IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 6 June 2024, accepted 25 June 2024, date of publication 27 June 2024, date of current version 5 July 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3419911

RESEARCH ARTICLE

Link Prediction in Industrial Knowledge Graphs: A Case Study on Football Manufacturing

MUHAMMAD YAHYA[1], ABDUL WAHID[1], LAN YANG[1], JOHN G. BRESLIN[1], (Senior Member, IEEE), EVGENY KHARLAMOV[2], AND MUHAMMAD INTIZAR ALI[3]

[1]Data Science Institute, University of Galway, Galway, H91 TK33 Ireland
[2]Bosch Center for Artificial Intelligence, 70005 Stuttgart, Germany
[3]School of Electronic Engineering, Dublin City University, Dublin 9, D09 V209 Ireland

Corresponding author: Muhammad Yahya (m.yahya1@universityofgalway.ie)

This work was supported in part by the Science Foundation Ireland (SFI) (Insight) under Grant 12/RC/2289_P2, in part by SFI (Confirm) under Grant 16/RC/3918, and in part by European Union (OntoCommons) under Grant 958371.

ABSTRACT The integration of heterogeneous and unstructured data in Industry 4.0, poses a significant challenge, particularly with advanced manufacturing techniques. To address this issue, Knowledge Graphs (KGs) have emerged as a pivotal technology, yet their deployment often encounters the problem of incompletion due to data diversity and diverse storage formats. This study tackles the challenge of KG completion by applying and evaluating state-of-the-art KG embedding models–ComplEx, DistMult, TransE, ConvKB, and ConvE–within a football manufacturing production line context. Our analysis employs two principal metrics of Mean Reciprocal Rank (MRR) and Hits@N (Hits@10, Hits@3, and Hits@1) to comprehensively assess model performance. Our findings reveal that TransE significantly outperforms its counterparts, achieving an average accuracy of 91%, closely followed by ComplEx and DistMult with accuracies of 87% and 84%, respectively. Conversely, ConvKB and ConvE exhibit lower performance levels, with accuracy values of 79% and 76%. Through rigorous statistical testing, including t-tests, meaningful differences in MRR values across the models have been observed, with TransE leading in MRR and ConvE at the lower end of the spectrum. Our research not only sheds light on the efficacy of various KG embedding models in managing tree-like structured datasets within the manufacturing domain but also offers insights into optimising KGs for improved integration and analysis of data in production lines. These contributions are valuable both from academic research in KG completion and industrial practices aiming to enhance production efficiency and data coherence in advanced manufacturing settings.

INDEX TERMS Industry 4.0, knowledge graph completion, link prediction, smart manufacturing.

I. INTRODUCTION

KnowledgeGraphs (KG) have attracted a lot of attention from the research community over years. They are currently being adopted in many domains, such as question-answering systems, information retrieval and recommendations in different domains, for instance, the supply chain system [1], the surface mounting process system [2], the automotive industry [3] and industries on the immediate list of industry 4.0. Knowledge graphs enable the integration of many data sources by establishing a unified and interconnected framework [4]. They differ from typical data fusion techniques by utilising semantic linkages and ontologies to present data in a more significant manner. This technique not only consolidates information from multiple sources but also improves the understanding and retrieval of data through extensive contextual associations. It facilitates the discovery of concealed connections and patterns among diverse information, resulting in enhanced precision and comprehensiveness of insights. The knowledge graphs facilitate the integration of diverse data sources by offering a shared semantic framework that enhances data quality and interoperability [5]. The authors

The associate editor coordinating the review of this manuscript and approving it for publication was Alba Amato.

in [6] reports on the capacity of KGs to depict connections and enable sophisticated data retrieval and examination was emphasised. The features of knowledge graphs make them an efficient instrument for integrating and analysing data in many fields.

Industry 4.0, also known as I4.0, is transforming the manufacturing industry by integrating modern technologies into production methods, resulting in substantial improvements in efficiency, adaptability, and product excellence [7].

Manufacturers are advancing their production lines by integrating technologies, for example, robots, advanced machines embedded with software, the Internet of Things (IoT), sensors, and so on [8], [9]. As a result, the employed technologies are producing vast amounts of data. However, the data entities are often stored in diverse data formats on different storage platforms and locations and are difficult to integrate [10]. Such heterogeneous and unstructured data need integration by using KG that can give a unified view of some technologies [11]. KGs are some of the emerging techniques [12] in I4.0. Industries such as Bosch [13], [14], Siemens [15], Valeo Vision Systems [16] and Forward Group [17] are working towards knowledge-based manufacturing systems that use semantic web technologies such as constructing ontologies[1] to build KGs. This is to facilitate the development of smart applications such as digital factories, production line automation, predictive maintenance, re-configurable manufacturing, data search, data inspection, system diagnostics, and building information summaries on top of the ontology.

As reported in the literature, the current research on I4.0-based KG is carried out in two dimensions: (i) techniques for building KGs [19], [20], [21], and (ii) applications of KGs [22], [23], [24]. To be more specific, regarding the first dimension, the current techniques used to build KGs focus on integrating data from heterogeneous sources, but most of the time, this results in imperceptible missing links between the graph entities [4]. As a consequence of the missing links within the KGs, it cannot be exploited for the aforementioned applications in conjunction with other powerful tools such as those for predictive maintenance, the prediction of the remaining useful life of complex systems, and product quality monitoring, among others. Moreover, the I4.0 data-based KGs are mostly prone to missing links [25]. Analyzing and predicting the missing links in such KGs is nearly impossible with human heuristics, and is highly dependent on the power of using relevant algorithms. [26].

The term ''link prediction'' refers to determining the likelihood of identifying pairs of nodes in a graph that will form a link or will not establish a link in the future. Graph-based link prediction research area has witnessed a number of models proposed using different architectures and approaches [27]. The proposed models are based on learning the features of KGs to predict links better than the previous ones [28]. Moreover, every model is built on different relational features such as relations, path information, and substructure information for training to improve the link prediction [29].

This study aims to explore state-of-the-art link prediction models such as TransE, DistMult, ComplEx, ConvKB, and ConvE on industrial data-based KGs. The topology of these KGs is different from the benchmark datasets commonly used. Our investigation has two key objectives: a) Academically, we aim to compare the performance of the link above prediction models and identify the best performance indicators for KG with a tree-like topology.[2] Identifying such indicators can help researchers and practitioners select the optimal learning model from similar KG. We employed standard evaluation metrics such as Mean Reciprocal Rank (MRR), and Hits@10, Hits@3 and Hits@1 to compare the performance of the different models. b) Industrially, our study has the potential to suggest missing links that have a high potential impact on the description of the entire production line knowledge model. Identifying missing links can support decision-making in production [30]. Moreover, our study can better understand how information flows through the production line, which can alternatively support process optimisation and quality control efforts. Below, we summarise the major contributions of our research:

1) To construct a knowledge graph(KG)-based dataset from the football manufacturing production line. The dataset comprises entities and relations relevant to the production process, enabling the analysis of the interrelationships among different components of the production line.
2) To train the state-of-the-art embedding models on the so obtained KG-based dataset. This will help analyze how the embedding models can capture the relationships between different entities in the simple KG, and to earn latent features that can be used to predict missing links.
3) To perform extensive experiments through datasets to evaluate the performance of embedding models. Using the standard metrics of MRR, and Hit@N (Hits@10, Hits@3 and Hits@1) to assess their ability to predict missing links in different manufacturing contexts.

A. PROBLEM FORMULATION

It is assumed that a manufacturing production line represents relationships between various nodes, such as material and manufacturing machines as shown in Figure 2. Nodes in the KG correspond to the machines, manufacturing processes,

[1] Here, we give a brief introduction to ontologies and refer readers to [18] to bring the artificial intelligence (AI) and digital twin (DT) technology on board. An ontology is the formal specification of an area of interest to describe a set of concepts, for example, `Machine, MachineParts` and the relationships (*hasPart*(`Machine, MachineParts`)) between them.

[2] KG with a tree-like topology are hierarchical data structures that display nodes and their relations as branching trees. Such topology efficiently models parent-child relationships into taxonomies for effective data retrieval and insights.

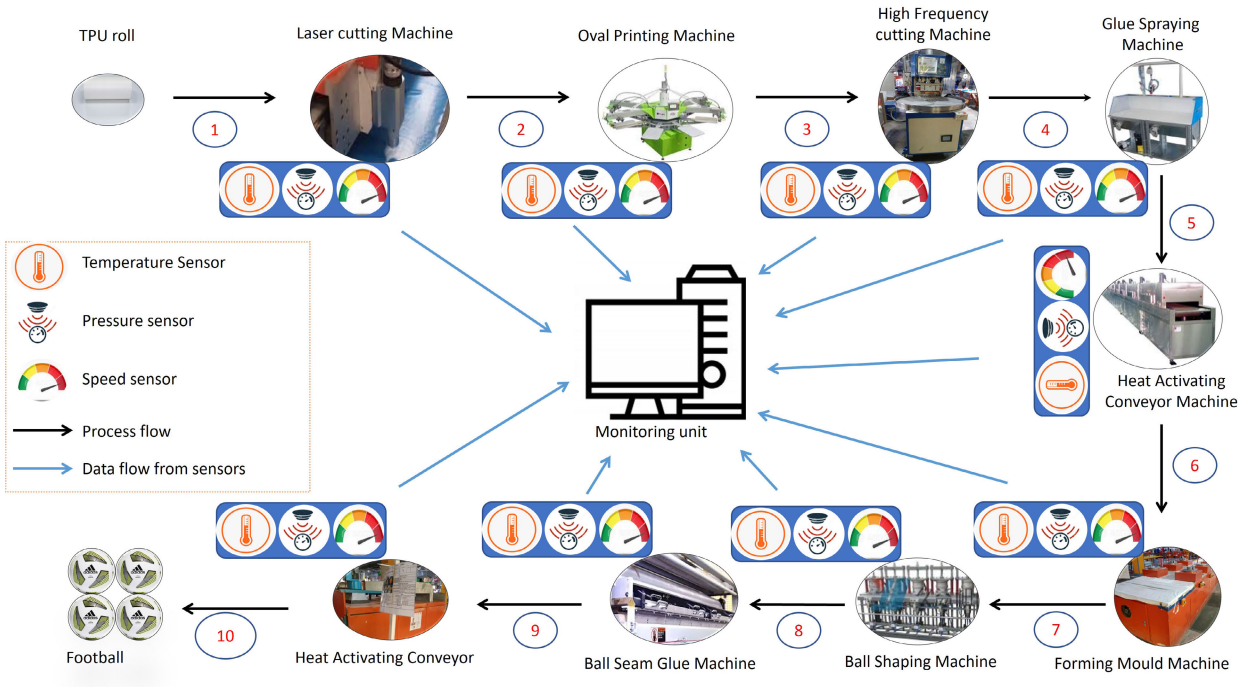


FIGURE 1. Data acquisition in football manufacturing production line. The black arrows shows the process flow in the production line and the blue arrows shows data flow from the sensors to the monitoring unit. ① In a single process, the TPU roll is feed into Laser Cutting Machine. ② Laser Cutting Machine converted the TPU roll into patches. ③ Patches are printed via squeegee by the oval printing machine. ④ Printed patches are cut into panels. ⑤ Back sides of panels and cores are sprayed with glue. ⑥ Glued panels and cores are pass by the heated conveyor to form moulding machine. ⑦ Cores and panels are moulded. ⑧ Balling shaping machine gives football shape to the moulded cores and panels. ⑨ The gaps between the panels are seal with glue via Ball seam glue machine.

materials, and their attributes while edges connect pairs of nodes representing some attributes that label the relationship, such as "Machines *hasInputMaterials* Materials". On this basis, a KG can be defined as a labelled directed graph $G = (V_e, E, T)$, such that V_e, and E are a set of nodes and labels representing entities and relations, and T represents the triples accordingly [31]. In the example scenario, an instance of a machine, Machine_3, is connected to an instance of a WorkStation, WorkStation_2, through the "hasMachine" relation in the KG. However, despite sharing common information and being instances of the same class, Machine_4 is not linked to WorkStation_2. Moreover, Table 4 includes some of the predicted triples by the trained models in the unseen data.

The rest of the paper is organized as follows. Section II presents the literature review. Section III explains the use case while Section IV provides an overview of the KG embedding models. Section V gives the details of the experiments and explains the results. Section VI provides the discussion. Finally, section VII concludes the paper and suggests possible future work.

II. RELATED WORK

KG has proved to be very efficient, therefore they have been extensively used in several downstream tasks such as recommendation systems [32], [33], question and answer [34], [35], natural language processing [36], [37], information extraction [38], [39], and many more. As KGs have been widely implemented and applied in a wide range of domains, and as a result an extensive amount of literature has been produced, especially about its completion. However, KGs suffer from incompleteness because of improper design and heterogeneous descriptions of entities, resulting in incorrect query results. However, link prediction studies in KG completion can help identify and fill the missing links or relationships between different entities which eventually help in the KG completion. In this section, (1) Link Prediction Models (i) Link prediction overview in general, (ii) Linked Prediction in I4.0-based KG), and (2) the datasets used for link prediction tasks are described.

A. LINK PREDICTION MODELS

An overview of the link prediction models for KG completion is presented below.

1) LINK PREDICTION MODEL OVERVIEW IN GENERAL

The main idea is that the learned embeddings should be able to generalize and assign high values to true facts that are not visible in the graph adjacency matrix, assuming that the model does not overfit the training set. In practice, the embeddings are learned as usual by optimizing the scoring function for all training facts, and the score of each fact is computed using that combination of the particular embeddings associated with that fact.

The first category among the link prediction models is the geometric-based (aka translation) model. It uses a spatial transformation for relation embeddings in the latent space. Provided a fact, a spatial transformation is used to represent the head embeddings where the values of relation embeddings are parameterized. Distance functions such as the L1 norm and L2 norm are employed to compute an offset between the resulting head and tail vectors. The additional constraints[3] in spatial transformation make the geometric models uniquely different from those of Tensor decomposition. Some of the examples of geometric models are RotateE [40], TransE [41], STransE [42], CrossE [43], TorusE [44]. Additionally, another approach that contributes to our understanding of complex relationships in multi-view data processing is introduced. A low-rank tensor regularized graph fuzzy learning (LRTGFL) method focuses on capturing nonlinear relationships and high-dimensional information through Jensen-Shannon divergence and tensor nuclear norm. [45]. Although this method is primarily aimed at multi-view clustering, its techniques can provide valuable insights into improving link prediction models by effectively capturing complex relationships within the data.

Deep learning models are the second category of link prediction models. These models consist of several architectures, such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), transformers, and generative adversarial networks (GANs), among others. Various models are appropriate for specific applications, including image recognition, natural language processing, and data production [46]. They employ convolutional neural networks (CNN) to learn features using weights and biases as estimators. These estimators are then combined with the input facts to extract features of significant importance. There are several different CNN architectures reported in literature [47]. However, their fundamental components are very similar. A CNN consists of three basic layers, namely, convolutional, pooling, and fully connected layers. The input feature set is represented in the convolutional layer, which consists of several convolution Kernels that are used to calculate various feature maps. Each neuron in a feature map is specifically linked to an area of nearby neurons in the layer underneath it. In the previous layer, this area is known as the neuron's receptive field. By first convolving the input with a learnt kernel and then using the convolutional results to apply an element-wise nonlinear activation function, the new feature map may be produced. The Kernel is shared by all spatial locations of the input to generate similarly produced feature maps. Several different Kernels are used to create the entire feature maps [47]. The final feature map is passed through a fully connected layer to compute the fact score. ConvE [48], ConvKB [49], ConvR [50], CapsE [51] are some of the deep learning models proposed on the aforementioned notion.

[3]For example, the rotation performed by model RotatE can be expressed as a matrix product, but the rotation matrix must be diagonal and contain elements with modulus 1 [40].

TABLE 1. Datasets used in link prediction tasks.

Dataset	Entities	Relations
FB15k	14951	1345
FB15k-237	14541	237
WN18	40943	18
WN18RR	40943	11

Deep learning has recently been applied to graph-structured data. Many existing algorithms use graph convolution neural network [52], which is a recursive creation of graph representation. GNN outperforms the conventional approaches based on off-the-shelf features as it allows for automatic and customized feature extraction from graphs and increases predictive performance [52].

In deep learning models, neural networks are utilized to learn features such as weights and biases that are grouped with the input facts to determine important patterns.

2) LINKED PREDICTION IN I4.0-BASED KG

In recent years, manufacturing industries have been moving towards adopting KGs in order to utilize properly their data [53]. A growing amount of research is being conducted on building and implementing KGs for use in manufacturing production lines. Ontologies are thus being proposed by researchers from industries to build KGs.

According to the literature, machine learning models such as Naive Bayesian, Random Forest, Decision Tree and Logistics Regression are analysed to predict missing links in the manufacturing production line KGs [25]. Thus, the KG-based embedding models such as ComplEx, Distmult, TransE, ConvKB and ConvE for predicting missing links in the manufacturing industries seem to have been overlooked.

B. DATASETS USED FOR LINK PREDICTION TASKS

The benchmark datasets used for the link prediction task have utilized for KG completion are being discussed in this section. The benchmark datasets are collected by sampling the real-world KG and provided in three file sets; train, validation and test. So far, the most commonly used benchmark datasets are listed in Table 1 and described as follows.

1) FB15K AND FB15K-237

FB15k is so far the most extensively used benchmark dataset for the link prediction tasks [27]. The FB15k dataset is created from the FreeBase[4] that contains more than 1.1 billion facts with 80 million entities [41]. Bordes et al. extracted around 592,213 facts that consisted of 14,951 entities and 1, 345 relations with a random split between training, validation and test set data. Toutana et al. [54] observed that FB15k suffers from test leakage, which occurs when models see test data during training time. This issue in FB15k dataset has been due to the presence of inverse or near-identical relations. Toutana et al. demonstrated that a basic model

[4]https://dbpedia.org/page/Freebase_database#

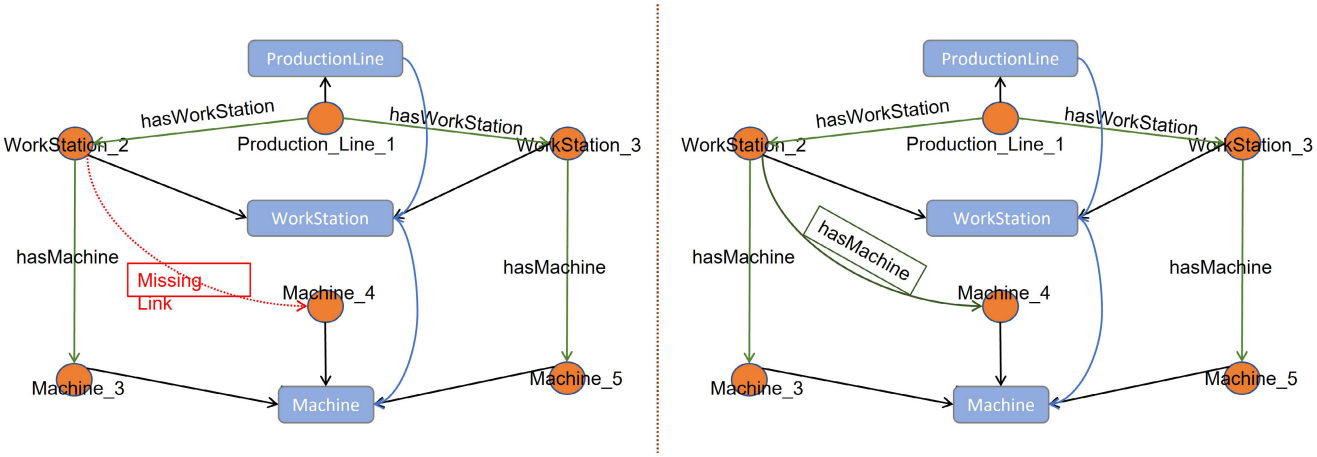


FIGURE 2. The figure depicts a snippet of the football manufacturing production line KG. The green arrow represents the semantic-based connectivity between instances while the blue arrow represents the connectivity between ontology classes (See Figure 6 in the appendix A to get a detailed high level understand of the ontology used in this work). The black arrows connect the instances to classes.

with observable features can easily achieve state-of-the-art performance on the FB15k. In regards to overcoming this issue of inverse or identical relations, the FB15k-237 has been constructed which contained facts from FB15k including 237 relationships.

2) WN18 AND WN18RR

WN18 is extracted from WordNet as a benchmark dataset and it contains 40943 entities and 18 relations [41]. WordNet[5] is a linguistically rich ontology with KG designed to serve as a dictionary/thesaurus to help NLP and automatic text analysis. The entities and relations in WordNet represent the synsets and their lexical connections, respectively. Likewise the FB15k, and the WN18 also have suffered from the test leakage that is caused by the inverse relationships between entities as reported by the authors in [48]. The authors built WN18RR by employing a similar methodology as that of FB15k-237. The WN18RR contains the same number of entities as that of WN18 and 11 relations by eliminating the inverse.

The link prediction task is highly applied to social network analysis where the focus is to improve the accuracy of the models as discussed earlier. There is a lack of publicly available industry datasets to enhance link prediction. The industry KG is simple but has a huge variety of data. Moreover, the current embedding models are performed on real-world case data from social network analysis.

Several strategies have been put forth to enable data-driven pipelines that turn industrial data into useful knowledge in smart manufacturing [55]. A smart factory can enable the semantic description and alignment of ''similar'' products as knowledge graphs that represent nodes and relations among them and their classification according to existing frameworks. Finding alignments across I4.0, however, necessitates the encoding of domain-specific knowledge expressed in standards of various kinds and standardization frameworks designed with various industrial objectives. To embed significant linkages and aspects of the I4.0 ecosystem and to empower interoperability in smart factories, we rely on cutting-edge knowledge representation and discovery technologies. However, we address the issue of missing relationships between the nodes of the production line knowledge graphs.

[5]https://wordnet.princeton.edu/

III. USE CASE

A. FOOTBALL PRODUCTION LINE KG

This work is motivated by the inquiry of whether it is possible to predict new facts by utilizing a football manufacturing KG. In the process of KG-based data integration, data from various sources are integrated and harmonized into manufacturing settings. The example scenario in Figure 2 demonstrates the elements necessary for semantic data integration and harmonization to create a KG. Although the harmonization of data offers significant benefits, the KG also provides a means of discovering new relations or links between data. These links can be created automatically based on the semantics encoded in the KG, allowing missing information to be completed. However, these links are often manually created and maintained, which is a time-consuming task.

B. DATASET ANALYSIS AND PREPARATION

This study investigates the possibility of using a football manufacturing KG to predict new facts. In the process of KG-based data integration, data from various sources is integrated and harmonised into manufacturing settings. The example scenario in Figure 1 demonstrates the elements necessary for semantic data integration to create a KG. Although the integration of data offers significant benefits, the KG also provides a means of discovering new relationships or links between data. These links can be created automatically

based on the semantics encoded in the KG, allowing missing information to be completed or restored.

The I40KG has a hierarchical structure where the nodes at the top are loosely connected. In state-of-the-art datasets, the nodes are somehow tightly connected due to the node types. The size of the football manufacturing production line KG is comprised of a total of 180701 entities and 35 relations that make 386905 triples. Among these, 9955 nodes are pendants that represent entities that are not highly connected in the KG. The KG has a density [56] of 2.369×10^{-5}, which represents the KG's connectivity that is calculated using equation (1).

$$d = \frac{m}{n^2} \tag{1}$$

where d represents the density, m is the number of edges, and n is the number of nodes. This metric indicates the overall connectivity of the KG. Furthermore, the KG has a mean degree centrality of 2.20×10^{-5}. The degree of centrality of a node is defined as the number of edges it has in the graph, normalised by the maximum possible number of edges. In addition to the KG's degree of centrality, the node `Heat_conveyor_operation` has a maximum degree of centrality, which is 5.53×10^{-6}. This indicates that even the most connected node in the graph has a relatively low number of connections, again in comparison to the maximum possible. Moreover, the KG has a mean network eigenvector centrality of 0.143. This number indicates the average influence of a node in the graph. Unlike degree centrality, eigenvector centrality considers the significance of the nodes to which a node is connected. A mean network eigenvector centrality of 0.143 indicates that the nodes in the network have a certain level of influence. In the context of our dataset, this value suggests that while there are some highly connected nodes, most nodes are moderately well-connected. This balance implies that the network does not have extreme centralization around a few nodes, but rather a more distributed influence across the network.

IV. KNOWLEDGE GRAPH EMBEDDING MODELS

To predict the missing links, we choose the five well-known models, that is, ComplEx, DistMult, TransE, ConvKB, and ConvE. About the aforementioned embedding models (see Table 2), we start with the TransE model's working process followed by the rest.

A. TransE

TransE is one of the most popular state-of-the-art embedding models. The training set S is made up of triplets (e_1, r, e_2), where $e_1, e_2 \in E$ (the set of entities) and $r \in L$ (the set of relationships). TransE learns how to embed entities and relationships into these triplets. These embeddings belong to $\mathbb{R}^k$ (k is a model hyperparameter) and are represented by boldface letters. The main concept of the TransE is that if the triplet $(\mathbf{e_1}, \mathbf{r}, \mathbf{e_2})$ does not hold (i.e., $\mathbf{e_1}$ and $\mathbf{e_2}$ are not connected by the relationship $\mathbf{r}$), then $\mathbf{e_1} + \mathbf{r}$ should be far away from $\mathbf{e_2}$ in the embedding space. This is an essential part of the TransE model, ensuring that the embeddings accurately reflect both the presence and absence of relationships. When the relationship $\mathbf{r}$ does not hold between $\mathbf{e_1}$ and $\mathbf{e_2}$, the model aims to maximize the distance between $\mathbf{e_1} + \mathbf{r}$ and $\mathbf{e_2}$, thus correctly representing non-connected nodes. Furthermore, using an energy-based framework, the energy of a triplet equals $d(\mathbf{e_1} + \mathbf{r}, \mathbf{e_2})$ for a dissimilarity measure d, which is chosen by TransE to be either the L1-norm or L2-norm. To obtain embeddings, TransE utilises a margin-based ranking criterion that is minimised over the training set, as given in Equation 2.

$$L = \sum_{(e_1,r,e_2)\in S} \sum_{(e_1',r,e_2')\in S'(e_1,r,e_2)} [\gamma + d(\mathbf{e_1} + \mathbf{r}, \mathbf{e_2}) - d(\mathbf{e_1'} + \mathbf{r}, \mathbf{e_2'})]_+ \tag{2}$$

where $[x]_+$ represents the positive features of x, $\gamma > 0$ is a margin hyperparameter.

$$S'(h, r, t) = \{(h', r, t) \,|h' \in E\} \cup \{(h, r, t') \,|t' \in E\} \tag{3}$$

To construct a set of corrupted triplets according to Equation 3, each training triplet is modified by replacing either the head or tail entity with a random entity, but not both at the same time. This strategy is effective because the loss function Equation 2 is designed to assign lower energy values to training triplets compared to corrupted triplets. By doing so, the loss function encourages the model to learn the embeddings that satisfy the intended criterion, and this happens naturally during the training process. It is worth noting that the embedding vector for a given entity is the same whether the entity appears as the head or the tail of a triplet. The optimisation process is conducted using stochastic gradient descent in minibatch mode over the possible e_1, r, and e_2 values.

B. DistMult

DistMult aims to learn the representations of entities and relations in a KG so that valid triplets receive high scores. Given a KG that is represented as a list of relation triplets (e1, r, e2) denoting a relationship r between entities e1 and e2. In order to learn the embeddings, a two-layer neural network is used. The first layer projects the input entities to low-dimensional vectors, and the second layer combines these vectors using a scoring function with relation-specific parameters to produce a scalar for comparison.

In relation to embedding learning, DistMult associates each input entity with a high-dimensional vector that can be either a "one-hot" index vector or an "n-hot" feature vector. The input vectors for entity e_1 and e_2 are denoted as x_{e_1} and x_{e_2}, respectively. Additionally, the first layer projection matrix is denoted by $\mathbf{W}$.

After passing the input vectors through the neural network, the model learned entity representations y_{e_1} and y_{e_2}. These representations can be expressed through Equation 4.

$$y_{e_1} = f(\mathbf{W}x_{e_1}), \quad y_{e_2} = f(\mathbf{W}x_{e_2}) \tag{4}$$

TABLE 2. Overview of scoring functions $\psi_r(e_s, e_o)$ utilised by various link prediction models, detailing their relation-dependent parameters.

Model	Scoring Function $\psi_r(e_s, e_o)$	Relation Parameters
TransE ([41])	$\|e_s + r_r - e_o\|_p$	$r_r \in \mathbb{R}^k$
DistMult ([57])	$\langle e_s, r_r, e_o \rangle$	$r_r \in \mathbb{R}^k$
ComplEx ([58])	$\langle e_s, r_r, \overline{e_o} \rangle$	$r_r \in \mathbb{C}^k$
ConvKB ([49])	$concat(g([\mathbf{v}_{e_s}, \mathbf{v}_r, \mathbf{v}_{e_o}] * \Omega)) \cdot \mathbf{w}$	$\Omega \in \mathbb{R}^{1\times 3}, \mathbf{w} \in \mathbb{R}^{\tau k\times 1}$
ConvE ([48])	$f(\text{vec}(f([e_s; r_r] * w))W)_{e_o}$	$r_r \in \mathbb{R}^{k'}$

where f is a function that can be either linear or non-linear and is applied element-wise to the result of the matrix multiplication between $\mathbf{W}$ and x_{e_1} or x_{e_2}.

Furthermore, DistMult utilises a basic bi-linear scoring function Equation 5.

$$g_r^b(y_{e_1}, y_{e_2}) = y_{e1}^T M_r y_{e_2} \tag{5}$$

DistMult's scoring function is a modified version of the Neural Tensor Network (NTN) scoring function. The NTN scoring function typically involves a non-linear layer and a linear operator. However, DistMult differs from NTN by removing the aforementioned components and utilizing a 2-dimensional matrix operator $M_r \in \mathbb{R}^{n\times n}$ instead of a tensor operator. Moreover, other matrix factorization models have also utilised the bilinear formulation of DistMult's scoring function, along with various forms of regularisation. To simplify the model and reduce the number of relation parameters, DistMult imposes a constraint on M_r such that it must be a diagonal matrix. This straightforward approach has been shown to be both simple and effective.

C. ComplEx

Let R and E denote the sets of relations and entities present in a KG. The ComplEx model aims to recover the matrices of scores X_r for all relations $r \in R$. Given two entities e_1 and $e_2 \in E$, the log-odds of the probability that the fact $r(e_1, e_2)$ is true can be expressed in Equation 6.

$$P(Y_{r,e_1,e_2} = 1) = \sigma(\varphi(r, e_1, e_2; \Theta)) \tag{6}$$

where φ is a scoring function and is based on observed relations factorization, and Θ represents the corresponding model's parameters. Although the entire X matrix is unknown, it is assumed that there exists a set of partially observed adjacency matrices for different relations, denoted as $\{Y_{re_1e_2}\}_{r(e_1,e_2)\in\Omega} \in \{-1, 1\}$. These matrices consist of true and false facts for the observed triples in the KG, where $\Omega \subseteq R\times E\times E$ is the set of observed triples. The objective is to determine the likelihood of whether entries Y_{r',e'_1,e'_2} are true or false, where the triples $r'(e'_1, e'_2)$ are targeted and unobserved, i.e., $r'(e'_1, e'_2) \notin \Omega$.

The scoring function adopted by the ComplEx model is given in Equation 7.

$$\sigma(\varphi(r, e_1, e_2; \Theta)) = Re(\langle w_r, e_1, e_2 \rangle) \tag{7}$$

where $w_r \in \mathbb{C}^K$ and represents a complex vector. The function $Re(\langle w_r, e_1, e_2 \rangle)$ in Equation 7 represents the real part of the complex dot product between the relation r embedding and the embeddings of entities e_1 and e_2.

D. ConvKB

ConvKB represents the dimensionality of embeddings as k, such that each embedding triple ($\mathbf{ve_1}$, $\mathbf{vr}$, $\mathbf{ve_2}$) is seen as a matrix $\mathbf{A}_i \in \mathbb{R}^{k\times 3}$, with $\mathbf{A}_i \in \mathbb{R}^{1\times 3}$ indicating the i-th row of $\mathbf{A}$. And utilise a filter $\boldsymbol{\omega} \in \mathbb{R}^{1\times 3}$ within the convolution layer. The purpose of $\boldsymbol{\omega}$ is not only to investigate the global relationships between identical dimensional entries of the embedding triple ($\mathbf{ve_1}$, $\mathbf{vr}$, $\mathbf{ve_2}$), but also to capture the transitional features in transition-based models. We repeatedly apply $\boldsymbol{\omega}$ over each row of $\mathbf{A}$ to ultimately produce a feature map $\mathbf{v} = [v_1, v_2, \ldots, v_k] \in \mathbb{R}^k$ is given in Equation 8.

$$v_i = g(\boldsymbol{\omega} \cdot \mathbf{A}_i + b) \tag{8}$$

where $b \in \mathbb{R}$ represents a bias term, and g denotes an activation function, for instance, the Rectified Linear Unit (ReLU).

ConvKB employs different filters $\in \mathbb{R}^{1\times 3}$ to produce distinct feature maps. Denote the collection of filters as Ω and the total number of filters as τ, such that $\tau = |\Omega|$. This leads to the generation of τ feature maps. These τ feature maps are then merged into a single vector in $\mathbb{R}^{\tau k\times 1}$, which is subsequently calculated with a weight vector $\mathbf{w} \in \mathbb{R}^{\tau k\times 1}$ through a dot product, yielding a score for the triple (e_1, r, e_2). Equation 9 presents the scoring function that has been adopted by ConvKB.

$$f(e_1, r, e_2) = concat(g([v_{e_1}, v_r, v_{e_2}] * \Omega)) \cdot w \tag{9}$$

where Ω and w represent shared parameters that are not dependent on e_1, r, or e_2; the symbol $*$ signifies a convolution operator; and the term 'concat' denotes a concatenation operator. The ConvKB model training loss is minimised via using Adam optimiser with L2 regularization on the weight vector was shown in Equation 10.

$$L = \sum_{(e_1,r,e_2)\in\{G\cup G'\}} \log(1 + \exp(l(e_1, r, e_2) \cdot f(e_1, r, e_2))) + \frac{\lambda}{2}\|w\|_2^2 \tag{10}$$

where $l(e_1, r, e_2)$ is a function that assigns labels to triples, and G' represents a set of invalid triples created by altering valid triples found in G.

E. ConvE

ConvE utilises a neural link prediction model that leverages convolutional and fully-connected layers to model the interactions between input entities and the relationships. The key feature of the ConvE model is that the score is established through a convolution performed over embeddings shaped in 2D. ConvE defines the scoring function as follows.

$$\psi_r(e_1, e_2) = f(\text{vec}(f([\overline{e_1}; \overline{r_r}] * \omega)) \cdot \mathbb{W}) \cdot e_2 \tag{11}$$

where the relation parameter, $r_r \in \mathbb{R}_k$, in the Equation 11 depends on r. Additionally, e_1 and r_r are subject to 2D reshaping, denoted as $\overline{e_1}$ and $\overline{r_r}$ respectively. Specifically, if both e_1 and r_r are elements of $\mathbb{R}^k$, then their reshaped forms $\overline{e_1}$ and $\overline{r_r} \in \mathbb{R}^{k_w \times k_h}$, where k is equal to $k_w \times k_h$.

During the feed-forward pass, the model conducts a row-vector lookup operation on two embedding matrices: one for entities, represented as $E^{|\mathcal{E}| \times k}$, and another for relations, denoted as $R^{|\mathcal{R}| \times k'}$. Here, k and k' are the dimensions of entity and relation embeddings respectively, and $|\mathcal{E}|$ and $|\mathcal{R}|$ represent the number of entities and relations respectively. The model concatenates $\overline{\mathbf{e_1}}$ and $\overline{\mathbf{r_r}}$ and uses the resulting vector as input to a 2D convolutional layer with filters ω. This layer produces a feature map tensor $\mathcal{T} \in \mathbb{R}^{c \times m \times n}$, where c is the number of 2D feature maps and m and n are their dimensions. The tensor $\mathcal{T}$ is then reshaped into a vector $\text{vec}(\mathcal{T}) \in \mathbb{R}^{cmn}$, which is subsequently projected into a k-dimensional space via a linear transformation that is parameterised by the matrix $\mathbf{W} \in \mathbb{R}^{cmn \times k}$. Finally, this projection is matched with the object embedding, e_o, through an inner product. It is important to note that the convolutional filter parameters and the matrix W parameters are independent of the parameters used for the entities e_1 and e_2, as well as the relationship r. Equation 12 represents the binary cross entropy function that is used to minimise the model loss.

$$\mathcal{L}(p, t) = -\frac{1}{N} \sum_{i=1} (t_i \cdot \log(p_i) + (1 - t_i) \cdot \log(1 - p_i)) \tag{12}$$

where t represents the label vector and p_i represents the predicted probability.

V. EXPERIMENTS AND RESULTS

A. EXPERIMENTAL SETUP

1) DATASET AND TRAINING

The data pre-processing used for the training the embedding models consists of several steps. Initially, the dataset is filtered for removing irrelevant predicates from the Football Production dataset. These triples are then subsequently reorganized in training and testing data. The data is then loaded to undergo text tokenization. Following this, a vocabulary is constructed and tokens are converted into indices. This pre-processed data is then fed into the selected embedding models for validation.

The dataset used for training and testing is 70% and 30%, respectively. Section IV-B summarises the dataset used in this research. The hypermeters are chosen by trying different values and observing their impact on model performance. Additionally, a learning rate of 0.0001 and latent feature dimensions k of 200 are chosen to train the state-of-the-art model. The number of negative triplets is set to five during training for each positive triplet. With a batch count of 100, the models are trained over 50 epochs. The loss function is minimised using the Adam algorithm.

The missing links are generated by creating corrupted triples, where either the head or tail of a valid triple is replaced with a random entity, but not both at the same time. During evaluation, for each test triple, the model computes and ranks the dissimilarities of these corrupted triples after replacing the head and tail with each entity from the dictionary, in turn, to determine the rank of the correct entity. The performance is then measured using metrics like mean rank and Hits@N, which reflect the proportion of correct entities ranked in the top 10, 3 and 1 predictions. Moreover, the ranking method involves evaluating test triples against all other candidate triples not present in the training, validation, or test sets. This is achieved by substituting either the subject or the object of a test triple with every entity in the knowledge graph, thereby generating candidate triples.

2) EVALUATION METRICS

An overview of the evaluation metrics employed to evaluate the accuracy of the rankings generated by these models is discussed. We use two main assessment metrics: Mean Reciprocal Rank (MRR) and Hits@N.

Mean Reciprocal Rank (MRR) calculates the average of the reciprocal ranks of the true (or correct) triplets. The reciprocal rank is the multiplicative inverse of the rank (that is, 1/rank) Equation 13.

$$\text{MRR} = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{R_i} \tag{13}$$

MRR is sensitive to how well the model ranks the highest-ranked true triplet, and a higher MRR indicates better performance. MRR ranges from 0 to 1, with 1 being the best possible score.

Hits@N (Equation 14) computes the percentage of true triplets that appear within the top N positions in the ranked list. We have used Hits@1, Hits@3 and Hits@10 for the model evaluation. A higher Hits@N value indicates better performance, as it means a larger proportion of true triplets are ranked within the top N positions.

$$\text{Hits @ N} = \frac{1}{Q} \sum_{i}^{Q} \delta(\text{rank}_i \leq N) \tag{14}$$

where Q is the count of positive and negative triples, rank_i is the rank of the positive triples within these triples, and δ is an indicator function that is 1 if $\text{rank}_i \leq N$, and 0 otherwise.

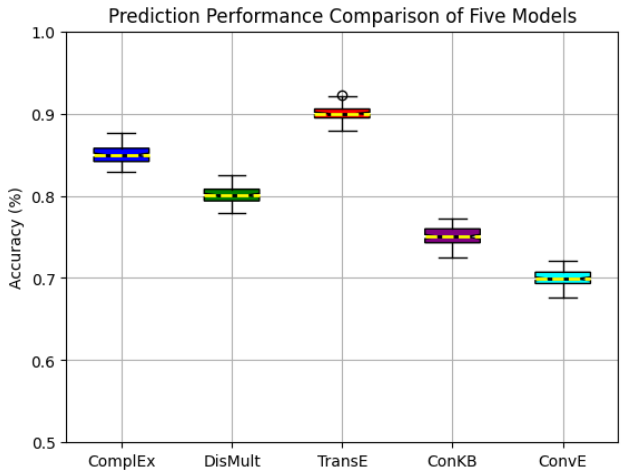


FIGURE 3. Performance comparison of five KG embedding models on unseen test data from the football manufacturing KG.

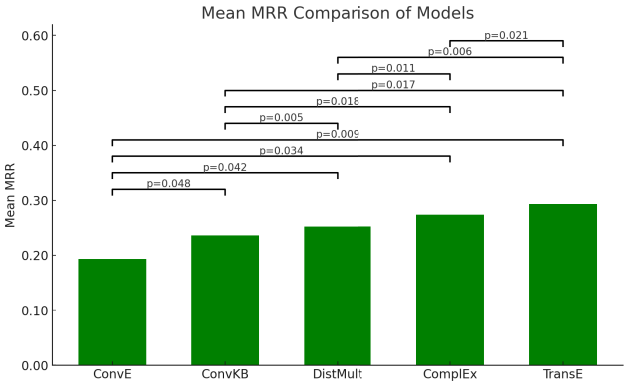


FIGURE 4. Comparison of mean reciprocal rank (MRR) for link prediction models i.e., ComplEx, DistMult, TransE, ConvKB, and ConvE on a football manufacturing dataset. The p-values from pairwise t-tests are annotated above the bars.

By comparing these metrics across different models, it can be determined which model performs better in ranking true triplets.

B. RESULTS

1) OVERALL RESULTS

The effectiveness of several KG embedding models, including ComplEx, DistMult, TransE, ConvKB, and ConvE, is thoroughly assessed. The experiments are carried out five times (named as five tests) and evaluated the models using the Mean Reciprocal Rank (MRR), Hits@10, Hits@3, and Hits@1 metrics shown in Table 3. It is observed from the overall results that the TransE model outperforms the other models for all test scenarios for football manufacturing production data in terms of MRR, Hits@10, Hits@3, and Hits@1. Additionally, ConvKB showed competitive outcomes, but none of the evaluation metrics saw it outperform TransE.

2) MODEL PREDICTION RESULTS

The models described in Section IV are used to predict the relationships between entities based on the known triples in the KG.

The prediction results (Figure 3) on unseen test data from the football manufacturing KG show that the models have achieved varying average accuracy levels between 0 and 1. TransE outperforms the other models with an average accuracy of 0.91, closely followed by ComplEx at 0.87 and DistMult at 0.84. The ConvKB and ConvE models have lower accuracies, with 0.79 and 0.76, respectively. The better performance of the TransE is due to its strategy of modelling the relationships as translations in the entity embedding space. This approach works well for hierarchical data, as entities in a hierarchical structure often have simple and direct relationships. On the other hand, ConvKB and ConvE are based on convolutional neural networks (CNNs), which are better suited for capturing complex and non-linear patterns in the data. As the manufacturing production line KG has a simple hierarchical structure, the convolutional layers in ConvKB and ConvE could not provide significant results in this case. Table 4 shows the accuracy achieved by five trained models, ComplEx, DistMult, TransE, ConvKB, and ConvE, for example, triples of unseen data.

3) STATISTICAL RESULTS

Significance tests, such as the t-test, are fundamental tools in statistics used to determine whether the differences observed between groups or models in an experiment are statistically significant or merely due to random chance. The p-values indicate the level of significance [59]. We performed pairwise t-tests between the MRR values of all potential model pairs to statistically evaluate the performance of these models. To evaluate the importance of the variations in MRR values between the models, the resulting p-values were computed from Table 3. Higher p-values imply that the difference between the compared models is not statistically significant, whereas lower p-values show a statistically significant difference between the compared models. Figure 4 represents the mean MRR values of the models, along with the pairwise p-values, highlighting the differences in their performance. The chart reveals that TransE outperforms the other models, while ConvE has the lowest mean MRR. Furthermore, the statistical analysis using t-tests shows significant differences between several model pairs, as indicated by the low p-values. Our research shows that the TransE model on the Hierarchical KGs such as the manufacturing football dataset performs best in terms of MRR.

4) TRAINING TIME ANALYSIS OF THE MODELS

The time analysis of training the models on the football KG dataset is explained in this section. Our study has analysed the training times of five state-of-the-art KG embedding models (ComplEx, DistMult, TransE, ConvKB, and ConvE) for 50 epochs. The training times for each model have been

TABLE 3. Comparative evaluation of KG embedding models ComplEx, DistMult, TransE, ConvKB, and ConvE across five test scenarios using mean reciprocal rank (MRR), Hits@10, Hits@3, and Hits@1 as performance metrics.

Models	Test 1				Test 2				Test 3			
	MRR	Hits@10	Hits@3	Hits@1	MRR	Hits@10	Hits@3	Hits@1	MRR	Hits@10	Hits@3	Hits@1
ComplEx	0.271	0.331	0.288	0.236	0.275	0.338	0.301	0.245	0.273	0.335	0.293	0.237
DistMult	0.255	0.311	0.274	0.221	0.252	0.307	0.271	0.218	0.258	0.312	0.276	0.225
TransE	**0.289**	**0.348**	**0.322**	**0.249**	**0.292**	**0.354**	**0.320**	**0.253**	**0.291**	**0.352**	**0.325**	**0.250**
ConvKB	0.240	0.321	0.268	0.193	0.238	0.322	0.266	0.191	0.227	0.320	0.265	0.170
ConvE	0.195	0.281	0.195	0.165	0.191	0.279	0.189	0.162	0.201	0.296	0.197	0.172
	Test 4				Test 5							
	MRR	Hits@10	Hits@3	Hits@1	MRR	Hits@10	Hits@3	Hits@1				
ComplEx	0.272	0.328	0.288	0.238	0.277	0.337	0.296	0.242				
DistMult	0.254	0.311	0.272	0.220	0.243	0.308	0.266	0.203				
TransE	**0.295**	**0.350**	**0.323**	**0.258**	**0.300**	**0.354**	**0.325**	**0.264**				
ConvKB	0.238	0.322	0.271	0.188	0.239	0.323	0.270	0.191				
ConvE	0.193	0.279	0.193	0.165	0.187	0.273	0.185	0.159				

TABLE 4. Performance comparison of KG embedding models on example triples from unseen data.

Triple	ComplEx	DistMult	TransE	ConKB	ConvE
WorkStation_2 hasMachine Machine_4	0.87	0.84	0.92	0.81	0.78
Machine2_Motor_State_177 hasState working	0.89	0.89	0.90	0.83	0.81
Squeegee3_Pressure_Sensor madeObservation Observation_180	0.86	0.80	0.92	0.75	0.73
Machine1_motor1 hasSpeed Machine1_motor_Speed_232	0.85	0.82	0.91	0.79	0.76
Oval_Printning_Process_3 useTool Machine2_Bed3	0.88	0.85	0.90	0.77	0.72

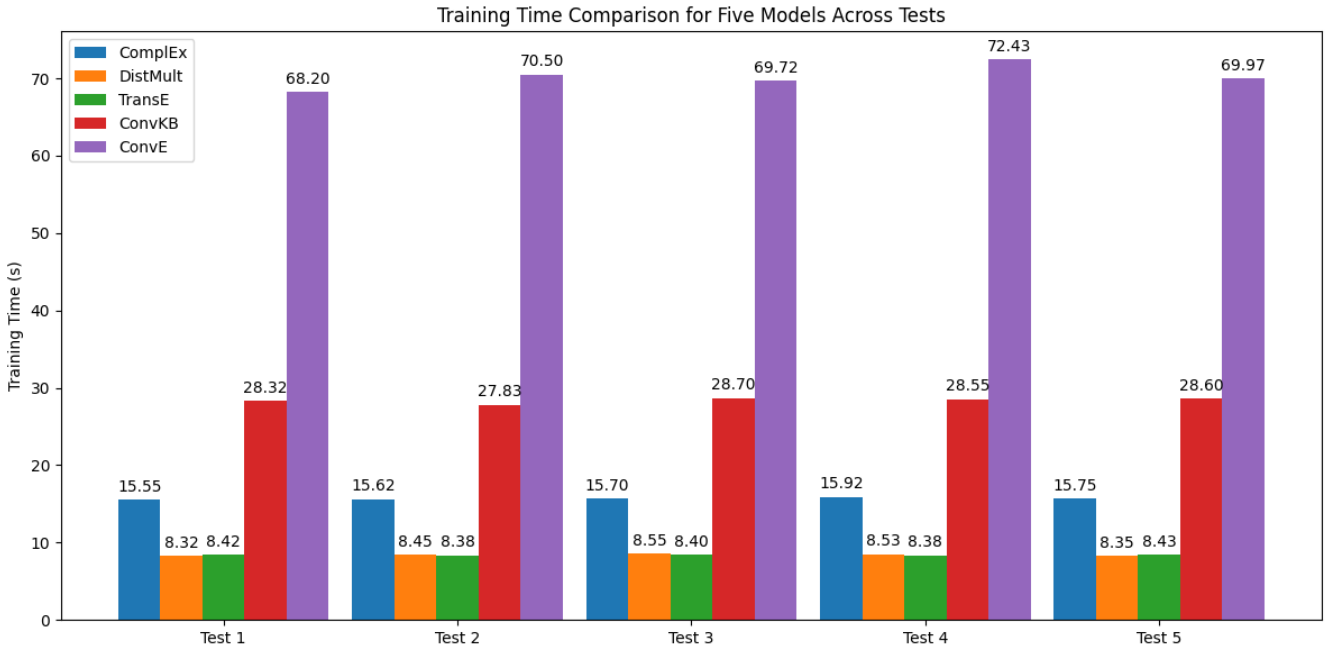


FIGURE 5. Training time for five models (50 epochs) - Tests 1 to 5.

recorded across five tests (see 5), and the results have been converted to minutes for easier comparison. The hardware used for experiments and implementation involved Nvidia GeForce GTX 1180 (8 GB of RAM) and Ubuntu 18.04.3 LTS (64-bit) operating system. The DistMult and TransE models have been found to have had the shortest training times, taking an average of eight (8) minutes and twenty six (26) seconds and eight (8) minutes and 25 (tewnty five) seconds, respectively to complete 50 training epochs. On the other hand, the ConvE model requires the longest training time,

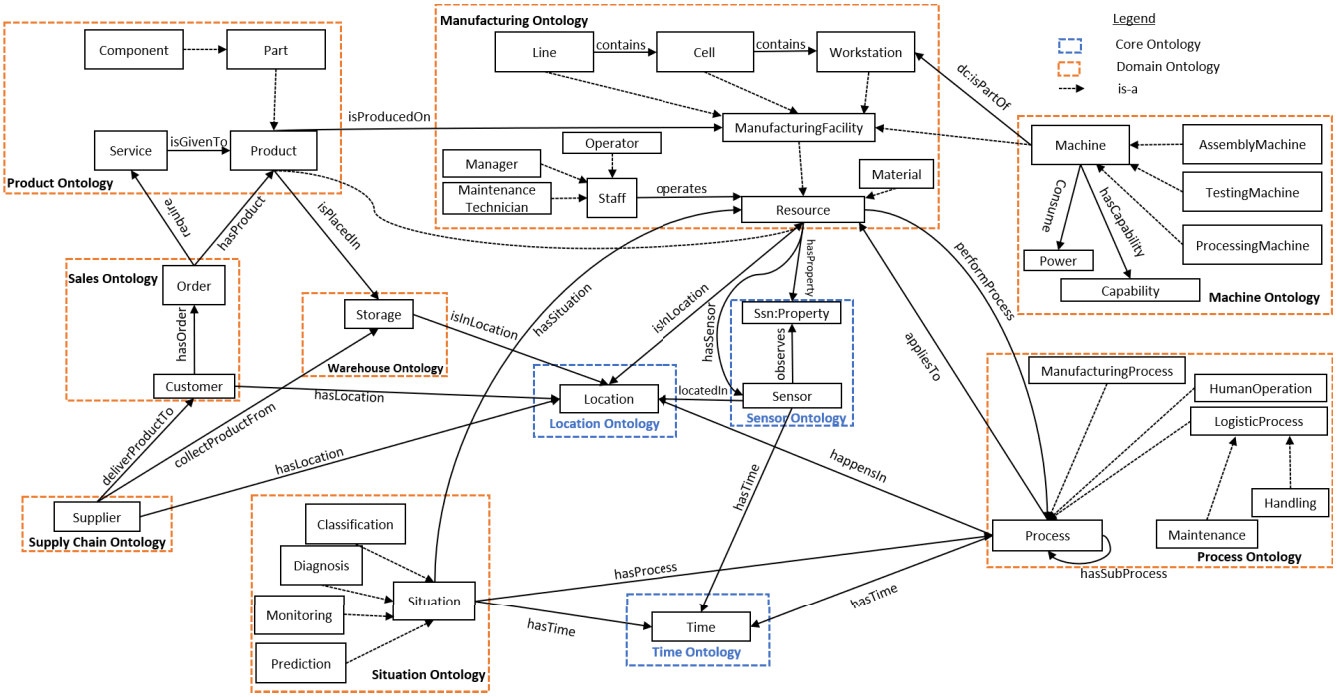


FIGURE 6. High level representation of Reference generalized ontological model used to develop Football production line knowledge graph.

with an average of seventy (70) minutes and twelve (12) seconds, indicating that it to be slowest model to train for the same number of epochs. The ComplEx and ConvKB models have had intermediate training times.

The analysis also have revealed some variation in training times between the different tests, particularly for the ConvE model, which has shown significant differences in training times between tests. Despite this variation, the DistMult and TransE models consistently have demonstrated the fastest training times throughout all tests. Overall, the findings suggest that the DistMult and TransE models are the most efficient models in terms of training times for KG embedding, whereas the ConvE model is the slowest. These results inform the choice of KG embedding models in different settings, particularly those where fast training times are crucial for KGs.

VI. DISCUSSION

Table 3 presents the results for various KG embedding models, offering critical insights into their interaction within specific KG frameworks, especially in manufacturing production lines. The varied performance among the models, especially the notable efficacy of translation-based models such as TransE compared to neural network approaches like ConvE, highlights the crucial role of choosing the right model based on the KGs' unique features, including its straightforward structure and sparse inter-entity connections. These observations emphasise the necessity for customised KG embedding strategies, indicating that models with a more straightforward, direct approach may outperform others in environments with less complex entity relationships. For instance, the better performance of TransE on our dataset can be attributed to several key properties of the data. The dataset has a hierarchical structure, with entities such as `Machine4_motor_Speed_3987` linked to specific value like `199RPM` via the relationship *smo:hasValue*. Additionally, processes like `Oval_Printning_Process_1` use tools such as `Machine2_Bed_1` and `Machine2_Heater_1` through *smo:useTool* relationships. TransE excels in this context because it models relationships as translations in the embedding space, effectively capturing these simple and hierarchical relationships. The sparse connectivity and straightforward nature of the relationships in the dataset further enhance TransE's performance. TransE's ability to model relationships as $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$fits well with the data's characteristics, allowing it to efficiently optimize translations without the need for complex computations. These findings suggest that TransE's performance would translate well to other knowledge graphs with similar properties, such as organizational charts, taxonomies, and Industry 4.0 applications where manufacturing processes are clearly defined. In these scenarios, TransE can effectively model hierarchical and straightforward relationships, ensuring robust performance and efficiency. Thus, TransE is particularly suitable for applications involving clear and direct relational data.

Moreover, these results require further exploration into refining KG embedding techniques, stressing the need to match model strengths with KG attributes for enhanced performance. This qualitative assessment not only highlights the existing constraints of current models but also paves the

way for future research and the practical deployment of KG to specific domains.

VII. CONCLUSION

In this study is evaluated the performance of five state-of-the-art knowledge graph (KG) embedding models, namely ComplEx, DistMult, TransE, ConvKB, and ConvE, on a KG constructed from an industrial dataset of a football manufacturing production line. The objective of the research has been to assess the models' effectiveness in order to identify the most suitable ones for a tree-like KG topology. The TransE model has performed better than the other models in terms of KG completion for the football dataset. Although rest of the models have delivered competitive results, however, it did not outperform TransE in any of the assessment metrics. The t-tests statistical analysis has revealed significant differences between several model pairs, with TransE achieving the best result in terms of mean reciprocal rank (MRR). In summary, this study advances the practical use of KGs, which facilitate improved production decision-making, and fosters the creation of more robust link prediction models for complex KGs.

APPENDIX A

See Figure 6.

REFERENCES

[1] E. E. Kosasih, F. Margaroli, S. Gelli, A. Aziz, N. Wildgoose, and A. Brintrup, "Towards knowledge graph reasoning for supply chain risk management using graph neural networks," *Int. J. Prod. Res.*, pp. 1–17, Jul. 2022.

[2] E. G. Kalaycı, I. G. González, F. Lösch, G. Xiao, A. Ul-Mehdi, E. Kharlamov, and D. Calvanese, "Semantic integration of Bosch manufacturing data using virtual knowledge graphs," in *Proc. Int. Semantic Web Conf.* Springer, 2020, pp. 464–481.

[3] D. G. Rajpathak, "An ontology based text mining system for knowledge discovery from the diagnosis data in the automotive domain," *Comput. Ind.*, vol. 64, no. 5, pp. 565–580, Jun. 2013.

[4] I. Grangel-González, F. Lösch, and A. U. Mehdi, "Knowledge graphs for efficient integration and access of manufacturing data," in *Proc. 25th IEEE Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, vol. 1, Sep. 2020, pp. 93–100.

[5] H. Paulheim, "Knowledge graph refinement: A survey of approaches and evaluation methods," *Semantic Web*, vol. 8, no. 3, pp. 489–508, Dec. 2016.

[6] L. Ehrlinger and W. Wöß, "Towards a definition of knowledge graphs," in *Proc. SEMANTiCS*, 2016, vol. 48, nos. 1–4, p. 2.

[7] T. Zheng, M. Ardolino, A. Bacchetti, and M. Perona, "The applications of industry 4.0 technologies in manufacturing context: A systematic literature review," *Int. J. Prod. Res.*, vol. 59, no. 6, pp. 1922–1954, Mar. 2021.

[8] A. Tarantino, "Introduction to industry 4.0 and smart manufacturing," in *Smart Manufacturing: The Lean Six Sigma Way*, 2022, pp. 1–19.

[9] M. Yahya, "Building semantic models and knowledge graphs for intelligent smart manufacturing applications," Rep., 2024.

[10] J. Yan, Y. Meng, L. Lu, and L. Li, "Industrial big data in an industry 4.0 environment: Challenges, schemes, and applications for predictive maintenance," *IEEE Access*, vol. 5, pp. 23484–23491, 2017.

[11] G. Xiao, L. Ding, B. Cogrel, and D. Calvanese, "Virtual knowledge graphs: An overview of systems and use cases," *Data Intell.*, vol. 1, no. 3, pp. 201–223, Jun. 2019.

[12] S. R. Bader, I. Grangel-Gonzalez, P. Nanjappa, M.-E. Vidal, and M. Maleshkova, "A knowledge graph for industry 4.0," in *Proc. 17th Int. Conf. Semantic Web (ESWC)*, Heraklion, Greece. Springer, Jun. 2020, pp. 465–480.

[13] M. Yahya, B. Zhou, Z. Zheng, D. Zhou, J. G. Breslin, M. I. Ali, and E. Kharlamov, "Towards generalized welding ontology in line with ISO and knowledge graph construction," in *Proc. ESWC*. Springer, 2022, pp. 83–88.

[14] M. Yahya, B. Zhou, J. G. Breslin, M. I. Ali, and E. Kharlamov, "Semantic modeling, development and evaluation for the resistance spot welding industry," *IEEE Access*, vol. 11, pp. 37360–37377, 2023.

[15] T. Hubauer, S. Lamparter, P. Haase, and D. M. Herzig, "Use cases of the industrial knowledge graph at Siemens," in *Proc. ISWC (PD/Industry/BlueSky)*, 2018.

[16] M. Yahya, A. Breathnach, F. Khan, I. Abaspur, and R. Ranganathan, "Towards semantic modeling of camera from image quality testing perspective: Valeo vision systems case," Tech. Rep., 2023.

[17] M. Yahya, A. Ali, Q. Mehmood, L. Yang, J. G. Breslin, and M. I. Ali, "A benchmark dataset for industry 4.0 and knowledge graphs," *Semantic Web J.*, 2022.

[18] P. Hitzle, M. Krótzsch, B. Parsia, P. F. Patel-Schneider, and S. Rudolph, "OWL 2 web ontology language primer," *W3C Recommendation*, vol. 27, no. 1, p. 123, 2009.

[19] G. Gawriljuk, A. Harth, C. A. Knoblock, and P. Szekely, "A scalable approach to incrementally building knowledge graphs," in *Proc. Int. Conf. Theory Pract. Digit. Libraries*. Springer, 2016, pp. 188–199.

[20] S. Tiwari, F. N. Al-Aswadi, and D. Gaurav, "Recent trends in knowledge graphs: Theory and practice," *Soft Comput.*, vol. 25, no. 13, pp. 8337–8355, Jul. 2021.

[21] D. Dessì, F. Osborne, D. R. Recupero, D. Buscaldi, and E. Motta, "Generating knowledge graphs by employing natural language processing and machine learning techniques within the scholarly domain," *Future Gener. Comput. Syst.*, vol. 116, pp. 253–264, Mar. 2021.

[22] S. Ji, S. Pan, E. Cambria, P. Marttinen, and P. S. Yu, "A survey on knowledge graphs: Representation, acquisition, and applications," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 2, pp. 494–514, Feb. 2022.

[23] X. Zeng, X. Tu, Y. Liu, X. Fu, and Y. Su, "Toward better drug discovery with knowledge graph," *Current Opinion Structural Biol.*, vol. 72, pp. 114–126, Feb. 2022.

[24] I. Tiddi and S. Schlobach, "Knowledge graphs as tools for explainable machine learning: A survey," *Artif. Intell.*, vol. 302, Jan. 2022, Art. no. 103627.

[25] I. Grangel-González and F. Shah, "Link prediction with supervised learning on an industry 4.0 related knowledge graph," in *Proc. 26th IEEE Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2021, pp. 1–8.

[26] L. Lü and T. Zhou, "Link prediction in complex networks: A survey," *Phys. A, Stat. Mech. Appl.*, vol. 390, no. 6, pp. 1150–1170, Mar. 2011.

[27] A. Rossi, D. Barbosa, D. Firmani, A. Matinata, and P. Merialdo, "Knowledge graph embedding for link prediction: A comparative analysis," *ACM Trans. Knowl. Discovery Data*, vol. 15, no. 2, pp. 1–49, Apr. 2021.

[28] M. Wang, L. Qiu, and X. Wang, "A survey on knowledge graph embeddings for link prediction," *Symmetry*, vol. 13, no. 3, p. 485, Mar. 2021.

[29] Y. Lin, X. Han, R. Xie, Z. Liu, and M. Sun, "Knowledge representation learning: A quantitative review," 2018, *arXiv:1812.10901*.

[30] L. Li, S. Bai, M. Leng, L. Wang, and X. Chen, "Finding missing links in complex networks: A multiple-attribute decision-making method," *Complexity*, vol. 2018, pp. 1–16, Sep. 2018.

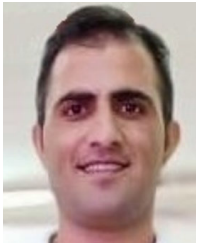
[31] X. Zou, "A survey on application of knowledge graph," *J. Phys., Conf. Ser.*, vol. 1487, no. 1, Mar. 2020, Art. no. 012016.

[32] Q. Guo, F. Zhuang, C. Qin, H. Zhu, X. Xie, H. Xiong, and Q. He, "A survey on knowledge graph-based recommender systems," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 8, pp. 3549–3568, Aug. 2022.

[33] B. Shao, X. Li, and G. Bian, "A survey of research hotspots and frontier trends of recommendation systems from the perspective of knowledge graph," *Expert Syst. Appl.*, vol. 165, Mar. 2021, Art. no. 113764.

[34] P. Tong, Q. Zhang, and J. Yao, "Leveraging domain context for question answering over knowledge graph," *Data Sci. Eng.*, vol. 4, no. 4, pp. 323–335, Dec. 2019.

[35] Z. Jiang, C. Chi, and Y. Zhan, "Research on medical question answering system based on knowledge graph," *IEEE Access*, vol. 9, pp. 21094–21101, 2021.
[36] F. Ameri, R. Yoder, and K. Zandbiglari, "Skos tool: A tool for creating knowledge graphs to support semantic text classification," in *Proc. IFIP Int. Conf. Adv. Prod. Manage. Syst.* Springer, 2020, pp. 263–271.
[37] S. Sharifirad, B. Jafarpour, and S. Matwin, "Boosting text classification performance on sexist tweets by text augmentation and text generation using a combination of knowledge graphs," in *Proc. 2nd Workshop Abusive Lang. Online (ALW2)*, 2018, pp. 107–114.
[38] Q. Xie, J. Pan, T. Liu, B. Qian, X. Wang, and X. Wang, "A survey of event relation extraction," in *Proc. Int. Conf. Frontier Comput.* Springer, 2022, pp. 1818–1827.
[39] R. Li, J. Zhong, Z. Xue, Q. Dai, and X. Li, "Heterogenous affinity graph inference network for document-level relation extraction," *Knowl.-Based Syst.*, vol. 250, Aug. 2022, Art. no. 109146.
[40] Z. Sun, Z.-H. Deng, J.-Y. Nie, and J. Tang, "RotatE: Knowledge graph embedding by relational rotation in complex space," 2019, *arXiv:1902.10197*.
[41] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 26, 2013.
[42] D. Q. Nguyen, K. Sirts, L. Qu, and M. Johnson, "STransE: A novel embedding model of entities and relationships in knowledge bases," 2016, *arXiv:1606.08140*.
[43] W. Zhang, B. Paudel, W. Zhang, A. Bernstein, and H. Chen, "Interaction embeddings for prediction and explanation in knowledge graphs," in *Proc. 12th ACM Int. Conf. Web Search Data Mining*, Jan. 2019, pp. 96–104.
[44] T. Ebisu and R. Ichise, "TorusE: Knowledge graph embedding on a lie group," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018.
[45] B. Pan, C. Li, H. Che, M.-F. Leung, and K. Yu, "Low-rank tensor regularized graph fuzzy learning for multi-view data processing," *IEEE Trans. Consum. Electron.*, vol. 70, no. 1, pp. 2925–2938, Feb. 2024.
[46] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 7553.
[47] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, and T. Chen, "Recent advances in convolutional neural networks," *Pattern Recognit.*, vol. 77, pp. 354–377, May 2018.
[48] T. Dettmers, P. Minervini, P. Stenetorp, and S. Riedel, "Convolutional 2D knowledge graph embeddings," in *Proc. AAAI Conf. Artif. Intell.*, vol. 32, 2018.
[49] D. Q. Nguyen, T. D. Nguyen, D. Q. Nguyen, and D. Phung, "A novel embedding model for knowledge base completion based on convolutional neural network," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.* New Orleans, LO, USA: Association for Computational Linguistics, 2018, pp. 327–333.
[50] X. Jiang, Q. Wang, and B. Wang, "Adaptive convolution for multi-relational learning," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2019, pp. 978–987.
[51] D. Q. Nguyen, T. Vu, T. D. Nguyen, D. Q. Nguyen, and D. Phung, "A capsule network-based embedding model for knowledge graph completion and search personalization," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2019, pp. 2180–2189.
[52] D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams, "Convolutional networks on graphs for learning molecular fingerprints," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015.
[53] L. Nagy, T. Ruppert, and J. Abonyi, "Ontology-based analysis of manufacturing processes: Lessons learned from the case study of wire harness production," *Complexity*, vol. 2021, pp. 1–21, Nov. 2021.
[54] K. Toutanova and D. Chen, "Observed versus latent features for knowledge base and text inference," in *Proc. 3rd Workshop Continuous Vector Space Models Compositionality*, 2015, pp. 57–66.
[55] P. O'Donovan, K. Leahy, K. Bruton, and D. T. J. O'Sullivan, "An industrial big data pipeline for data-driven analytics maintenance applications in large-scale smart manufacturing facilities," *J. Big Data*, vol. 2, no. 1, pp. 1–26, Dec. 2015.
[56] M. Zloch, M. Acosta, D. Hienert, S. Conrad, and S. Dietze, "Charaterizing RDF graphs through graph-based measures – framework and assessment," *Semantic Web*, vol. 12, no. 5, pp. 789–812, Aug. 2021.
[57] B. Yang, S. W.-T. Yih, X. He, J. Gao, and L. Deng, "Embedding entities and relations for learning and inference in knowledge bases," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2015.
[58] T. Trouillon, J. Welbl, S. Riedel, E. Gaussier, and G. Bouchard, "Complex embeddings for simple link prediction," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 2071–2080.
[59] M. D. Smucker, J. Allan, and B. Carterette, "A comparison of statistical significance tests for information retrieval evaluation," in *Proc. 16th ACM Conf. Conf. Inf. Knowl. Manage.* New York, NY, USA: Association for Computing Machinery, Nov. 2007, pp. 623–632.



MUHAMMAD YAHYA received the B.S. degree in computer science from KP Agricultural University, Peshawar, in 2013, the M.S. degree in electrical and electronic engineering from Univeristi Kuala Lumpur, in 2019, and the Ph.D. degree in engineering and informatics from the Data Science Institute, University of Galway, Ireland, under the supervision of Prof. John Breslin and Dr. Intizar Ali. He has published in major venues in these areas, such as IEEE Access, SWJ, ESWC, and ISWC.

ABDUL WAHID received the B.E. degree in software engineering from BGSB University, India, in 2012, the M.S. degree in electronics and information engineering from Chonbuk National University, South Korea, in 2016, and the Ph.D. degree from the Data Science Institute, University of Galway, Ireland, in 2023. Currently, he is a Postdoctoral Researcher with the Data Science Institute, University of Galway. He has published in many peer-reviewed journals and conferences.

LAN YANG received the B.E., M.E., and M.Eng.Sc. degrees, and the Ph.D. degree from the University of Galway, in September 2020, funded by Hardiman Scholarship, co-supervised by Tsinghua University. She is currently a Postdoctoral Researcher with the Unit for Social Semantics at Insight, University of Galway. Before joining Insight, she delivered lectures with the College of Science and Engineering and Centre for Adult Learning and Professional Development and conducted research with the Enterprise Research Centre, University of Galway. Her research interests include formal ontologies, knowledge graphs, semantic web, big data, healthcare analytics, and medical informatics. She has constantly been cooperating with reputed universities and organizations, such as Tsinghua University, the University of Oxford, Norwegian University of Science and Technology, and the International Council on Systems Engineering and publishing articles in leading journals, including IEEE Transactions on Industrial Informatics, *Computers in Industry*, and *Sustainability* (Switzerland). Her research achievements have been recognized through plenary presentations at international conferences and symposia. She is a reviewer of many peer-reviewed journals in her field and a scientific committee member of many international conferences.

JOHN G. BRESLIN (Senior Member, IEEE) is currently a Personal Professor (Personal Chair) in electronic engineering with the College of Science and Engineering, University of Galway Ireland, where he is also the Director of the TechInnovate/AgInnovate programmes. He has taught electronic engineering, computer science, innovation, and entrepreneurship topics during the past two decades. Associated with three SFI Research Centres, he is also a Co-Principal Investigator at Confirm (Smart Manufacturing) and Insight (Data Analytics) and a Funded Investigator with VistaMilk (AgTech). He has written more than 200 peer-reviewed academic publications (H-index of 42, 7400 citations, best paper awards from IoT, DL4KGS, SEMANTiCS, ICEGOV, ESWC, and PELS) and coauthored the books *Old Ireland in Colour*, *The Social Semantic Web*, and *Social Semantic Web Mining*. He co-created the SIOC framework (Wikipedia article), implemented in 100's of applications (by Yahoo, Boeing, and Vodafone) on at least 65,000 websites with 35 million data instances.

EVGENY KHARLAMOV has been with Bosch, since 2018. His work aims at developing AI methods that combine symbolic reasoning and machine learning, manufacturing knowledge captured as semantic conceptual models, digital twins, and knowledge graphs with production data. Such methods allow us to enhance and democratize industrial data analytics and analyses and develop industrial AI solutions, for example, semantically-enhanced machine learning pipelines for monitoring discrete manufacturing operations. His work aims for solutions that are based on solid theory and have a high impact on industry 4.0. He is active in international scientific and engineering communities and has more than 130 scientific publications. Some of them are awarded for the best paper award at top-tier venues. In 2021, he was ranked Nr 18 among "AI 2000 Knowledge Engineering Most Influential Scholars" according to AMiner. Moreover, he participates in publicly funded projects with multiple academic and industrial partners. Currently, he is running several AI and semantic-based projects with Bosch.

MUHAMMAD INTIZAR ALI received the Ph.D. degree (Hons.) from Vienna University of Technology, Austria, in 2011. He is currently an Assistant Professor with the School of Electronic Engineering, Dublin City University. His research interests include semantic web, data analytics, the Internet of Things (IoT), linked data, federated query processing, stream query processing, and optimal query processing over large-scale distributed data sources. He is actively involved in various EU-funded and industry-funded projects aimed at providing IoT-enabled adaptive intelligence for smart applications. He serves as a PC member for various journals, international conferences, and workshops.