Model. *Automation in Construction*, 57:202–212.

Graves, A., Mohamed, A. R., and Hinton, G. (2013). Speech recognition with deep recurrent neural networks. In *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pages 6645–6649. IEEE.

Guo, D., Onstein, E., and La Rosa, A. D. (2021). A Semantic Approach for Automated Rule Compliance Checking in Construction Industry. *IEEE Access*, 9:129648–129660.

Haller, A., Janowicz, K., Cox, S., Le Phuoc, D., Taylor, K., and Lefrançois, M. (2017). Semantic Sensor Network (SSN) Ontology-W3C Recommendation 19 October 2017.

Halmos, P. R. (1974). *Naive Set Theory*. Undergraduate Texts in Mathematics. Springer New York, New York, NY.

Hamilton, W. L., Ying, Z., and Leskovec, J. (2017). Inductive Representation Learning on Large Graphs. In *31st Conference on Neural Information Processing Systems (NIPS),*.

Harris, S. and Seaborne, A. (2013). SPARQL 1.1 Query Language-W3C Recommendation 21 March 2013.

Hitzler, P., Krötzsch, M., Parsia, B., F.Patel-Schneider, P., and Rudolph, S. (2012). OWL 2 Web Ontology Language Primer (Second Edition)-W3C Recommendation 11 December 2012.

Hoff, P. D. (2007). Modeling homophily and stochastic equivalence in symmetric relational data. *Advances in Neural Information Processing Systems 20 - Proceedings of the 2007 Conference*.

Hopke, J. E. (2020). Connecting Extreme Heat Events to Climate Change: Media Coverage of Heat Waves and Wildfires. *Environmental Communication*, 14(4):492–508.

International Energy Agency (IEA) (2023). Energy Efficiency-The Decade for Action Ministerial Briefing IEA 8th Annual Global Conference on Energy Efficiency Versailles. Technical report, International Energy Agency: IEA, Paris.

ISO 10303-11 (2004). Industrial automation systems and integration – Product data representation and exchange – Part 11: Description methods: The EXPRESS language reference manual. *Geneva: International Organization for Standardization*.

ISO 16739:2024 (2024). Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries. *Geneva: International Organization for Standardization*.

ISO 29481-1 (2016). Building information modelling- Information delivery manual- Part 1: Methodology and format. *Geneva: International Organization for Standardization.*

Jabeen, S., Li, X., Amin, M. S., Bourahla, O., Li, S., and Jabbar, A. (2023). A Review on Methods and Applications in Multimodal Deep Learning. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 19(2s):1–41.

Jia, X., Pan, Y., Zhu, M., Zhu, H., Li, Z., Zhang, J., Zhou, X., Pan, S., Wang, C., Yan, D., Wu, Z., Deng, H., Pan, Y., Xie, J., and Xu, L. (2023). Occupant behavior modules development for coupled simulation in DeST 3.0. *Energy and Buildings*, 297.

Junk, J., Goergen, K., and Krein, A. (2019). Future heat waves in different european capitals based on climate change indicators. *International Journal of Environmental Research and Public Health*, 16(20).

Juran, J. M., Gryna, F. M., and Bingham, R. S. (1979). *Quality control handbook.*, volume 3. McGraw-hill New York.

Kellogg, G. and Champin, P.-A. (2019). JSON-LD 1.1-A JSON-based Serialization for Linked Data.

Kingma, D. P. and Ba, J. L. (2014). Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*. International Conference on Learning Representations, ICLR.

Kipf, T. N. and Welling, M. (2016). Semi-Supervised Classification with Graph Convolutional Networks. *arXiv preprint*, pages 1–14.

Knight, S.-a. and Burn, J. (2005). Developing a Framework for Assessing Information Quality on the World Wide Web. *Informing Science*, 8.

Kofler, M. J., Reinisch, C., and Kastner, W. (2012). A semantic representation of energy-related information in future smart homes. *Energy and Buildings*, 47:169–179.

Kolda, T. G. and Bader, B. W. (2009). Tensor decompositions and applications. *SIAM Review*, 51(3):455–500.

Kriebel-Gasparro, A. (2022). Climate Change: Effects on the Older Adult. *Journal for Nurse Practitioners*, 18(4):372–376.

Krima, S., Barbau, R., Fiorentini, X., Sudarsan, R., and Sriram, R. D. (2009). OntoSTEP : OWL-DL Ontology for STEP. *National Institute of Standards and Technology, NISTIR*, 7561.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems 25 (NIPS)*, pages 1097–1105.

Kumar, V. and Teo, A. L. E. (2021a). Development of a rule-based system to enhance the data consistency and usability of COBie datasheets. *Journal of Computational Design and Engineering*, 8(1):343–361.

Kumar, V. and Teo, E. A. L. E. (2021b). Exploring the application of property graph model in visualizing COBie data. *Journal of Facilities Management*, 19(4):500–526.

Le, Q. V. (2013). Building high-level features using large scale unsupervised learning. In *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pages 8595–8598.

Le, T. and David Jeong, H. (2016). Interlinking life-cycle data spaces to support decision making in highway asset management. *Automation in Construction*, 64:54–64.

LeCun, Y., Boser, B. E., Denker, J. S., Hernderson, D., Howard, R. E., Hubbard, W. E., and Jackel, L. D. (1990). Handwritten Digit Recognition with a Back-Propagation Network. In *Advances in neural information processing systems*, pages 396–404.

Lefrançois, M., Kalaoja, J., Ghariani, T., and Zimmermann, A. (2016). SEAS Knowledge Model. Technical report.

Li, G., Muller, M., Thabet, A., and Ghanem, B. (2019). DeepGCNs: Can GCNs Go as Deep as CNNs? *Proceedings of the IEEE International Conference on Computer Vision*, 2019-October:9266–9275.

Lin, Y., Han, X., Xie, R., Liu, Z., and Sun, M. (2018). Knowledge Representation Learning: A Quantitative Review. *arXiv preprint arXiv:1812.10901*.

Lin, Y., Liu, Z., Sun, M., Liu, Y., and Zhu, X. (2015). Learning Entity and Relation Embeddings for Knowledge Graph Completion. *Proceedings of the AAAI Conference on Artificial Intelligence*, 29(1):2181–2187.

Liu, Z., Sun, M., Lin, Y., and Xie, R. (2016). Knowledge representation learning: A review. *Jisuanji Yanjiu yu Fazhan/Computer Research and Development*, 53(2):247–261.

Lösch, U., Bloehdorn, S., and Rettinger, A. (2012). Graph kernels for RDF data. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 7295 LNCS, pages 134–148.

Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 1150–1157.

Lu, S., Wang, S., Hameen, E., Shi, J., and Zou, Y. (2019a). Comfort-based Integrative HVAC System With Non-intrusive Sensing In Office Buildings. In *Annual Conference of the Association for Computer-Aided Architectural Design Research in Asia-CAADRIA*, volume 1, pages 785–794.

Lu, S., Wang, W., Lin, C., and Hameen, E. C. (2019b). Data-driven simulation of a thermal comfort-based temperature set-point control with ASHRAE RP884. *Building and Environment*, 156:137–146.

MacLean, F. (2021). Knowledge graphs and their applications in drug discovery. *Expert Opinion on Drug Discovery*, 16(9):1057–1069.

Madjiheurem, S. and Toni, L. (2019). Representation Learning on Graphs: A Reinforcement Learning Application. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 3391–3399. PMLR.

Mannan, M. and Al-Ghamdi, S. G. (2021). Indoor Air Quality in Buildings: A Comprehensive Review on the Factors Influencing Air Pollution in Residential and Commercial Structure. *International Journal of Environmental Research and Public Health*, 18(6):1–24.

Manola, F., Miller, E., and McBride, B. (2014). RDF 1.1 Primer-W3C Working Group Note 24 June 2014.

Mario, E., Raffaele, L., Onofrio, C., Maria, C.-S. J., Valentina, B., Vincenzo, G., Shao, C., and Giovanni, S. (2024). Coupling heat wave and wildfire occurrence across multiple ecoregions within a Eurasia longitudinal gradient. *Science of The Total Environment*, 912:169269.

Mason, K. and Grijalva, S. (2019). A Review of Reinforcement Learning for Autonomous Building Energy Management. *Computers & Electrical Engineering*, 78:300–312.

McGlinn, K., Debruyne, C., McNerney, L., and O'Sullivan, D. (2017). Integrating building information models with authoritative Irish geospatial information. In *ISWC (Posters, Demos & Industry Tracks)*, volume 1963, pages 1–4.

McGlinn, K., Wicaksono, H., Lawton, W., Weise, M., Kaklanis, N., Petri, I., and Tzovaras, D. (2016). Identifying Use Cases and Data Requirements for BIM Based Energy Management Processes. In *CIBSE Technical Symposium*.

Merlet, Y., Rouchier, S., Jay, A., Cellier, N., and Woloszyn, M. (2022). Integration of phasing on multi-objective optimization of building stock energy retrofit. *Energy and Buildings*, 257.

Miller, S., Chua, K., Coggins, J., and Mohtadi, H. (2021). Heat waves, climate change, and economic output. *Journal of the European Economic Association*, 19(5):2658–2694.

Mitchell, D., Heaviside, C., Vardoulakis, S., Huntingford, C., Masato, G., P Guillod, B., Frumhoff, P., Bowery, A., Wallom, D., and Allen, M. (2016). Attributing human mortality during extreme heat waves to anthropogenic climate change. *Environmental Research Letters*, 11(7).

Muggleton, S. and de Raedt, L. (1994). Inductive Logic Programming: Theory and methods. *The Journal of Logic Programming*, 19(1):629–679.

Nawi, M., Baluch, N., and Bahauddin, A. Y. (2014). Impact of Fragmentation Issue in Construction Industry: An Overview. In *MATEC Web of Conferences*, volume 15, page 1009.

Nguyen, T. H. and Grishman, R. (2015). Relation Extraction: Perspective from Convolutional Neural Networks. In *NAACL-HLT*, pages 39–48. Association for Computational Linguistics (ACL).

Nickel, M. (2013). *Tensor Factorization for Relational Learning*. PhD thesis, Ludwig Maximilians Universität München.

Nickel, M., Murphy, K., Tresp, V., and Gabrilovich, E. (2016). A review of relational machine learning for knowledge graphs. In *Proceedings of the IEEE*, volume 104, pages 11–33. Institute of Electrical and Electronics Engineers Inc.

Nickel, M., Tresp, V., and Kriegel, H.-P. (2011). A Three-Way Model for Collective Learning on Multi-Relational Data. In *ICML*, pages 3104482–3104584.

Nickel, M., Tresp, V., and Kriegel, H.-P. (2012). Factorizing YAGO: Scalable machine learning for Linked Data. In *Proceedings of the 21st international conference on World Wide Web*, pages 271–280.

Niles, I. and Pease, A. (2001). Towards a standard upper ontology. *Proceedings of the international conference on Formal Ontology in Information Systems - FOIS '01*, pages 2–9.

O'Brien, W., Tahmasebi, F., Andersen, R. K., Azar, E., Barthelmes, V., Belafi, Z. D., Berger, C., Chen, D., De Simone, M., Simona d'Oca, Hong, T., Jin, Q., Khovalyg, D., Lamberts, R., Novakovic, V., Park, J. Y., Plagmann, M., Rajus, V. S., Vellei, M., Verbruggen, S., Wagner, A., Willems, E., Yan, D., and Zhou, J. (2020). An international review of occupant-related aspects of building energy codes and standards. *Building and Environment*, 179.

O'Donnell, J., See, R., Rose, C., Maile, T., Bazjanac, V., and Haves, P. (2011). SIMMODEL : A domain data model for whole building energy simulation. In *Proceedings of Building Simulation2011: 12th Conference of International Building Performance Simulation Association*, pages 382–389.

Pan, J., Anumba, C., and Ren, Z. (2004). Potential Application of the Semantic Web. In *20th Annual Conference of the Association of Researchers in Construction Management (ARCOM), Heriot Watt University EdinBurgh*, volume 2, pages 923–929.

Park, J. Y., Dougherty, T., Fritz, H., and Nagy, Z. (2019a). LightLearn: An adaptive and occupant centered controller for lighting based on reinforcement learning. *Building and Environment*, 147.

Park, J. Y., Mistur, E., Kim, D., Mo, Y., and Hoefer, R. (2022). Toward human-centric urban infrastructure: Text mining for social media data to identify the public perception of COVID-19 policy in transportation hubs. *Sustainable Cities and Society*, 76.

Park, J. Y. and Nagy, Z. (2018). Comprehensive analysis of the relationship between thermal comfort and building control research - A data-driven literature review. *Renewable and Sustainable Energy Reviews*, 82:2664–2679.

Park, J. Y., Ouf, M. M., Gunay, B., Peng, Y., O'Brien, W., Kjærgaard, M. B., and Nagy, Z. (2019b). A critical review of field implementations of occupant-centric building controls. *Building and Environment*, 165.

Pascanu, R., Mikolov, T., and Bengio, Y. (2012). On the difficulty of training Recurrent Neural Networks. *30th International Conference on Machine Learning, ICML 2013*, (PART 3):1310–1318.

Pauwels, P., Corry, E., and O'Donnell, J. (2014a). Representing SimModel in the Web Ontology Language. In *Computing in Civil and Building Engineering (2014)*, pages 2271–2278, Reston, VA. American Society of Civil Engineers.

Pauwels, P., Corry, E., and O'Donnell, J. (2014b). Making SimModel information available as RDF graphs. *eWork and eBusiness in Architecture, Engineering and Construction*, pages 439–445.

Pauwels, P., Costin, A., and Rasmussen, M. H. (2022). Knowledge Graphs and Linked Data for the Built Environment. In *Industry 4.0 for the Built Environment: Methodologies, Technologies and Skills*, pages 157–183. Springer.

Pauwels, P., De Meyer, R., and Van Campenhout, J. (2010). Interoperability for the Design and Construction Industry through Semantic Web Technology. In *International Conference on Semantic and Digital Media Technologies*, pages 143–158.

Pauwels, P., Krijnen, T., Terkaj, W., and Beetz, J. (2017a). Enhancing the ifcOWL ontology with an alternative representation for geometric data. *Automation in Construction*, 80:77–94.

Pauwels, P., McGlinn, K., Törmä, S., and Beetz, J. (2018). Linked Data. In *Building Information Modeling Technology Foundations and Industry Practice*, pages 181–197. Springer.

Pauwels, P. and Roxin, A. (2017). SimpleBIM: From full ifcOWL graphs to simplified building graphs Building Topology Ontology (BOT) View project SemanticGIS View project. In *eWork and eBusiness in Architecture, Engineering and Construction: ECPPM 2016 (11 European Conference on Product and Process Modelling)*, pages 11–18. CRC Press.

Pauwels, P. and Terkaj, W. (2016). EXPRESS to OWL for construction industry: Towards a recommendable and usable ifcOWL ontology. *Automation in Construction*, 63:100–133.

Pauwels, P., Terkaj, W., Krijnen, T., and Beetz, J. (2015). Coping with lists in the ifcOWL ontology. *22nd EG-ICE International Workshop*, pages 113–122.

Pauwels, P., van den Bersselaar, E., and Verhelst, L. (2024). Validation of technical requirements for a BIM model using semantic web technologies. *Advanced Engineering Informatics*, 60:102426.

Pauwels, P., Zhang, S., and Lee, Y. C. (2017b). Semantic web technologies in AEC industry: A literature overview. *Automation in Construction*, 73:145–165.

Peng, R. D., Bobb, J. F., Tebaldi, C., McDaniel, L., Bell, M. L., and Dominici, F. (2011). Toward a quantitative estimate of future heat wave mortality under global climate change. *Environmental Health Perspectives*, 119(5):710–706.

Radulovic, F., Poveda-Villalón, M., Vila-Suero, D., Rodríguez-Doncel, V., García-Castro, R., and Gómez-Pérez, A. (2015). Guidelines for Linked Data generation and publication: An example in building energy consumption. *Automation in Construction*, 57:178–187.

Rasmussen, M. H., Lefrançois, M., Bonduel, M., Hviid, C. A., and Karlshø, J. (2018). OPM: An ontology for describing properties that evolve over time. In *CEUR Workshop Proceedings*, volume 2159, pages 23–33.

Rasmussen, M. H., Lefrançois, M., Schneider, G. F., and Pauwels, P. (2019). BOT: the Building Topology Ontology of the W3C Linked Building Data Group. *Semantic Web Journal*, 12(1):143–161.

Rasmussen, M. H., Pauwels, P., Hviid, C. A., and Karlshøj, J. (2017a). Proposing a Central AEC Ontology That Allows for Domain Specific Extensions. In *Joint Conference on Computing in Construction*, volume 1, pages 237–244.

Rasmussen, M. H., Pauwels, P., Lefrançois, M., Schneider, G. F., Hviid, C., and Karlshøj, J. (2017b). Recent changes in the Building Topology Ontology. In *5th Linked Data in Architecture and Construction Workshop*.

Reinisch, C., Kofler, M. J., Iglesias, F., and Kastner, W. (2011). Thinkhome energy efficiency in future smart homes. *EURASIP Journal on Embedded Systems*, 2011:1–18.

Ricquebourg, V., Durand, D., Menga, D., Marhic, B., Delahoche, L., Logé, C., and Jolly-Desodt, A. M. (2007). Context inferring in the smart home: An SWRL approach. In *Proceedings - 21st International Conference on Advanced Information Networking and Applications Workshops/Symposia, AINAW'07*, volume 2, pages 290–295. IEEE.

Ristoski, P. and Paulheim, H. (2016). RDF2Vec: RDF graph embeddings for data mining. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9981 LNCS:498–514.

Rossello-Busquet, A., Brewka, L. J., Soler, J., and Dittmann, L. (2011). OWL Ontologies and SWRL Rules Applied to Energy Management. In *2011 UkSim 13th International Conference on Computer Modelling and Simulation*, pages 446–450. IEEE.

Russomanno, D. J., Kothari, C. R., and Thomas, O. A. (2005). Building a Sensor Ontology: A Practical Approach Leveraging ISO and open geospatial consortium (OGC) models. In *The 2005 International Conference on Artificial Intelligence, Las Vegas, NV*, pages 637–643.

Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., and Monfardini, G. (2009). The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80.

Scherer, R., Katranuschkov, P., Kadolsky, M., and Laine, T. (2012). Ontology-based building information model for integrated lifecycle energy management. In *eWork and eBusiness in Architecture, Engineering and Construction*, pages 951–956. CRC Press.

Schevers, H. and Drogemuller, R. (2005). Converting the Industry Foundation Classes to the Web Ontology Language. In *2005 First International Conference on Semantics, Knowledge and Grid*, pages 73–73. IEEE.

Schlichtkrull, M., Kipf, T. N., Bloem, P., van den Berg, R., Titov, I., and Welling, M. (2017). Modeling Relational Data with Graph Convolutional Networks. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 10843 LNCS:593–607.

Schneider, G. F. (2017). Towards Aligning Domain Ontologies with the Building Topology Ontology. In *5th LDAC workshop, 13-15 November.*

Shah, N., Chao, K. M., Zlamaniec, T., and Matei, A. (2011). Ontology for home energy management domain. In *Digital Information and Communication Technology and Its Applications: International Conference, DICTAP 2011, Dijon, France, June 21-23, 2011, Proceedings, Part II*, volume 167 CCIS, pages 337–347. Springer Berlin Heidelberg.

Shaikh, P. H., Nor, N. B. M., Nallagownden, P., and Elamvazuthi, I. (2018). Intelligent multi-objective optimization for building energy and comfort management. *Journal of King Saud University - Engineering Sciences*, 30(2):195–204.

Shen, Y., Guo, C., Li, H., Chen, J., Guo, Y., and Qiu, X. (2021). Financial Feature Embedding with Knowledge Representation Learning for Financial Statement Fraud Detection. *Procedia Computer Science*, 187:420–425.

Sommaruga, L., Perri, A., and Furfari, F. (2005). DomoML-env: An ontology for human home interaction. In *CEUR Workshop Proceedings*, volume 166, pages 1–8.

Sterne, J. A., White, I. R., Carlin, J. B., Spratt, M., Royston, P., Kenward, M. G., Wood, A. M., and Carpenter, J. R. (2009). Multiple imputation for missing data in epidemiological and clinical research: Potential and pitfalls. *BMJ (Online)*, 339(7713):157–160.

Stolk, S. and McGlinn, K. (2020). Validation of IfcOWL datasets using SHACL. In *8th Workshop on Linked Data in Architecture and Construction (LDAC 2020) in: CEUR Workshop Proceedings*, pages 91–104. CEUR Workshop Proceedings.

Studer, R., Grimm, S., and Abecker, A. (2007). *Semantic web services: Concepts, technologies, and applications.* Springer.

Sun, Z., Deng, Z. H., Nie, J. Y., and Tang, J. (2019). RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. *7th International Conference on Learning Representations, ICLR 2019.*

Tah, J. H. and Abanda, H. F. (2011). Sustainable building technology knowledge representation: Using Semantic Web techniques. *Advanced Engineering Informatics*, 25(3):547–558.

Teicholz, P. (2013). *BIM for facility managers.* John Wiley and Sons Inc.

Terkaj, W. and Šojić, A. (2015). Ontology-based representation of IFC EXPRESS rules: An enhancement of the ifcOWL ontology. *Automation in Construction*, 57:188–201.

Thomas, E. and Bowman, J. (2021). *Harnessing the Data Advantage in Construction.* San Rafael, CA: AUTODESK and FMI.

Toffolo, A. and Lazzaretto, A. (2002). Evolutionary algorithms for multi-objective energetic and economic optimization in thermal system design. *Energy*, 27(6):549–567.

Tomic, S., Fensel, A., and Pellegrini, T. (2010). SESAME Demonstrator: Ontologies, Services and Policies for Energy Efficiency. In *6th International Conference on Semantic Systems, I-SEMANTICS '10*, pages 1–4.

Veličković, P., Casanova, A., Liò, P., Cucurull, G., Romero, A., and Bengio, Y. (2017). Graph Attention Networks. *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings.*

Venugopal, M., Eastman, C. M., and Teizer, J. (2015). An ontology-based analysis of the industry foundation class schema for building information model exchanges. *Advanced Engineering Informatics*, 29(4):940–957.

Viguié, V., Lemonsu, A., Hallegatte, S., Beaulant, A. L., Marchadier, C., Masson, V., Pigeon, G., and Salagnac, J. L. (2020). Early adaptation to heat waves and future reduction of air-conditioning energy use in Paris. *Environmental Research Letters*, 15(7).

Völker, J. and Niepert, M. (2011). Statistical Schema Induction. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6643 LNCS(PART 1):124–138.

W3C (2014). Linked Building Data Community Group.

W3C-Linked Data Community Group (2018). Product Ontology (PRODUCT).

W3C OWL Working Group (2012). OWL 2 Web Ontology Language Document Overview (Second Edition)-W3C Recommendation 11 December 2012.

W3C SPARQL Working Group (2013). SPARQL 1.1 Overview-W3C Recommendation 21 March 2013.

Wainwright, M. J. and Jordan, M. I. (2008). Graphical Models, Exponential Families, and Variational Inference. *Foundations and Trends® in Machine Learning*, 1(1–2):1–305.

Wang, H., Wang, J., Feng, Z., Haghighat, F., and Cao, S. J. (2023). An intelligent anti-infection ventilation strategy: From occupant-centric control and computer vision perspectives. *Energy and Buildings*, 296.

Wang, Q., Mao, Z., Wang, B., and Guo, L. (2017). Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, 29(12):2724–2743.

Wang, R. Y. and Strong, D. M. (1996). Beyond Accuracy: What Data Quality Means to Data Consumers. *Journal of Management Information Systems*, 12(4):5–34.

Wang, X., Chen, Z., Wang, H., U, L. H., Li, Z., and Guo, W. (2024). Large Language Model Enhanced Knowledge Representation Learning: A Survey. *arXiv preprint arXiv:2407.00936*.

Wang, Z., Zhang, J., Feng, J., and Chen, Z. (2014). Knowledge Graph Embedding by Translating on Hyperplanes. *Proceedings of the AAAI Conference on Artificial Intelligence*, 28(1):1112–1119.

Werbrouck, J., Pauwels, P., and Bekers, W. (2018). *Linking Data : Semantic enrichment of the existing building geometry*. PhD thesis, Ghent University.

Werbrouck, J., Senthilvel, M., Beetz, J., and Pauwels, P. (2019). A Checking Approach for Distributed Building Data. In *31. Forum Bauinformatik*, pages 173–181. Universitätsverlag der TU Berlin.

Wetherill, M., Rezgui, Y., Lima, C., and Zarli, A. (2002). Knowledge management for the construction industry: The e-COGNOS project. *Electronic Journal of Information Technology in Construction*, 7:183–196.

Wijeratne, W. M. U., Samarasinghalage, T. I., Yang, R. J., and Wakefield, R. (2022). Multi-objective optimisation for building integrated photovoltaics (BIPV) roof projects in early design phase. *Applied Energy*, 309.

Wilcke, X., Bloem, P., and de Boer, V. (2017). The knowledge graph as the default data model for learning on heterogeneous knowledge. *Data Science*, 1(1-2):39–57.

Woods, J., James, N., Kozubal, E., Bonnema, E., Brief, K., Voeller, L., and Rivest, J. (2022). Humidity's impact on greenhouse gas emissions from air conditioning. *Joule*, 6(4):726–741.

Xu, M. (2021). Understanding Graph Embedding Methods and Their Applications. *SIAM Review*, 63(4):825–853.

Yang, B., Yih, W. t., He, X., Gao, J., and Deng, L. (2014). Embedding Entities and Relations for Learning and Inference in Knowledge Bases. In *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*. ICLR.

Yang, Q. and Zhang, Y. (2006). Semantic interoperability in building design: Methods and tools. *Computer-Aided Design*, 38(10):1099–1112.

Ye, Y. and Ji, S. (2019). Sparse Graph Attention Networks. *IEEE Transactions on Knowledge and Data Engineering*, 35(1):905–916.

Yi, H. C., You, Z. H., Huang, D. S., and Kwoh, C. K. (2022). Graph representation learning in bioinformatics: Trends, methods and applications. *Briefings in Bioinformatics*, 23(1).

Yong, Z., Li-juan, Y., Qian, Z., and Xiao-yan, S. (2020). Multi-objective optimization of building energy performance using a particle swarm optimizer with less control parameters. *Journal of Building Engineering*, 32.

Yurchyshyna, A. and Zarli, A. (2009). An ontology-based approach for formalisation and semantic organisation of conformance requirements in construction. *Automation in Construction*, 18(8):1084–1098.

Yurchyshyna, A., Zucker, C. F., Le Thanh, N., Lima, C., and Zarli, A. (2007). Towards an Ontology-based Approach for Conformance Checking Modeling in Construction. In *Proceedings of the 24th CIB W78 Conference*, pages 195–202.

Zaveri, A., Kontokostas, D., Sherif, M. A., Bühmann, L., Morsey, M., Auer, S., and Lehmann, J. (2013). User-driven quality evaluation of DBpedia. In *I-SEMANTICS '13: Proceedings of the 9th International Conference on Semantic Systems*, pages 97–104. ACM International Conference Proceeding Series.

Zaveri, A., Rula, A., Maurino, A., Pietrobon, R., Lehmann, J., and Auer, S. (2016). Quality assessment for Linked Data: A Survey. *Semantic Web*, 7(1):63–93.

Zhang, C. (2019). *Requirement checking in the building industry : enabling modularized and extensible requirement checking systems based on semantic web technologies.* PhD thesis, Technische Universiteit Eindhoven.

Zhang, C., Beetz, J., and de Vries, B. (2017). BimSPARQL: Domain-specific functional SPARQL extensions for querying RDF building data. *Semantic Web*, 9(6):829–855.

Zhang, C., Beetz, J., and Weise, M. (2014). Model view checking: automated validation for IFC building models. *eWork and eBusiness in Architecture, Engineering and Construction*, 0:123–128.

Zhang, J., Seet, B.-C., and Lie, T. (2015). Building Information Modelling for Smart Built Environments. *Buildings*, 5(1):100–115.

Zhang, Z., Cui, P., and Zhu, W. (2018). Deep Learning on Graphs: A Survey. *IEEE Transactions on Knowledge and Data Engineering*, 34(1):249–270.

Zhang, Z., Jia, J., Wan, Y., Zhou, Y., Kong, Y., Qian, Y., and Long, J. (2021). TransR *: Representation learning model by flexible translation and relation matrix projection. *Journal of Intelligent & Fuzzy Systems*, 40(5):10251–10259.

Zhao, W. and Liu, J. (2008). OWL/SWRL representation methodology for EXPRESS-driven product information model: Part I. Implementation methodology. *Computers in Industry*, 59(6):580–589.

Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., and Sun, M. (2020a). Graph Neural Networks: A Review of Methods and Applications. *AI Open*, 1:57–81.

Zhou, X., Carmeliet, J., Sulzer, M., and Derome, D. (2020b). Energy-efficient mitigation measures for improving indoor thermal comfort during heat waves. *Applied Energy*, 278.

Zhou, Y., Aryal, S., and Bouadjenek, M. R. (2024). Review for Handling Missing Data with special missing mechanism. *arXiv preprint arXiv:2404.04905*.

Appendix A

Source Code

A.1 Repository for the Project Source Code

github: `https://github.com/BIM-and-Automation-Laboratory/phd-source`

Appendix B

Datasets

B.1 Repository for the Project Datasets

github: `https://github.com/BIM-and-Automation-Laboratory/phd-datasets`