Data-Centric Engineering (2025), 6: e24
doi:10.1017/dce.2024.45

CAMBRIDGE UNIVERSITY PRESS

RESEARCH ARTICLE

On calculating structural similarity metrics in population-based structural health monitoring

Daniel S. Brennan, Timothy J. Rogers, Elizabeth J. Cross and Keith Worden

Dynamics Research Group, Department of Mechanical Engineering, University of Sheffield, Sheffield, UK
Corresponding author: Daniel S. Brennan; E-mail: d.s.brennan@sheffield.ac.uk

Received: 29 January 2024; **Revised:** 06 June 2024; **Accepted:** 20 August 2024

Keywords: Canonical form; graph-matching network; irreducible-element models; Jaccard index; population-based SHM

Abstract

The newly introduced discipline of *Population-Based Structural Health Monitoring* (PBSHM) has been developed in order to circumvent the issue of data scarcity in "classical" SHM. PBSHM does this by using data across an entire population, in order to improve diagnostics for a single data-poor structure. The improvement of inferences across populations uses the machine-learning technology of *transfer learning.* In order that transfer makes matters better, rather than worse, PBSHM assesses the similarity of structures and only transfers if a threshold of similarity is reached. The similarity measures are implemented by embedding structures as models —*Irreducible-Element* (IE) models— in a graph space. The problem with this approach is that the construction of IE models is subjective and can suffer from author-bias, which may induce dissimilarity where there is none. This paper proposes that IE-models be transformed to a *canonical form* through reduction rules, in which possible sources of ambiguity have been removed. Furthermore, in order that other variations —outside the control of the modeller— are correctly dealt with, the paper introduces the idea of a *reality model*, which encodes details of the environment and operation of the structure. Finally, the effects of the canonical form on similarity assessments are investigated via a numerical population study. A final novelty of the paper is in the implementation of a neural-network-based similarity measure, which learns reduction rules from data; the results with the new *graph-matching network* (GMN) are compared with a previous approach based on the *Jaccard index*, from pure graph theory.

Impact Statement

Data-based *Structural Health Monitoring* (SHM) has benefited from over three decades of research and offers an extremely promising means of automatically diagnosing damage in structures, thus improving operational safety and economy. Despite this effort, SHM has not made the transition to commonplace usage within the industry. One of the problems is that higher levels of diagnostics (damage location and quantification) require data from structures in damaged states, which are difficult or impossible to obtain. *Population-Based SHM* (PBSHM) has been proposed as a means of solving the problem of data scarcity by using data across entire populations of structures. Inferences in PBSHM are considerably strengthened if the population data are from similar structures. For this reason, a major part of the PBSHM framework involves assessing the similarity of structures, and this is accomplished by modeling structures in a graph space in which comparisons are facilitated mathematically. The comparison process itself introduces technical problems, not least the fact that structural models are subjective

This research paper was awarded Open Data and Open Materials badges for transparent practices. See the Data Availability Statement for details.

and affected by author bias. The current paper is a major contribution to the process of removing author bias and allowing objective structural comparison. As such, it is a step toward the practical implementation of PBSHM across the civil industry.

1. Introduction

In a traditional structural health monitoring (SHM) paradigm, data are acquired via a monitoring campaign on a single structure with the aim of determining the health of the given structure. However, this methodology has inherent obstacles; data must first be acquired from the structure of interest before any health state can be determined. Population-based structural health monitoring (PBSHM) attempts to overcome the aforementioned data obstacle within SHM by expanding the remit of available data sources. The belief is that by monitoring multiple structures —*the population*— knowledge on the health of a specific structure can be enhanced in comparison to the knowledge available when only utilizing its own data.

PBSHM operates under the premise that knowledge learnt on one structure may be transferred to another structure via a process of transfer learning (Pan and Yang, 2010; Weiss et al., 2016). To aid in the intention that any transferred knowledge improves rather than hinders the overall knowledge of the target structure: a structural similarity is first established between the source and target structures before any knowledge transfer is attempted.

The work included within this paper focuses on the portion of PBSHM, which establishes a degree of similarity between the structures. Before any metrics of similarity can be computed, there must first be a common domain in which to describe these structures: in this case, the set of *Irreducible Element* (IE) models. Brennan et al. (2025) introduced the second generation of IE model language and referred to the domain in which structures are compared within PBSHM as *the network.*

Previous work by Gosliga et al. (2021) established how the Jaccard Index (Jaccard, 1901) with a Maximum Common Subgraph (MCS) (Fernández and Valiente, 2001) approach can be used for generating a similarity metric between two structures. The problem with any similarity algorithm is in making the algorithm recognize differences in the models which are present because of underlying differences in the objects being represented (structures), and ignoring differences which are present because of limited human understanding of the problem (*author bias*).

The purpose of this paper is to explore the effect that variations within a model from author bias have on the network and the associated similarity metrics. This paper proposes a novel approach to dealing with these aforementioned variations by the introduction of a *Canonical Form* IE model, which provides a unique IE model for each structure, regardless of any author-introduced variations. An existing machine-learning technique is also proposed here as an alternative to the graph-theory approach utilized by Gosliga et al. (2021) to generate similarity metrics.

This paper is laid out as follows: Section 2 outlines the background of why variations are present within IE models because of author bias. Section 3 introduces the *Canonical Form* IE model and *Canonical Form Reduction Rules* (CFRR). The *Reality Model* is introduced, and the effect that it will have on the CFRR is discussed. The effect that the CFRR have on the network is evaluated using the Jaccard Index and MCS algorithm to generate a similarity matrix with and without using the CFRR. Section 4 introduces the *Graph Matching Network* (GMN) within the realm of PBSHM and explores the use of the GMN to generate similarity metrics using the *Canonical Form.* Finally, Section 5 provides the conclusions of this work.

2. Background

SHM (Farrar and Worden, 2007) aims to understand the health of a structure or system by analyzing sensor data from the structure. Over the decades, many approaches have been tried and tested with the vision of implementing SHM in the real world; however, there is one limitation that has plagued the field

https://doi.org/10.1017/dce.2024.45 Published online by Cambridge University Press

since its inception—the availability of damage labels for a given structure or system. PBSHM (Bull et al., 2021; Gosliga et al., 2021; Gardner et al., 2021; Tsialiamanis et al., 2021; Brennan et al., 2025) aims to address these inherent data availability issues by monitoring multiple structures with similar characteristics —*the population*— with the desire that data can be transferred from one similar structure to another.

These aforementioned goals of PBSHM can be broken down into two distinct subprocesses: determining which structures (*or components of the structure*) are similar —thus establishing a population— and transferring any learnt knowledge across the population. Before any similarity metrics can be drawn up for a given set of structures, there must be a standardized methodology for describing the structures in a consistent and meaningful manner. Gosliga et al. (2021) introduced the vehicle used within PBSHM to achieve such descriptions of structures: IE models.

The premise of an IE model is straightforward: create a representation of any structurally significant components within the structure and capture each interaction between these components. The level of detail required for an IE model is not the same as for a finite element (FE) model or for a computer-aided design (CAD) model; such levels of detail as to the complex geometrical mesh would only serve to hinder any similarity metric, instead of providing a grounding for an overall characteristic of the structure. The initial concept of IE model generation proved fruitful across two datasets: the initial toy dataset and a further real bridge dataset (Gosliga et al., 2022).

The recent work by Brennan et al. (2025) introduced the idea of a technological implementation supporting a shared-data domain in which PBSHM data reside; *network*, *framework*, and *database.* A second generation of IE models was also proposed, which facilitated an increased embedding of engineering knowledge and design choices within the model; in conjunction, it provided a standardized IE model language for structure descriptions, enabling IE model data to reside within the introduced shared-data domain. In the interest of completeness, a short recap of the second-generation IE model language is included here; however, the reader is recommended to read the original paper by Brennan et al. (2025) to gain a full understanding of the breadth and depth of rich engineering knowledge available for embedding within the second-generation IE model language.

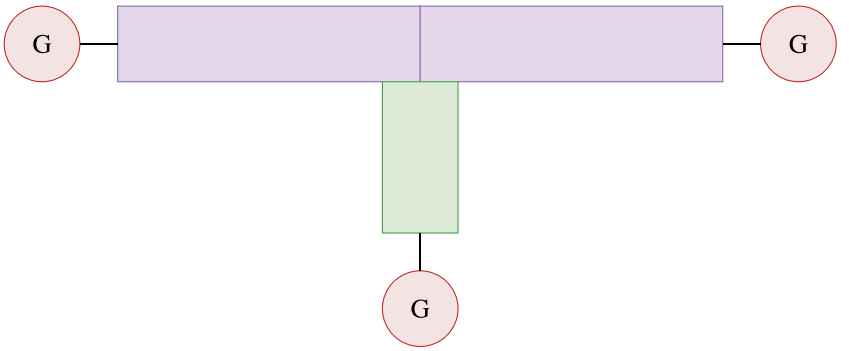
Any physical entity within a structure is labeled as an *element*; if the entity belongs to the structure in question, it is classified as a [regular] *element.* If the entity belongs to another structure, it is classified as a [ground] *element.* Any interactions between these *element*s are labeled as *relationship*s. Any time a larger [regular] *element* is divided into two or more [regular] *element*s of the same type, the interaction between those two [regular] *element*s is classified as a [perfect] *relationship.* When a [regular] *element* has been omitted from a model —because it is not classified as structurally significant— but interactions remain present which require to be modeled, this is classified as a [connection] *relationship.* If the physics between two [regular] *element*s is desired to be modeled, this interaction must be classified as a [joint] *relationship.* When the edge of the structure is to be denoted —when a [regular] *element* interacts with a [ground] *element*— this is modeled as a [boundary] *relationship.* Each type of *element* and *relationship* comes with its own set of accepted knowledge that is available to embed within the model.

Figure 1 depicts the change from a real-world structure to an IE model. The structure in question is a two-span beam-and-slab bridge from Northern Ireland (see Figure 1a), which —for the purposes of this paper— has been simplified into a single beam which runs horizontally from the left embankment to the right embankment —as pictured— and a single column supporting the horizontal beam, from the center of the beam to the road. Transitioning this scenario to an IE model (see Figure 1b) means that each embankment on the left and right side of the bridge is represented by independent [ground] *element*s. The beam traversing from left to right is represented by a single [regular] *element* and the column in the center of the beam providing vertical support, is represented by another [regular] *element.* One final [ground] *element* is also present to represent the ground on which the supporting column is resting. [boundary] *relationship*s are present between each [ground] *element* and the associated supported [regular] *element.* The interaction between the column and the beam are then modeled via a [joint] *relationship* with a [static] *nature.*

Within PBSHM, IE models may be the vehicle used to embed structural knowledge into the PBSHM framework; however, they are not the final domain in which this structural knowledge resides. The whole

https://doi.org/10.1017/dce.2024.45 Published online by Cambridge University Press

(a) A two-span beam-and-slab bridge example from Northern Ireland.



(b) A potential simplified Irreducible Element model representation of the two-span beam-and-slab bridge in Figure 1a. Each rectangle is a [regular] *element*. Each circle with a G inside, is a [ground] *element*.

Figure 1. *A simplified Irreducible-Element (IE) model representation of a two-span beam-and-slab bridge with two deck [regular] element s and one column [regular] element. The model interacts with the ground at the left and right side of the deck as well as at the bottom of the column and is considered a [grounded] IE model.*

purpose of embedding structural knowledge within *PBSHM* —and thus the necessity of IE models— is to facilitate the comparison of structures to collect a measured score of similarity between structures for determining potentially unknown *populations.* Brennan et al. (2025) refer to this final destination of structures as the *network*; a shared domain in which the similarity comparisons of PBSHM structures reside *and* —based upon the associated similarity— establish the strength of relationships between these structures. The implementation of these similarity algorithms will be present within the PBSHM framework, and as such, may support multiple different similarity algorithms that execute within the *network.* Each structure within the *network* will have a similarity score for every other structure, potentially for each supported similarity algorithm within the *framework.*

This affiliation of relationships between structures within PBSHM can be envisioned as a complete weighted graph, where each node is the model of a structure, and each edge is the similarity value between the two structures. Figure 2 visualizes the relationships between structures within the *network.* As the *network* is the final domain for IE model data, it is only natural that the field of graph theory (Barabási and Pósfai, 2016; Newman, 2018) be an avenue for exploration in the goal of determining the similarity of structures. IE models by their definition, naturally lend themselves to be represented as an Attributed Graph (AG): each *element* becomes a node, and each *relationship* becomes an edge. All the knowledge present within the IE model is then embedded as attributes on the corresponding node or edge.

Whilst PBSHM is a relatively recent branch of SHM, it does not invalidate the fundamentals upon which SHM was built and must honor these principles and practises within the theory of PBSHM. One of these aforementioned principles within SHM is the desire to locate where potential damage is located

https://doi.org/10.1017/dce.2024.45 Published online by Cambridge University Press

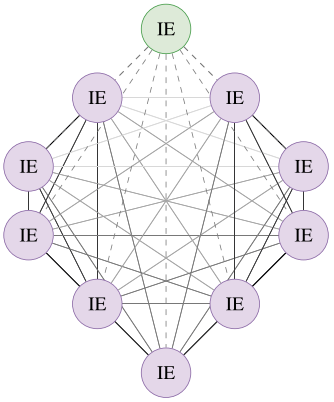


Figure 2. *A diagram of the similarity score-driven relationships between Irreducible Element (IE) models within the PBSHM network. Each existing IE model —a purple node— within the network, has a relationship with every other IE model within the network. Each relationship is derived from a similarity score generated by the PBSHM framework, and as such, may therefore necessitate multiple relationships between each pair of IE models in the network. The diagram also depicts a new IE model —the green node — being added to the network, and the process of relationships being discovered between the newly inserted IE model and existing network models.*

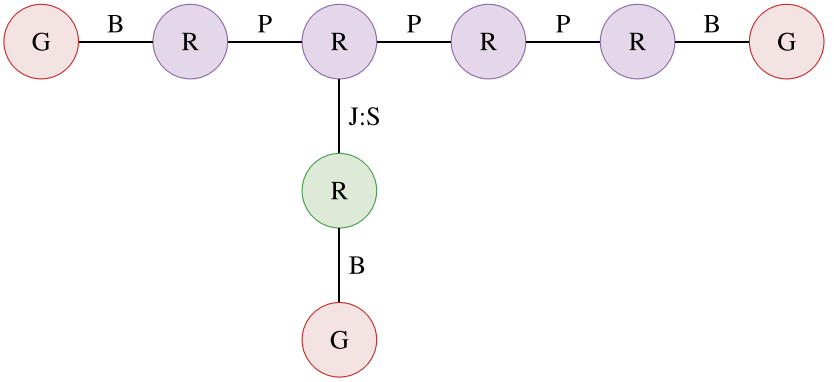
within the structure. The issue with honoring this principle is *subjectivity.* If one considers the two-span beam-and-slab bridge depicted in Figure 1a. One engineer may be particularly interested in locating the damage on the beam, and as such, add more details within the model on the beam section of the IE model. Another engineer may decide that the damage on the column is of paramount importance and thus add additional details to the column section of the IE model. These nuances in model objectives may appear insignificant within the grand scheme of PBSHM; however, they can vastly change the arrangement of an IE model and thus the associated AG.

Figure 3 illustrates how the subjectivity of the model creator can change the underlying model submitted into the *database* and ultimately the *network.* The first graph (see Figure 3a) shows the changes present within the IE model if the author decided that instead of the horizontal beam being a single [regular] *element*, the horizontal beam is initially split into two [regular] *element*s to locate damage to a particular span of the bridge, the right span of the bridge is further subdivided into three [regular] *element*s for either sensor placement or potential further damage localization given signs of wear on that span of the bridge. The second graph (see Figure 3b) shows that the horizontal beam has been left as a single [regular] *element*; however, the vertical column has been split into two [regular] *element*s to enable damage localization to either the top section of the column or the bottom section of the column. The third and final graph (see Figure 3c) splits the horizontal beam into two [regular] *element*s and a single [regular] *element* for the column; however, the engineer generating this IE model has determined that there should be a [joint] *relationship* to either span of the horizontal beam. These are only three of the potentially limitless variations that can be present in the simplified two-span beam-and-slab bridge.

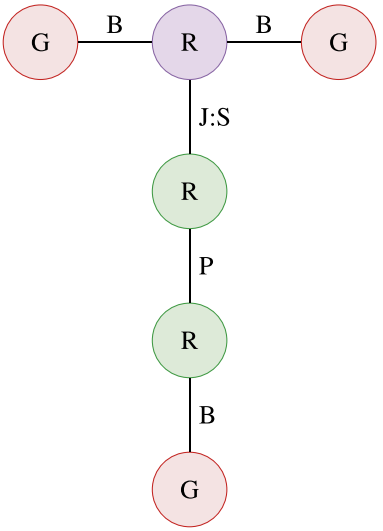
Variations present within a model because of author subjectivity are a fundamental issue with any modeling task. The problem was present in the initial version of IE models by Gosliga et al. (2021) and remains present in the second version of IE model language by Brennan et al. (2025); however, with the second version of the language, there is embedded knowledge stored within the model itself to help understand and interpolate why an author has chosen to dissect the structure in the manner present within the model. Research has already been initiated by Gosliga et al. (2021) into the viability of the Jaccard Index as a similarity metric within PBSHM. The Jaccard Index works by calculating the MCS between two graphs; in the case here, two attributed graphs.

To evaluate the impact these aforementioned variations have upon PBSHM's similarity results, a synthetic dataset was generated based on the simplified two-span beam-and-slab bridge example

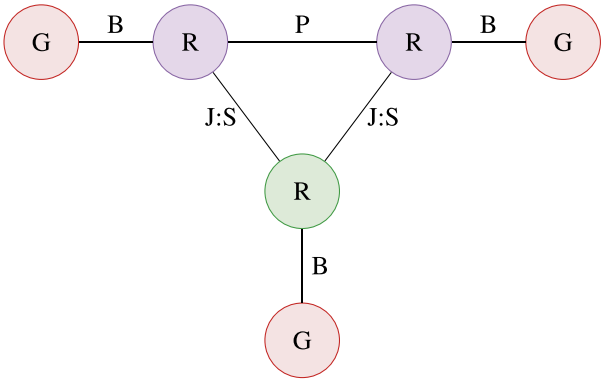
https://doi.org/10.1017/dce.2024.45 Published online by Cambridge University Press



(a) This IE model variation has been modelled with the deck as two separate [regular] *elements* with the column component only interacting with the right section of the deck. The right deck has also been subdivided into three [regular] *element*s to aid in damage localisation on that section of the deck.



(b) This IE model variation has been modelled with the deck as only a single [regular] *element*, whilst the column component has been divided into two [regular] *element*s for damage localisation within the column.

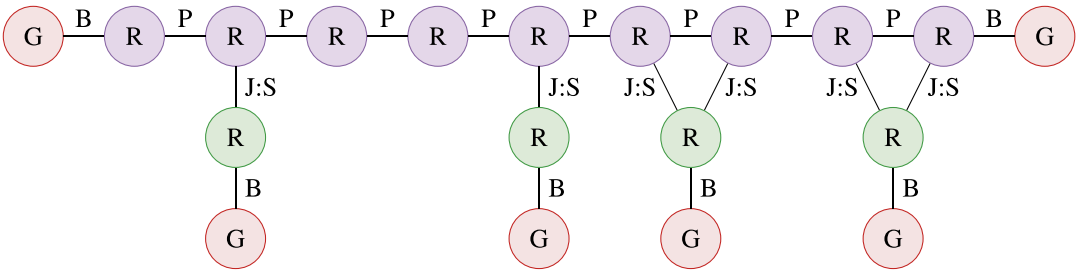


(c) This IE model variation has the deck split into two [regular] *element*s, with the column section being modelled as a single [regular] *element*; however, the interaction between the column and the deck has been modelled as interacting with both sections of the deck.

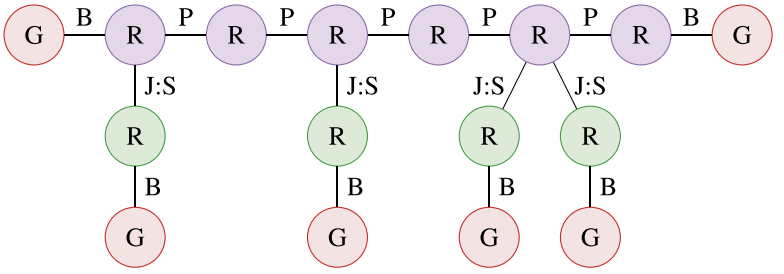
Figure 3. *Three of the potential Irreducible Element (IE) model representations —displayed as graphs— of the two-span bridge displayed in Figure 1a. [ground] element s are represented by a G in the centre of the node, [regular] element s are represented by an R in the centre of the node. [boundary] relationship s are represented by a B on the edge, [perfect] relationship s are represented by a P on the edge and a [joint] relationship with a [static] nature is represented by a J:S on the edge.*

illustrated within this paper. The dataset contains randomly generated beam-and-slab bridges from two to ten spans, with each span being potentially divided up into three subsections; furthermore, each column between the span was either joined to the previous span, the next span, or both spans to include the full set of variations presented in Figure 3. The dataset contains a total of 4500 randomly generated bridges —500

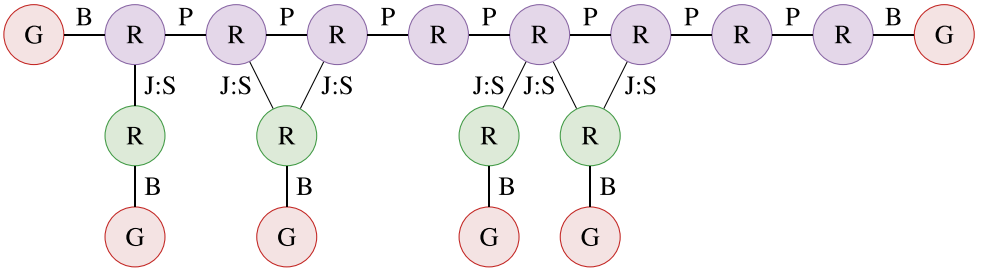
https://doi.org/10.1017/dce.2024.45 Published online by Cambridge University Press



(a) Graph ID #4-208



(b) Graph ID #4-329



(c) Graph ID #4-354

Figure 4. *An extract of the Irreducible Element (IE) models —displayed as graphs— contained within the generated beam-and-slab 'matching' dataset. Each IE model incorporates the following variations: spans being divided up into one to three subsections and each column being joined to either the previous span, the next span, or both spans. The examples chosen are from the test subset and are used in the similarity results in Figures 10,13, and 14.*

bridges per number of spans— and was then randomly separated into a training, validation, and test subsets. This dataset —for the purposes of this paper— will be henceforth known as, the *matching* dataset. Figure 4 displays an extract of the generated "five span" bridges included within the *matching* dataset.

To ensure consistency throughout the similarity matrix results depicted within this paper, the embedding of attributes into the AG representation from an IE model has been fixed to embedding only the *contextual* type —the ***type*** attribute value from the *contextual* object within a [regular] *element.* For nodes where there is no *contextual* type —such as a [ground] *element*— no attributes are embedded into the node. The edges in the AG representations have no attributes from the associated [relationship]s embedded within the graph.

Figure 5 shows the results of embedding only the [regular] *element*'s *contextual* type within the AG and evaluating each pair within the *network* for their given MCS similarity using the Jaccard Index. The axes of the similarity matrix are labeled with the number of spans of the bridge and their associated graph

https://doi.org/10.1017/dce.2024.45 Published online by Cambridge University Press

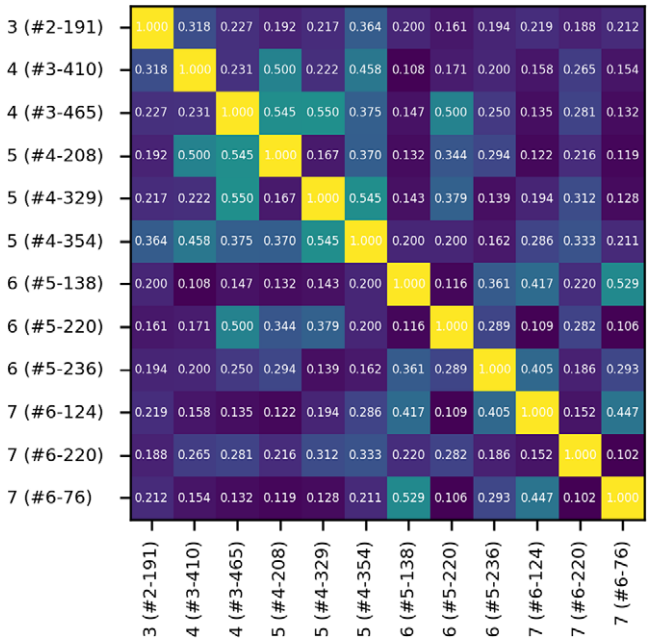


Figure 5. *The Jaccard Index similarity matrix results from the Maximum Common Subgraph on the test portion of the matching dataset when embedding only the contextual type as the node attribute. The axis are labelled with the number of spans the graph is associated with and the ID of the graph from within the dataset.*

number within the matching dataset. In the ideal scenario, all the graphs with the same number of spans should all identify as matching with a similarity value of 1. When a graph with either a descending or ascending number of spans —$N-1$, $N+1$— is compared to a graph with N number of spans, the similarity score should identify these as the next closest match, after N.

As the reader can see in the results in Figure 5, when the inherent ambiguity of the model author's subjectivity is included within the graphs, the algorithm is not able to find any strong recognizable pattern. The algorithm correctly identifies when the graph is compared to itself; however, the algorithm —at least within the matching dataset— is not able to correctly identify graphs with the same number of spans as identical; instead, it identifies graphs with differing number of spans as being the closest matches. If one looks at the result for 6 (#5–220), the algorithm identifies a four-span bridge (#3–465) as having a closer similarity than any of the six-span bridges.

3. Canonical Form

The observed variations present within the similarity metrics —when introducing the inherent model subjectivity— highlight two new scenarios which require attention within the comparison portion of PBSHM. When generating similarity scores, two graphs must always match as identical if the source structure from which both graphs have been generated is the same structure, and structures which are classed as nominally identical or from a homogenous population Bull et al. (2021), should further match as identical.

This paper proposes that the solution for addressing the aforementioned scenarios across all current and future similarity algorithms within the *framework* is a methodology for reducing IE models to a common form. A form which preserves the structural knowledge and engineering decisions present within the original model but facilitates a common representation of a single structure, regardless of any author subjectivity; a *Canonical Form.* IE models generated by authors would henceforth be known as *detailed*

https://doi.org/10.1017/dce.2024.45 Published online by Cambridge University Press

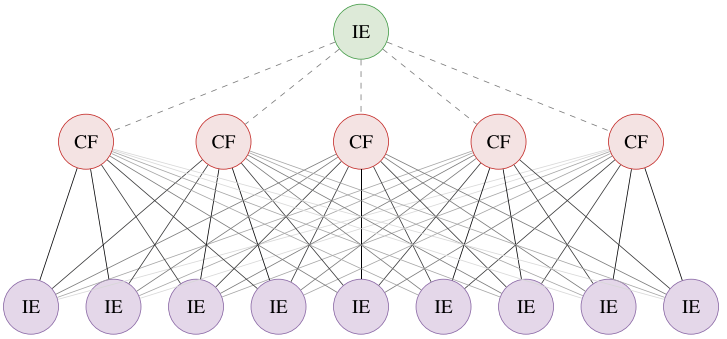


Figure 6. *The PBSHM Network using the Canonical Form as a common form for comparison. The red nodes represent the known Canonical Form representations within the network. The purple nodes represent existing detailed IE models, for whom similarity comparison values are already present against the known Canonical Form representations. The weight of the similarity between the existing detailed IE models and the known Canonical Form representations are represented by increased darkness of colour on the edge —higher similarity scores equal darker edges. The green node represents a new detailed IE model being inserted into the network and the dotted edges represent the similarity calculations made upon insertion.*

IE models and only reduced to a *Canonical Form* representation for the purpose of similarity matching within the *network. Detailed* IE models would still be submitted by authors into the *framework* and ultimately stored within the *database.*

Furthermore, the notion of a common form for a single structure has the potential to improve the performance of the *network.* Currently, the *network* acts as a complete weighted graph for each similarity algorithm within the *framework.* Computationally, this requirement involves each unique pair of graphs having their similarity computed. Whilst this mechanism may appear trivial when factoring a toy dataset numbering only a few hundred graphs, the logistics of performing this same computation become problematic when considering the potentially vast size and quantity of real-world structure graphs. The largest single graph within the matching dataset has in the order of tens of *element*s/nodes. Real-world structures may have *element*s numbering in the order of hundreds or even thousands within a single structure. If one factors in that, within a *network*, one may have thousands, if not tens of thousands of structures present, the reality of performing these computations becomes expensive — without factoring in the possible variations from model subjectivity.

This paper proposes that the solution to the computational problem is that the *Canonical Form* becomes an intermediary layer within the *network* to act as a known target for comparison against *detailed* IE models. Each *detailed* IE model within the *network* would have a similarity score to every *Canonical Form* within the *network.* When a new *detailed* IE model is inserted into the *network*, only similarity scores are drawn up for the newly inserted model and the existing known *Canonical Form*s. The proposed modified methodology of the *network* has the potential to not only reduce the number of computations performed within the *network* but also create a natural alignment of populations within the *network* for discovery by clustering algorithms. Figure 6 visualizes the configuration of the *Canonical Form*-inspired *network* and depicts the process of a new *detailed* IE model being included into the *network.*

3.1. Canonical Form reduction rules

To facilitate the process of reducing a *detailed* IE model to the corresponding *Canonical Form* representation, this paper proposes an initial set of three reduction rules to accomplish the desired common form; the CFRRs. The CFRR are a set of rules which can be applied to any *detailed* IE model, with the goal of removing any ambiguity from the model; however, the rules must not remove any

https://doi.org/10.1017/dce.2024.45 Published online by Cambridge University Press

embedded knowledge within the model, which may later be used within the similarity metrics. Each rule must be grounded in a solid reasoning as to why the associated modifications contained within the rule are able to modify the IE model, without the loss of knowledge from within the model. While the reduction rules may be applied whilst the model is within the IE model domain, the reality is that the *Canonical Form* representation occurs whilst within the *network* and as such, further discussions within this paper will refer to the changes made by the CFRR as within the associated graph domain of the *network.*

The upcoming subsections introduce the first three CFRR. Each reduction rule has an accompanying figure which depicts the methodology of its operations within the graph domain. To ensure continuity throughout this paper, the associated figures utilize the two-span beam-and-slab bridge example introduced in Figure 1. It is important to note that the generated *Canonical Form* representation of a *detailed* IE model is not intended to be immutable. When additional knowledge is accumulated on the attributes and topology that are important within the *network*, new CFRR will be included; therefore, transforming the representation referenced as the *Canonical Form.*

3.1.1. Individual ground

The first rule proposed within the CFRR is that each [ground] *element* within a graph must be unique. This rule requires that wherever a [boundary] *relationship* is present within a graph, the associated [ground] *element* included within the *relationship* must be unique to the [boundary] *relationship* and not shared with any other [boundary] *relationship*s. The reasoning behind this rule is, each [ground] *element* present within the graph, is the representation of another structure's presence within the model. Each interaction between the structure being modeled, and the third-party structure is unique, and as such, should be represented as a unique [ground] *element* within the model. As a [ground] *element* is only the reference to the presence of an external structure, no knowledge is lost by this reduction rule.

The *Individual Ground* reduction rule not only reduces the topological complexity of the graph by removing unnecessary loops but could also be applied as a general rule for [ground] *element*s in the *detailed* type. Figure 7 illustrates the process of selecting a [ground] *element* with more than one corresponding [boundary] *relationship*, creating new [ground] *element*s and [boundary] *relationship*s, and subsequently, removing the offending [ground] *element* and [boundary] *relationship*s.

3.1.2. Perfect-Joint-Joint relationships

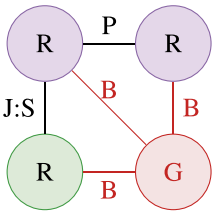
The second rule proposed within the CFRR is that any time within the graph where there is a pattern of three [regular] *element*s connected in a loop via a [perfect], [joint], and [joint] *relationship*; the loop can be broken and reduced to a [perfect] and [joint] *relationship.* If one takes the example illustrated in Figure 3c, there are two [regular] *element*s —representing the horizontal beam in the example bridge— connected via a [perfect] *relationship*, there is then a single [regular] element —representing the vertical support column— connected to both of the aforementioned [regular] *element*s of the horizontal beam, via independent [joint] *relationship*s.

The interaction between the three aforementioned [regular] *element*s can be modeled in three distinct manners: the vertical support column is connected via a [joint] *relationship* to both of the horizontal beam [regular] *elements* (as depicted within Figure 3c), the vertical support column is connected via a [joint] *relationship* to only the left horizontal beam [regular] *element*, and oppositely, the vertical support column is connected via a [joint] *relationship* to only the right horizontal beam [regular] *element.*

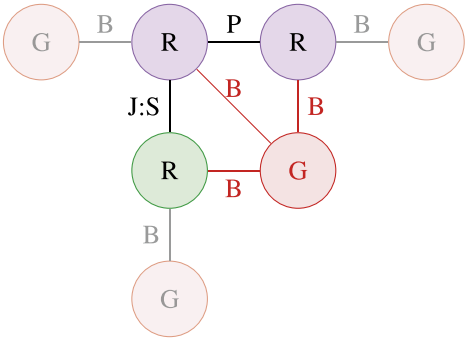
Each of these scenarios is a valid method for embedding the structural knowledge of the interaction between the horizontal beam and the vertical support column. In the last two scenarios, the physics between the vertical support column and the horizontal beam have been embedded once within the model; conversely, in the first scenario, the physics have been embedded twice within the model, once for each beam.

The *Perfect-Joint-Joint* reduction rule can safely reduce a [perfect], [joint], [joint] *relationship* loop to a single [perfect] and [joint] *relationship* as the physics of the interaction have been duplicated within the model; thus, one of the [joint] *relationship*s can safely be removed from the model without losing any structure knowledge regarding the interaction. The *Perfect-Joint-Joint* reduction rule also simplifies the topology of the graph by removing another unnecessary loop.

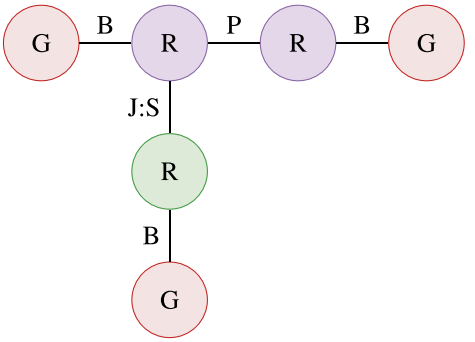
https://doi.org/10.1017/dce.2024.45 Published online by Cambridge University Press



(a) Step 1: Select any [ground] *element* where two or more [boundary] *relationship*s are present. The pattern is highlighted in red.



(b) Step 2: Create a new [ground] *element* and [boundary] *relationship* for each [boundary] *relationship* present within the previously-selected [ground] *element*. The creation of new *object*s, must retain within the target *object* (i.e. the newly-created *object*) any attributes present within the source *object* (i.e. the *object* which is being copied).



(c) Step 3: Remove the previously-selected common [ground] *element* from the model.

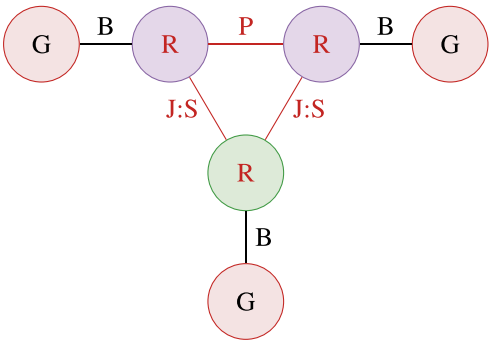
Figure 7. *The stages of a Individual Ground Canonical Form reduction against an Irreducible Element model graph. By performing this reduction, an unrequired loop is removed from the graph without the loss of any embedded knowledge within the model.*

Figure 8 illustrates the process of finding the [perfect], [joint], [joint] *relationship* loop, selecting one of the [joint] *relationship*s to remove, and finally removing the selected [joint] *relationship* from the graph. Whilst the *Perfect-Joint-Joint* reduction rule does not enforce which of the [joint] *relationship*s should be removed from the graph, any implementation of the *Perfect-Joint-Joint* reduction rule must be consistent in which [joint] *relationship* the algorithm decides to remove; if the same graph is reduced by a CFRR implementation, it must choose the same [joint] *relationship* to remove, each and every time i.e in a planar graph, an implementation may choose to consistently remove the "right" joint.

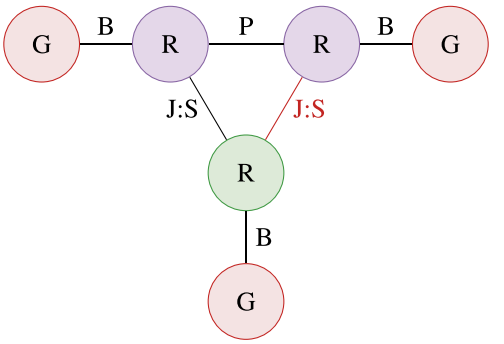
3.1.3. Perfect relationships

The third rule proposed within the CFRR for now, is that any [regular] *element*, with *exactly* two [perfect] *relationship*s may be removed from the graph with the associated [perfect] *relationship*s, by creating a new [perfect] *relationship* between the neighboring [regular] *elements* and migrating any knowledge within the [regular] *element* to be removed, to the neighboring [regular] *element*s.

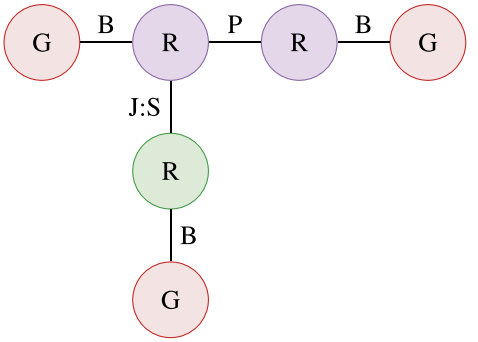
https://doi.org/10.1017/dce.2024.45 Published online by Cambridge University Press



(a) Step 1: Select any loop within the graph, constituted of three [regular] *element*s, connected via a [perfect], [joint] and [joint] *relationship*. The loop pattern is highlighted in red.



(b) Step 2: Select one of the [joint] *relationship*s within the loop. The selection between which of the [joint] *relationship*s within the loop is irrelevant; however, any implementation of the Perfect Joint Joint reduction, must be consistent in its [joint] *relationship* selection, as to enforce an equivalent operation across the PBSHM network.



(c) Step 3: Remove the previously selected [joint] *relationship* from the model.

Figure 8. *The stages of a Perfect-Joint-Joint Canonical Form reduction against an Irreducible Element model graph. By performing this reduction, an unrequired loop is removed from the graph without the loss of any embedded knowledge within the model.*

[Perfect] *relationship*s by their own definition, are present within an IE model where a larger component has been divided up into additional [regular] *element*s, either for representing a complex geometrical shape, or for the purpose of damage localization within the model. In both of the aforementioned scenarios, the [perfect] [relationship] is only present within the model to handle model subjectivity or SHM necessity of the creator. Embedding complex geometrical shapes is important to gain advanced knowledge of the form of a component; however, such detailed knowledge is potentially irrelevant when trying to compare the overall similarity of two structures, but becomes increasingly relevant when trying to compare the similarity of structure subsections or validate the comparisons to a third party. The same premise holds true for the division which has occurred because of damage localization: knowledge on where damage has transpired within the model is vitally important for the author of the model, or when trying to relay knowledge back to the owner or operator; however, these details are irrelevant when determining the similarity of structures.

https://doi.org/10.1017/dce.2024.45 Published online by Cambridge University Press

The *Perfect Relationship* reduction rule can safely reduce a [regular] *element* with two —and only two — [perfect] *relationship*s, as the knowledge contained within the selected [perfect] *relationship*s and associated [regular] *element* is irrelevant for similarity purposes, and can be merged into neighboring [regular] *element*s without losing any structural relevant knowledge in the context of the *network.*

Figure 9 illustrates the process of finding the [regular] *element* with two —and only two— [perfect] *relationship*s, creating a new [perfect] *relationship* between the neighboring [regular] *element*s of the selected [regular] *element*, and removing the original selected [regular] *element* and the associated redundant [perfect] *relationship*s from the graph. Whilst the *Perfect Relationship* reduction rule does not explicitly enforce that neighboring [regular] *element*s must obey the [perfect] *relationship* matching type rule defined by Brennan et al. (2025), it is expected that any implementation of the CFRR would ensure that the neighboring [regular] *element*s of the selected [regular] *element* have matching values for the *contextual*, *geometrical* and *material* types before actioning the defined reduction rule.

It is envisioned that the [perfect] *relationship* reduction rule will be refined in the future to handle [regular] *element*s that have more than two [perfect] *relationship*s. In the fullness of time, the CFRR will have additional rules included to facilitate the removal of all unrequired variations within the *network.* In the final version of the CFRR, there will be no [perfect] *relationship*s present in a CF IE model; however, this statement will not be valid within the remit of the *Reality Model* (see Section 3.3).

3.2. Jaccard Index results

As discussed earlier in the paper (see Section 1), the Jaccard Index —or Jaccard similarity coefficient— is a method for measuring the similarity between two datasets. In the case of determining the similarity of IE models, the algorithm was used by Gosliga et al. (2021) and Gosliga et al. (2022), to generate a similarity score between two attributed graphs (see Figure 10). The logic behind the Jaccard Index boils down to calculating the intersection between G_1 and G_2, over the union of G_1 and G_2:

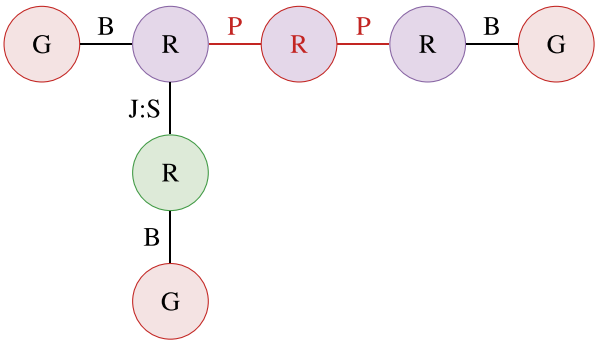
$$p(A,B) = \frac{|A| \cap |B|}{|A| \cup |B|} = \frac{|A| \cap |B|}{|A| + |B| - |A| \cap |B|} \tag{1}$$

The output from the Jaccard Index is a similarity score between 0 and 1, where 1 is similar and 0 is dissimilar. The calculation of the MCS between G_1 and G_2 is implemented via a backtracking algorithm to find the largest common subgraph between two graphs — G_1 and G_2 in this case. In the interest of brevity, the logic of implementing the backtracking algorithm is excluded from this paper, the interested reader is recommended to read the original paper by Gosliga et al. (2021) to understand the finer workings of the algorithm.

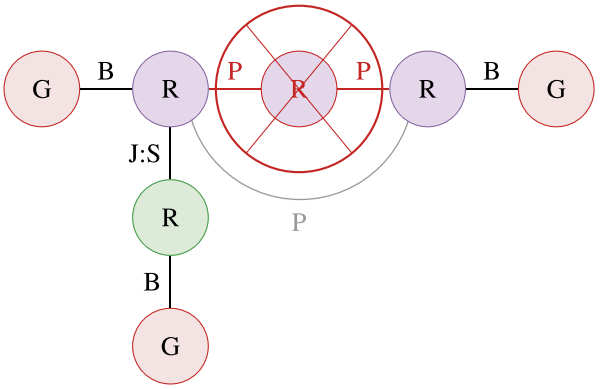
Figure 10 displays the similarity matrix results using the Jaccard similarity coefficient against the matching dataset used in Figure 5 and the known *Canonical Form* dataset for bridges with spans from 3 to 7. The ideal scenario for these similarity metrics is that a bridge from the matching dataset should match as near identical —a value as close to 1 as possible— to the known *Canonical Form* bridge with the same number of spans. The similarity value should decrease in value the further away the number of spans being compared.

The first results in Figure 10a show the similarities when using none of the CFRRs and instead using the *Canonical Form* as a common form for comparison against. As the reader can evaluate, the Jaccard similarity coefficient is unable to find any discernible pattern between the matching dataset and the *Canonical Form* dataset. The second results in Figure 10b show the similarities when the matching dataset —containing *detailed* IE models— has first been reduced using an implementation of the CFRR before being evaluated against the common form *Canonical Form* dataset, using the Jaccard similarity coefficient. As the reader can see, the implementation of the CFRR within the *network* improves the indicated values with the desired pattern of similarity (results within the same span should match identically with similarity values gradually decreasing through the change in number of spans) starting to emerge when comparing the *matching* dataset to the *Canonical Form* dataset.

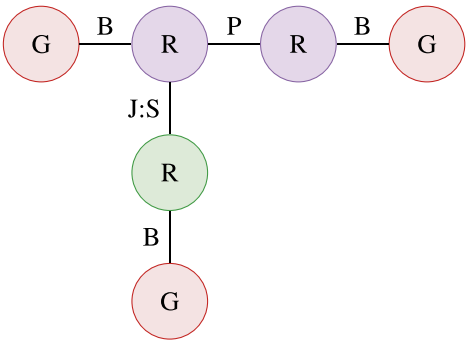
https://doi.org/10.1017/dce.2024.45 Published online by Cambridge University Press



(a) Step 1: Select any [regular] *relationship* within the graph that has two — and only two — [perfect] *relationship*s. The pattern is highlighted in red.



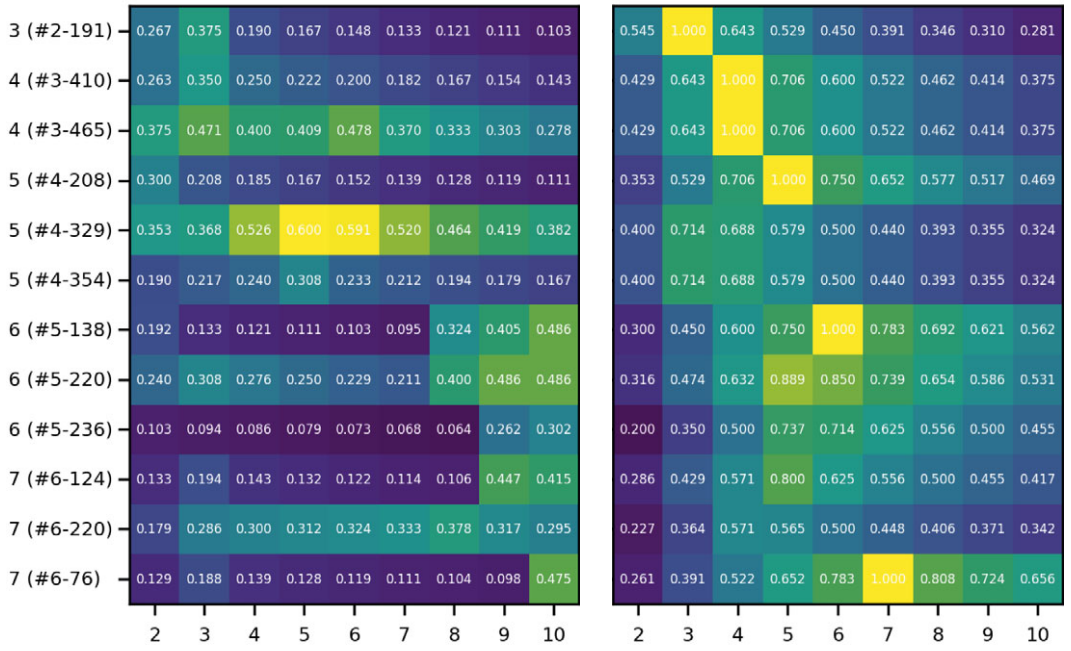
(b) Step 2: Isolate the previously selected [regular] *element* and two [perfect] *relationship*s from the graph. Create a new [perfect] *relationship* between the two neighbouring [regular] *element*s from the isolated [regular] *element*.



(c) Step 3: Remove the previous-selected [regular] *element* and two [perfect] *relationship*s from the model.

Figure 9. *The stages of a Perfect-Perfect Canonical Form reduction against an Irreducible Element model graph. By performing this reduction, an unrequired node is removed from the graph without the loss of any embedded knowledge required for similarity matching. Iterating over the graph with this reduction rule until no further regular elements are removed will remove the unrequired sequences of repeated [regular] elements and [perfect] relationships from the graph.*

https://doi.org/10.1017/dce.2024.45 Published online by Cambridge University Press



(a) The Jaccard Index similarity matrix results using the *detailed* Irreducible Element model without the *Canonical Form Reduction Rules*.

(b) The Jaccard Index similarity matrix results using the *Canonical Form Reduction Rules* to reduce the *detailed* Irreducible Element model before comparison.

Figure 10. *The Jaccard Index similarity matrix when comparing the matching dataset to the known Canonical Form dataset using both the Jaccard Index without the Canonical Form Reduction Rules (10a) and then with the Canonical Form Reduction Rules (10b). The Attributed Graph contains only the embedding of the [regular] element 's contextual type as a node attribute to keep results in direct comparison to Figure 13. The X axes are labelled with the number of spans of the Canonical Form graph, the Y axes are labelled with the number of spans the graph is associated with and the IE of the graph from the matching dataset. The label for the Y axis is missing from the second figure because the labels are the same as in the first figure.*

3.3. Reality Model

An IE model is only concerned with structural composition. The environment in which the IE model is placed, the operational constraints of the structure, and the concerns of a structure owner are but three examples of knowledge that, while being vitally important in the overall makeup of a structures' health, are out of the remit for an IE model. The aforementioned missing knowledge provides critical context to conditions a structure must endure; as such, they are required to be included within the global scope of PBSHM, whilst still remaining out of bounds to the structural comparisons portion of the PBSHM architecture.

A new model is required to capture the circumstances in which a structure resides, an encapsulation of the world in which the structure lives: the *Reality Model.* This model does not invalidate any of the proceeding research on capturing the structural composition of a model or any of the defined shared-data domains: *network*, *framework* and *database.* Instead, the model builds upon and encapsulates all of these PBSHM-defined fundamentals into a hierarchical overarching model. A *Reality Model* by itself will not be an official specification or list of requirements akin to the specification and language of an IE model: instead, the model will be the summation of all available knowledge on a structure: structural

https://doi.org/10.1017/dce.2024.45 Published online by Cambridge University Press

composition, channel values, extracted features, sensor network, environmental and operational variables, and damage localization concerns, to name but a few. Figure 11 depicts the potential hierarchical knowledge areas within the *Reality Model.*

The specifications and definitions of required knowledge will be devolved to the individual areas of knowledge. The decisions as to what is required to capture structural composition, belong to an IE model and as such are controlled by the IE model section within the PBSHM Schema. The decisions as what is required to ensure a full picture of a sensor network, belong to the sensor network and as such, will be defined by a future sensor-network section within the PBSHM Schema. It is only when the aforementioned knowledge areas are brought together, that the *Reality Model* achieves its full identity and has a powerful and meaningful purpose within PBSHM.

By the definition of a *Reality Model*, each knowledge area is devolved and has complete control of the associated data and language required to embed the associated required knowledge. The PBSHM shared data domain — *network*, *framework*, and database — must be aware of the *Reality Model* and understand how the presence of the model determines any confounding influences. The *database* will naturally become aware of any influences the *Reality Model* produces by the expansion of new defined knowledge areas within the PBSHM Schema. The *framework* will further organically expand to be *Reality Model*-aware, via the inclusion of new algorithms designed to process the enhanced available state of a structure contained within the *database.*

While the *network* operates its comparisons within the IE model domain, being *Reality Model*-aware means that additional restrictions may be required when considering the introduced *Canonical Form.* The whole purpose of an IE model and subsequently the *Canonical Form* is to find similarities between structures, thus enabling new populations of structures to be established, and finally, learnt knowledge being transferred across the population. There is no point in attempting to transfer learnt knowledge across the population, if the knowledge being transferred is not applicable to the target structure because of the world in which the structure lives.

As such, each area of knowledge encompassed within the *Reality Model* must have the potential to restrict and inform the produced *Canonical Form* representation of a structure. This may be by the

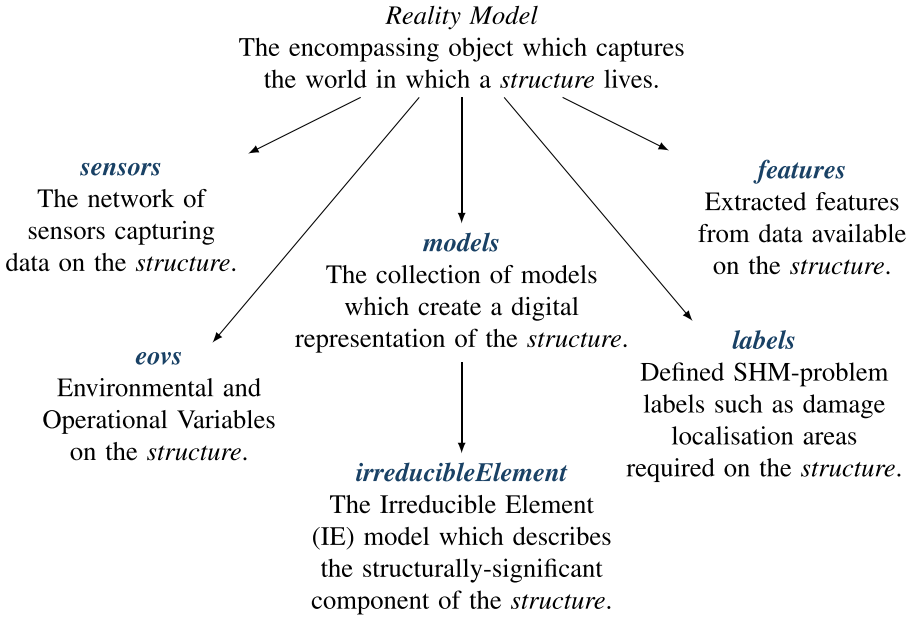


Figure 11. *A selection of potential knowledge areas included within the hierarchical layout of the Reality Model.*

https://doi.org/10.1017/dce.2024.45 Published online by Cambridge University Press

introduction of additional *CFRRs*, which are only pertinent if certain data are present within the *Reality Model.* They may also be in the form of restrictions on when certain *CFRRs* can be applied. A specific value within the *labels* section of the *Reality Model* may dictate that certain *element*s are protected and may not be removed from the model via the *CFRRs.* In a *network*, where only IE model data resides, the *Canonical Form* representation of two homogenous structures should be identical; however, when additional *Reality Model* data are included within the *network*, the *Canonical Form* representations of two homogenous structures, may no longer be identical.

4. Graph matching network

The Jaccard Index is simply one methodology for generating a similarity between two sets of graph data, once one has established the known intersection between these two sets of data. The way in which this intersection has been found previously —within the context of a graph— is by using the *Maximum Common Subgraph* (MCS). The MCS is an object from graph theory (Barabási and Pósfai, 2016; Newman, 2018) and is the result of finding the largest shared graph between two graphs (see Figure 12). The problem with this approach, is that each node within G_1 and G_2, have to match exactly.

If one takes the example of *material* within a [regular] *element*, say a beam on a bridge. Both bridges are classified as two-span beam-and-slab bridges; however, in the first bridge, the beam has a *material* type set of "metal" → "ferrousAlloy" → "steel," in the second bridge, the beam has a *material* type set to 'metal' → "aluminiumAlloy". No matter how this *material* knowledge is encoded into an attributed graph, the nodes of the corresponding [regular] *element*s would never be included within the MCS, without a decision to omit knowledge from the AG. To facilitate the inclusion of these nodes within the MCS, a decision would have to be made to only include the first level of *material* type within each node. Such modifications to knowledge encoding within the AG necessitate knowledge of both the context in

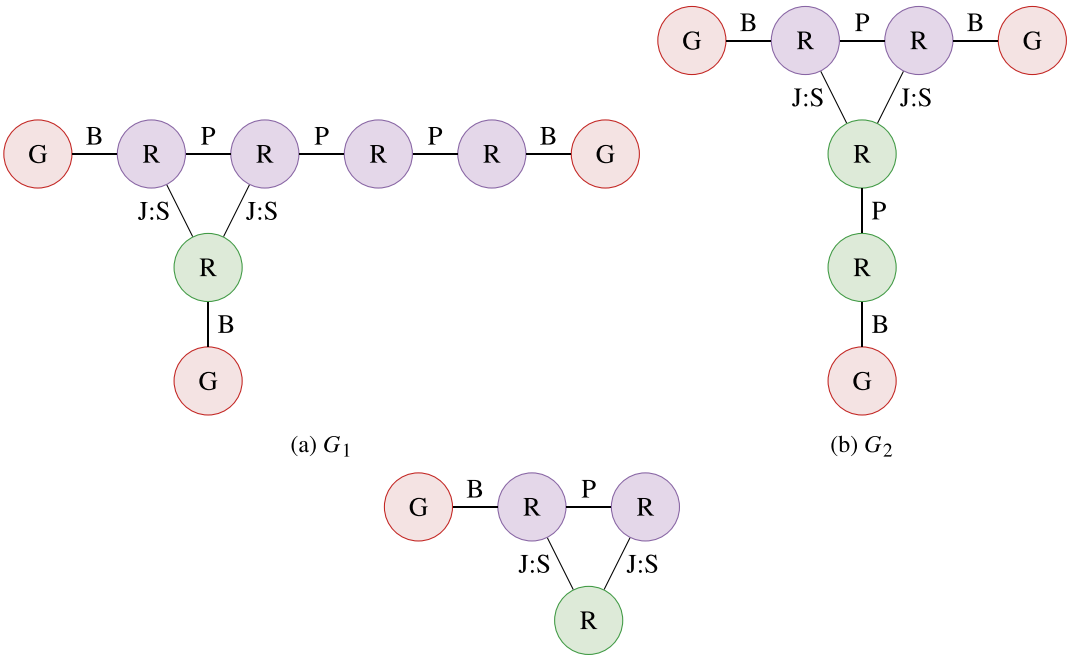


Figure 12. *The Maximum Common Subgraph (MCS) between G_1 and G_2, where the graphs are two bridge IE models with the contextual type from the [regular] element embedded as an attribute within the associated nodes.*

https://doi.org/10.1017/dce.2024.45 Published online by Cambridge University Press

which the structures are based and the mechanics of the similarity metrics. Alternately, a method in which all available knowledge from the IE model can be encoded within the attributed graph, and then the similarity algorithm itself can determine which of these attributes are necessary for determining the similarity of the *network.*

Neural network (Bishop, 2010) models are a subset of machine-learning paradigms aiming to replicate how the neurons inside the brain process and pass data between themselves. If one examines the process of how a multi-layer perceptron (MLP) (Bishop, 2010) approximates an input–output mapping for classification or regression. The MLP adopts a layered structure, with each layer receiving a vector of real numbers from the previous layer and passing on a processed vector to the next layer. The input layer receives training or test data from the outside world, and the output layer communicates final results to the outside world. All other layers are termed hidden layers. It is assumed here that the reader has some familiarity with the basic MLP structure.

In contrast to MLPs, graph neural networks (GNNs) (Bacciu et al., 2020) receive graph-structured objects at their input and produce them at their output (Tsialiamanis et al., 2021). The graphs of interest will generally be attributed graphs where the nodes and edges each carry an associated vector of parameters. Training then corresponds to optimizing these parameters to satisfy some purpose, and the graph topology itself will be unchanged at the output. In this case, the training is conducted in a series of blocks, in which the edge attributes are updated, followed by the node updates. It is also possible to assign global attributes to a graph, and these are updated at the end of each block. The updates are computed locally; that is node attributes are updated on the basis of values of attributes on some neighborhood set of nodes in a process rather like message passing in learning graphical models (Bishop, 2010). The actual update rules themselves can be based on update functions learning from the training data using standard learners, like MLPs. More general GNNs can output graphs with changed topologies as well as changed attributes.

Li et al. (2019) have recently introduced the graph matching networks (GMN) variant within the GNN family, where instead of categorizing or regressing on data, the objective is to determine the similarity between graphs. The GMN can be trained in two ways: pairs of labeled graphs or triplets of unlabeled graphs. In the first method, each graph within the dataset G_1, is paired with another graph within the dataset, G_2. If the graphs, G_1 and G_2, are determined to be similar, then a label of 1 is assigned to the pair; however, if the graphs are determined to be dissimilar, then a label of -1 is assigned to the pair.

$$(G_1, G_2) = t \epsilon \{-1, 1\} \tag{2}$$

In the second method of training the GMN, each graph, G_1, is paired with one graph within the dataset that it is similar, G_2, and one graph within the dataset that is dissimilar, G_3. The formed triplet does not require a label; however, it does require the order of the graphs within the triplet to be observed:

$$(G_1, G_2, G_3); G_1 \text{ is similar to } G_2, \text{but } G_1 \text{ is dissimilar to } G_3 \tag{3}$$

The work outlined in this paper has shown the potential of a common form within the PBSHM *network.* The main disadvantage with a method such as this is manually learning and forming the *CFRRs* to reduce the *detailed* IE model down to the *Canonical Form* representation. The hope of using a method such as the GMN, is that the neural network in the code of the GMN, can learn yet unknown reductions. To evaluate the use case of a GMN within the context of the PBSHM *network*, one first must establish if the GMN can learn the similarity without using the common form.

For the purpose of this paper, the GMN is trained using sets of labelled graph pairs (see equation 2), applying a loss function of the margin-based Euclidean distance, and utilizing the Adam optimizer (Kingma and Ba, 2014) for the minimization of the loss function. As mentioned in Section 2, the matching dataset is randomly separated into three subsets: training, validation, and test. Labelled pairs are generated for each unique directed combination of graphs within the subset i.e.

$$S_p = \{(G_1, G_2), (G_2, G_1), \ldots, (G_{N-1}, G_N), (G_N, G_{N-1})\} \tag{4}$$

https://doi.org/10.1017/dce.2024.45 Published online by Cambridge University Press

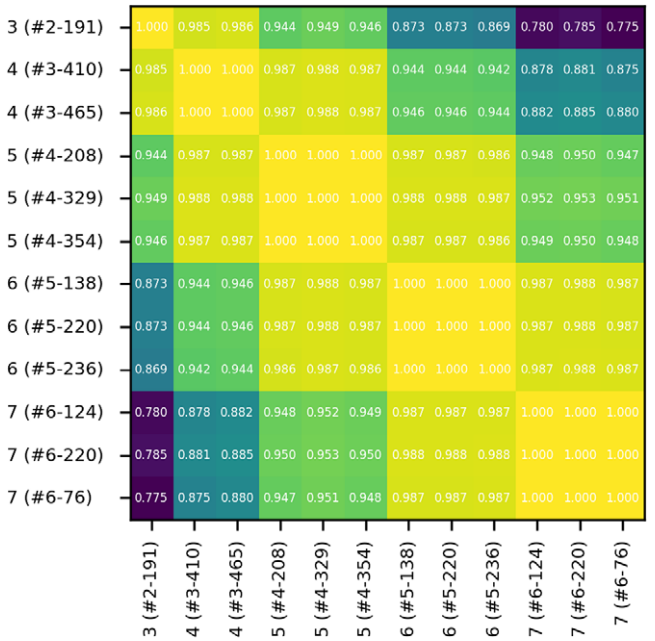


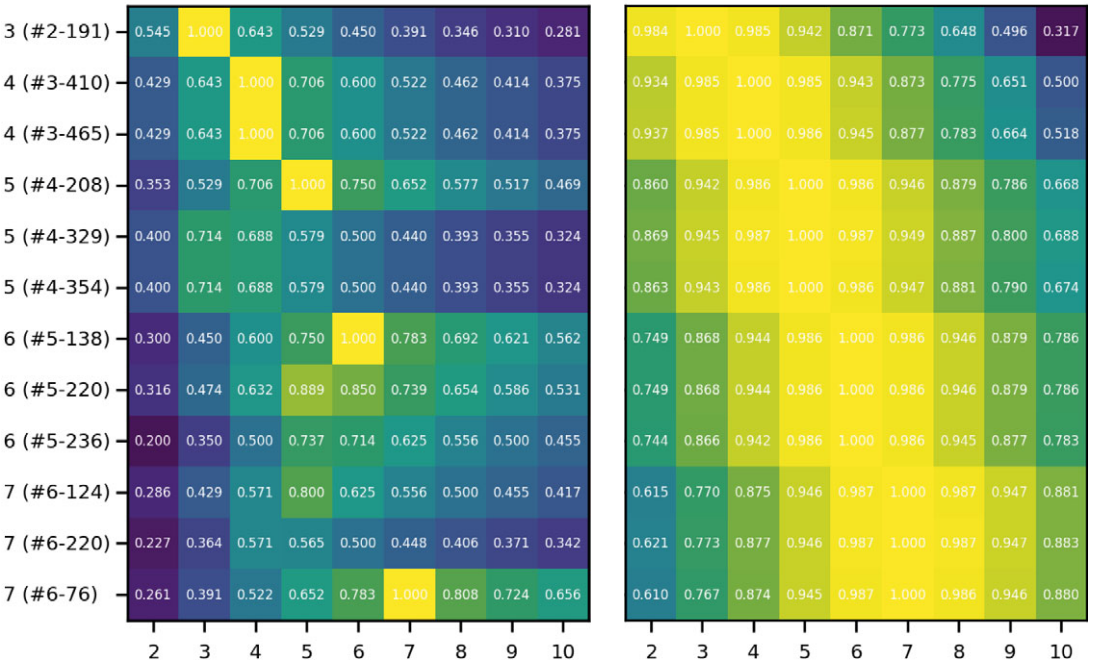
Figure 13. *The Graph-Matching Network similarity matrix results when comparing the detailed Irreducible Element model against itself. The axes are labelled with the number of spans the graph is associated with and the ID of the graph from within the dataset.*

In this particular problem, if the number of spans is the same in both graphs within the pair, a label of 1 is assigned to the pair; otherwise, a label of -1 is assigned. To limit the impact of overfitting, the GMN uses only the labelled pairs within the training subset for training. The learnt parameters are evaluated using the validation subset, with the test subset used as the independent "not seen before" dataset to generate the similarity results displayed within this paper. Finally, to align the numerical values between the GMN results and the Jaccard Index results, the similarity values generated by the GMN have been scaled between 0 and 1 —where 1 is similar— using the minimum and maximum values within the subset.

Figure 13 depicts the results of using the GMN against only the matching dataset. As one can see, the GMN is able to learn and identify the beam-and-slab bridges of the same span as identical, with a result of 1. The GMN is also able to identify the desired tiered similarity, when traveling away from the number of spans. If one looks at the results for the six-span bridges, the bridges with the closest similarity are the group of six-span bridges. The bridges with the next-nearest similarity are the bridges with five and seven spans, then the four-span bridges and finally the three-span bridges. While the results are not as separated in distance as the Jaccard Index results in Figure 10b, there is a small noticeable change in the results as the further one moves away from the target span.

Figure 14b shows the results of introducing the *Canonical Form* representation into the GMN comparisons; instead of the GMN learning the reductions needed between *detailed* IE models, the GMN learns the reductions required to reduce the *detailed* IE model to the *Canonical Form* representations. This requires modifying the labelled pairs within the training, validation, and test subsets to have one graph within the pair be a detailed IE model G_n^d, and the other graph be the *Canonical Form* representation G_n^{cf} i.e. $\left(G_n^d, G_n^{cf}\right)$. As one can see, the GMN is still able to identify *detailed* IE models to the *Canonical Form* representation with the same number of spans as identical. The results also show that the pattern of similarity decreasing with neighboring number of spans from the target span is also preserved. These results illustrate the flexibility of the GMN, the algorithm is able to learn the reduction rules between *detailed* IE model to *detailed* IE model or from *detailed* IE model to the *Canonical Form* representation.

https://doi.org/10.1017/dce.2024.45 Published online by Cambridge University Press



(a) The Jaccard Index similarity matrix results using the *Canonical Form Reduction Rules* to reduce the *detailed* Irreducible Element model before comparison. These are the same results as in Figure 10b.

(b) The Graph Matching Network similarity matrix results when comparing the *detailed* Irreducible Element model with the known *Canonical Form* representation.

Figure 14. *The similarity matrix results for both the Jaccard Index (see Figure 14a) and the Graph-Matching Network (see Figure 14b) when comparing the matching dataset —containing detailed Irreducible Element models— against the known Canonical Form dataset. For the Jaccard Index results, the Canonical Form Reduction Rules were used to reduce the detailed IE models before comparison. For the Graph Matching Network results, the Graph Matching Network learnt the reductions required against the training dataset —a labelled graph pairing of detailed Irreducible Element models and known Canonical Form representations. The Attributed Graphs for both algorithms contain only the embedding of the [regular] elements contextual type as a node attribute to keep results in direct comparison to Figure 5 and 13. The X axes are labelled with the number of spans of the Canonical Form graph, and the Y axes are labelled with the number of spans the graph is associated with and the IE of the graph from the matching dataset. The label for the Y axis is missing from the second figure because the labels are the same as the first figure.*

Figure 14 illustrates the results of comparing the performance of the Jaccard Index using the *CFRRs* (see Figure 14a), verses the GMN comparing the matching dataset to the known *Canonical Form* representation dataset (see Figure 14b). From the initial inspection of the results, it is clear to see that the GMN algorithm outperforms the Jaccard Index with *CFRRs* when considering the ability to identify a pattern of similarity within the example *network*; however, when one considers the context of the algorithms, the outcome is not so clear.

If one looks at the comparisons for the bridge *7 (#6–124)*, the Jaccard Index with *CFRRs* incorrectly identifies the five-span *Canonical Form* representation as the closet match to the input bridge, whereas with the GMN, the algorithm correctly identifies the seven-span *Canonical Form* representation as the closest match. The GMN is evidently —within the context of the example scenario— able to learn reduction rules which are not currently understood or implemented within the *CFRRs*; this may lead one to

https://doi.org/10.1017/dce.2024.45 Published online by Cambridge University Press

imagine that the GMN algorithm should be used above the Jaccard Index with *CFRRs*; however, to achieve this learnt knowledge, a not insignificant amount of bridges were required for the GMN to build the aforementioned knowledge. In direct comparison, the Jaccard Index with *CFRRs* required no previous examples of similar bridges before it could establish a similarity.

Without modification to the existing GMN algorithm, there is no methodology for extracting which *element*s or *relationship*s cause the similarity, thus providing a stumbling block in the algorithm's ability to communicate back to a *framework* user, why the given similarity is thus. The Jaccard Index with *CFRRs*, however, is able to communicate back to a *framework* user, as to where the similarity has been established via the MCS. Both of the aforementioned algorithms are able to generate a similarity within the *network*: and as such belong within the *framework.* When each algorithm should be used, perhaps, requires a larger viewpoint of the lifecycle of PBSHM.

While PBSHM is still within its infancy, one cannot rely upon the *network* having existing examples to generate learnt knowledge; instead, the *network* will need to depend upon algorithms which require no previous examples to learn from, such as the Jaccard Index with *CFRRs.* Once PBSHM has established itself to the extent of having multiple examples of a single type of structure, learning algorithms such as the GMN will have their place within PBSHM. The problem of data availability should not block research into learning algorithms; on the contrary, research should continue into machine-learning approaches — using simulated datasets— and focus on identifying what knowledge can be extracted from these approaches, and incorporated back into the global knowledge of similarity and processes such as the *CFRRs.*

5. Conclusions

In conclusion, this paper has highlighted the effect that author bias has in the variations present within the *network*, and the direct effect these variations have upon the computed similarity scores when using a graph theory-based calculation. The *Canonical Form* was introduced as the vehicle within PBSHM to reduce the effect that the variations have on the *network.* A *detailed* author-generated IE model is submitted into the *network*, the CFRRs then reduce the detailed IE model into the Canonical Form representation for comparison within the *network*, such that no author bias-based variations are present within the model, while retaining all structural knowledge relevant to the similarity comparisons.

The first three CFRR are introduced; however, these rules are not fixed and are in fact intended to be expanded over the course of time as further knowledge is obtained upon what structural knowledge is crucial for comparisons within the *network.* As the Jaccard Index and MCS algorithm is utilized in the previous published literature as a potential similarity metric within the *network*, the algorithm is used within this paper to benchmark the generated similarity scores when utilizing the CFRR. When using no CFRR before similarity comparison, the algorithm is unable to detect any noticeable pattern of similarity within the input graph dataset; however, when utilizing the CFRR to reduce the input graphs before comparison to the reference Canonical Form graphs, an initial pattern of similarity begins to appear. This highlights the potential of the CFRR to remove the variations introduced by author bias from the *network.*

An IE model is only concerned about encapsulating the knowledge of the structural components that encompass the system being modeled. The aforementioned remit of the knowledge contained within an IE model is immutable; however, the remit of knowledge to be included within the *network* is not. The *network* should —by its own definition— include all available knowledge on a structure which is pertinent to the similarity comparisons. Therefore, the *Reality Model* is introduced as the vehicle within PBSHM to encapsulate the knowledge regarding the world in which an IE model is placed. The direct consequence of the *Reality Model* upon the work included within this paper, is that labels or data defined within the *Reality Model* may restrict which *element*s within the IE model may or may not be reduced from the model, thus directly impacting the CFRR and the associated Canonical Form IE model.

Lastly, this paper evaluates the use of a machine-learning approach to deriving the similarity within the *network.* The GMN algorithm is used in comparison to the Jaccard Index and MCS method described above. The GMN is able to find the desired similarity patterns within the *network* and identify potential reductions which were not previously known when using the CFRR approach.

https://doi.org/10.1017/dce.2024.45 Published online by Cambridge University Press

The results of the GMN demonstrate the potential of machine-learning methodologies in calculating the similarity of structures within the *network*; however, it also highlights the requirements for future work. Additional research needs to be conducted into evaluating the included CFRR on IE models which produce non-planar graphs; new CFRR must be identified to align the results of graph theory-based algorithms with results from machine learning-based algorithms, and new machine-learning methods must be evaluated for computing the similarity within the *network*.

Acknowledgments. For the purpose of open access, the authors have applied a Creative Commons Attribution (CC BY) licence to any Author Accepted Manuscript version arising.

Data availability statement. Replication data and code can be found on github: https://github.com/dsbrennan/dce-2024-similarity-metrics.

Author contribution. Conceptualization: D.S.B; K.W. Methodology: D.S.B; T.J.R. Data curation: D.S.B. Data visualization: D.S.B. Software: D.S.B. Writing original draft: D.S.B. Writing review and editing: T.J.R; K.W. Supervision: E.J.C; K.W. Funding acquisition: E.J.C; K.W. All authors approved the final submitted draft.

Funding statement. The authors of this paper gratefully acknowledge the support of the UK Engineering and Physical Sciences Research Council (EPSRC) via grant reference EP/W005816/1.

Competing interest. None.

Ethical standard. The research meets all ethical guidelines, including adherence to the legal requirements of the study country.

References

Bacciu D, **Errica F**, **Micheli A and Podda M** (2020) A gentle introduction to deep learning for graphs. *Neural Networks 129*,203–221.

Barabási A-L and Pósfai M (2016) *Network Science*. Cambridge: Cambridge University Press.

Bishop CM (2010) *Neural Networks for Pattern Recognition*. Oxford: Oxford University Press, reprinted edition.

Brennan D. S, **Gosliga J**, **Cross E. J and Worden K** (2025) Foundations of population-based SHM, Part V: network, framework and database. *Mechanical Systems and Signal Processing 223*, 111602.

Bull L, **Gardner P**, **Gosliga J**, **Rogers T**, **Dervilis N**, **Cross E**, **Papatheou E**, **Maguire A**, **Campos C and Worden K** (2021) Foundations of population-based SHM, part I: homogeneous populations and forms. *Mechanical Systems and Signal Processing 148*, 107141.

Farrar C. R and Worden K (2007) An introduction to structural health monitoring. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences 365*(1851), 303–315.

Fernández M-L and Valiente G (2001) A graph distance metric combining maximum common subgraph and minimum common supergraph. *Pattern Recognition Letters 22*(6–7), 753–758.

Gardner P, **Bull L**, **Gosliga J**, **Dervilis Nand Worden K** (2021) Foundations of population-based SHM, Part III: Heterogeneous populations—mapping and transfer. *Mechanical Systems and Signal Processing 149*, 107142.

Gosliga J, **Gardner P**, **Bull L**, **Dervilis N and Worden K** (2021) Foundations of population-based SHM, Part II: heterogeneous populations—graphs, networks, and communities. *Mechanical Systems and Signal Processing 148*, 107144.

Gosliga J, **Hester D**, **Worden K and Bunce A** (2022) On Population-based structural health monitoring for bridges. *Mechanical Systems and Signal Processing 173*, 108919.

Jaccard P (1901) Distribution de la flore alpine dans le bassin des Dranses et dans quelques régions voisines. *Bulletin de la Société vaudoise des sciences naturelles 37*, 241–272.

Kingma D. P and Ba J (2014) Adam: a method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Li Y, **Gu C**, **Dullien T**, **Vinyals O and Kohli P** (2019) Graph matching networks for learning the similarity of graph structured objects. *arXiv:1904.12787 [cs, stat]*.

Newman M. E. J (2018) *Networks*, 2nd Edn. Oxford: Oxford University Press, .

Pan S. J and Yang Q (2010) A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering 22*(10), 1345–1359.

Tsialiamanis G, **Mylonas C**, **Chatzi E**, **Dervilis N**, **Wagg D. J and Worden K** (2021) Foundations of population-based SHM, Part IV: structures and feature spaces as geometry. *Mechanical Systems and Signal Processing 157*, 107692.

Weiss K, **Khoshgoftaar T. M and Wang D** (2016) A survey of transfer learning. *Journal of Big Data 3*(1), 9.

Cite this article: Brennan DS, Rogers TJ, Cross EJ and Worden K (2025). On calculating structural similarity metrics in population-based structural health monitoring. *Data-Centric Engineering*, 6, e24. doi:10.1017/dce.2024.45