TBRm: A Time Representation Method for Industrial Knowledge Graph

Keyan Cao * and Chuang Zheng

School of Computer Science and Engineering, Shenyang Jianzhu University, Shenyang 110168, China
* Correspondence: caokeyan@gmail.com; Tel.: +86-13889866997

Abstract: With the development of the artificial intelligence industry, Knowledge Graph (KG), as a concise and intuitive data presentation form, has received extensive attention and research from both academia and industry in recent years. At the same time, developments in the Internet of Things (IoT) have empowered modern industries to implement large-scale IoT ecosystems, such as the Industrial Internet of Things (IIoT). Using knowledge graphs (KG) to process data from the Industrial Internet of Things (IIoT) is a research field worthy of attention, but most of the researched knowledge graph technologies are mainly concentrated in the field of static knowledge graphs, which are composed of triples. In fact, many graphs also contain some dynamic information, such as time changes at points and time changes at edges; such knowledge graphs are called Temporal Knowledge Graphs (TKGs). We consider the temporal knowledge graph based on the projection and change of space. In order to combine the temporal information, we propose a new representation of the temporal knowledge graph, namely TBRm, which increases the temporal dimension of the translational distance model and utilizes relational predicates in time add representation in time dimension. We evaluate the proposed method on knowledge graph completion tasks using four benchmark datasets. Experiments demonstrate the effectiveness of TBRm representation in the temporal dimension. At the same time, it is also practiced on a network security data set of the Industrial Internet of Things. The practical results prove that the TBRm method can achieve good performance in terms of the degree of harm to IIoT network security.

Keywords: Industrial Internet of Things; knowledge graph; relational mapping; temporal incremental

1. Introduction

With the rapid development of Industrial Internet of Things (IIoT) and Knowledge Graph (KG) in recent years, IIoT applications in the industrial field are highly sensitive and critical, such as Industrial Control System (ICS), which integrates hardware and software to monitor and control the industrial environment operation of the system and its related components [1]. A Knowledge Graph (KG) stores facts about the real world as a collection represented by triples. Each triple in the KG is represented as \( \langle s, p, o \rangle \), where \( s \) and \( o \) represent the subject and object, and \( p \) represents the predicate connecting the subject \( s \) and the object. The problem with link prediction is to find the most suitable triples (\( s, p, \) ?) or (? \( p, o \)) to complete knowledge graphs [2]. Our focus is on the temporal knowledge graph, which adds time information to the triples. As shown in Figure 1, the time series knowledge graph essentially wants to extend the traditional static knowledge graph in the time dimension. The form of the problem with link prediction becomes that it is most likely to be completed under the given time information. That is to say, facts in temporal KG can also have the form of (subject, predicate, object, timestamp) or (subject, predicate, object, time predicate, timestamp), which is used to increase the general triple (\( s, p, o \)). For example, facts such as (Donald Trump, born, US, 1946) or (Donald Trump, President, US, occurs Since 2017-01) express time information about the facts related to Donald Trump [3].
The former expresses the relation type of the predicate that occurs at a specific point in time, whereas the latter uses the time predicate “occurs Since” to express the disclosed time period (time range).

![Figure 1. An example of a time-series knowledge graph.](image)

Recently, a lot of research work has been devoted to the representation learning of TKG. The method of link prediction is generally to embed the subject and predicate of the triple, and then use the scoring function to score. The temporal representation of knowledge graphs remains challenging due to the sparsity and irregularity of dynamic temporal information. The CYGNet model [4] utilizes the historical information of knowledge by designing a special replication module; meanwhile, a generation module is designed to predict the knowledge that appears for the first time. TA-DistMult [5] creates a temporal relationship by treating the characters in r and t as a sequence, and proposes a digit-level LSTM for learning the factual representations that contain temporal information in KGs and can be directly applied to current KGs. In the existing scoring function method in the completion task. Through the study of these models, we improved the TransR model and proposed a new TKG representation learning method. The proposed TBRm (Time boundary relationship mapping) represents time information as a single dimension. We map the relationship formed by the TransR model to the time dimension, and by embedding the mapped time predicate into the standard scoring function of knowledge graph completion. Circular neural networks are used to learn the time-aware representation of relationship types. We conduct extensive experiments on four benchmark TKG datasets, and the results show that TBRm can effectively model TKG data through relationships with temporal attributes.

In the direction of IIoT, taking cyberattacks on industrial IOT devices as an example, according to statistics from Kaspersky researchers, the number of cyberattacks on IOT devices jumped from 639 million in 2020 to 1.5 billion in 2021 [6]. In addition to the attack itself, there are other factors that affect the damage of network attacks. In this paper, according to the Edge-IIoT dataset, we use the attacker (represented by IP address) as the subject, the sensor and brake (represented by IP address) as the object, and the attack type as the predicate relationship to establish a static knowledge map. Furthermore, based on the established TBRm method, we take the duration of network attacks as another indicator to judge the degree of harm, form a temporal knowledge graph based on the degree of harm of network attacks, increase the judgment variables to judge the degree of harm of network attacks, and make the judgment basis for the degree of harm more convincing.
2. Related Work

With the development of artificial intelligence technology, people are more and more interested in KG embedding tasks, among which the most successful method is the embedding-based technique proposed by Nayyeri et al. [7–9]. These techniques map entities and relationships into a continuous space and define a scoring function to infer missing information. For static knowledge graphs, i.e., knowledge graphs without temporal dynamic facts, a class of classic models is translational distance models, such as TransE [10] and its extensions [11,12], which represent two entities as vectors and the relation Modelled as a translation vector. The TransH [13] model is improved on the basis of TransE, and the relationship is represented by two vectors, among which, H in TransH represents Hyperplane; this article is also the first to propose the negative sampling method of unif and bern. The TransR [14] model takes the projection of TransH to the hyperplane one step further, that is, to the space. The essence is to convert the projection vector into a projection matrix. The entity is still represented by a vector, and the relationship is represented by a vector and a matrix. The improvement in effect is not large, but the amount of calculation increases significantly. The R in TransR stands for relation space. Another class of classic models are semantic matching models, which represent relations as matrices and combine head and tail entities using multiplication, using triangular norm to determine how plausible a fact is [15,16]. With the continuous iteration and updating of technology, other models based on neural network methods using feedforward or convolutional layers [17,18] have received extensive attention for better performance. However, these models do not incorporate the temporal dynamics of the facts.

In terms of temporal knowledge graph, Jiang et al. improved the TransE model and proposed the TTransE model [19], which embeds temporal information into the score function, captures the temporal order between relation types, and uses common sense to constrain it to generate more accurate connection predictions. The HyTE [20] model maps each time to a hyperplane, which is equivalent to mapping different time points to different hyperplanes, and then models a relationship on each hyperplane, which is actually a triple relationship modeling. Although the time series knowledge graph is a very large graph, it can be a subgraph from each time point. In the article of Temp [21], the time series knowledge graph is divided into multiple subgraphs, and then each subgraph is convolved through GNN, and then the direct time sequence of multiple graphs is modeled based on RNN, and GNN’s results are concatenated and such methods take into account not only the spatial dimension, but also representation learning in the temporal dimension.

Compared with the traditional static knowledge graph, the temporal knowledge graph has more time information. The knowledge in the knowledge graph is not static, but will change with time. We divide all algorithm models into three categories based on their differences in the way they process time information: temporal knowledge graph representation model with time constraints, time series coding temporal knowledge graph representation model and path reasoning temporal knowledge graph representation model, as shown in Table 1.

Table 1. Classification of known temporal knowledge construction methods.

<table>
<thead>
<tr>
<th>Temporal Knowledge Graph Represents Categories</th>
<th>Model Abbreviation</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Temporal knowledge graph representation model with time constraints</td>
<td>ETA-TransE [22]</td>
<td>On the basis of the TransE model, a time transfer matrix is constructed based on the difference in the time granularity of the application scenario, which can distinguish the impact of the same time on different types of entities.</td>
</tr>
<tr>
<td></td>
<td>ATiSE [23]</td>
<td>The influence of mining time on the evolution of entities.</td>
</tr>
<tr>
<td></td>
<td>TTransE [24]</td>
<td>On the basis of the TransE model, time information is used to embed and represent time points in the triplet by relationship-time merging.</td>
</tr>
</tbody>
</table>
### Table 1. Cont.

<table>
<thead>
<tr>
<th>Temporal Knowledge Graph Represents Categories</th>
<th>Model Abbreviation</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time series coding temporal knowledge graph representation model</td>
<td>Know-Evolve [25]</td>
<td>By constructing an RNN network to update the embedded representation of the entity after it is affected by time changes.</td>
</tr>
<tr>
<td></td>
<td>RE-NET [26]</td>
<td>Convert the time information into a sequence of events (triples) with time information, and finally use the RGCN network to aggregate the information of entities at the same time.</td>
</tr>
<tr>
<td>Path reasoning temporal knowledge graph representation model</td>
<td>Chang2vec [27]</td>
<td>Split the time series knowledge graph into multiple static knowledge graphs according to time nodes Spectrum, recalculate the changed node entity representation and update its embedded representation.</td>
</tr>
<tr>
<td></td>
<td>xERTE [28]</td>
<td>The model can visualize the interpretability of reasoning and show the reasoning path.</td>
</tr>
</tbody>
</table>

### 3. Research Methods

To address the fact that TransE and TransR do not mention time in fact, we propose TBRm, which represents a mapping of relational spaces to temporal dimensions bridged by time-specific matrices.

#### 3.1. Problem Statement

Compared with traditional knowledge graphs, Temporal Knowledge Graphs have additional information. Each representation in the original knowledge graph is a triple, whereas each representation in the time-series knowledge graph is a quadruple, which not only includes Subject, object and relational predicate, as well as the time point or time range in which the relational predicate is established. Figure 2 shows an example of including time in a knowledge graph. In the example, the fact that Donald Trump is the President of the United States is only true in the time period 2017–2021, not in all time periods. When we ask a question, we need to know the specific time to find out exactly what we want. The tail entity of, so the facts in the time-series knowledge graph are time-dependent, and it is very important to consider the time of facts.

![Figure 2. A clip of ICEWS showing records of the incumbents of the U.S. President over different time periods. Not all U.S. presidents in office are listed in the figure, and unlisted presidents and future presidents are represented by dotted lines.](image-url)
In terms of network security, the requirements knowledge graphs based on the IIoT network security system, as shown in Figure 3. Determining the degree of harm caused by network attacks, and deploying and focusing on prevention of potentially harmful network attacks in advance are conducive to ensuring the network security of IIoT. Clearly, different types of attacks have different degrees of harm, but it is worth thinking about which variables can be added in addition to the types of attacks to make the determination of the degree of harm more accurate.

**Figure 3.** Knowledge Graphs of requirements based on IIoT network security.

In TKG, each fact has a relation (or predicate) $p \in \mathcal{R}$ with subject entity $s \in \mathcal{E}$ and object entity $o \in \mathcal{E}$ within time $t \in \mathcal{T}$. where $\mathcal{E}$ and $\mathcal{R}$ represent the vocabulary sets corresponding to entity and relation predicates, respectively, and $\mathcal{T}$ is the set of time periods or timestamps (if they exist). Bold words $s$, $p$, $o$, $t$ represents the embedding vectors of subject entity $s$, predicate $p$, object entity $o$, and time $t$ in the factual events with temporal information. Let $\mathcal{G}_t$ denote the snapshot of TKG in $t$ time period, $t = (t_p, t_e)$, where $t_p$ denotes the time when a relationship starts, and $t_e$ denotes the time when a relationship ends. $g = (s, p, o, t)$ represents the quadruple fact in $\mathcal{G}_t$. Both TransE and TransH assume embedding entities and relations in the same space $\mathbb{R}^k$. In TransR, for each triple $(s, p, o)$, the entity embedding is set to $s, o \in \mathbb{R}^k$, and the relation embedding is set to $r \in \mathbb{R}^d$. Here, the dimensions owned in entity embedding and relationship embedding may not be the same, that is, $k \neq d$. However, none of these three classical models contain temporal information. To solve this problem, we propose a new method that projects triples in relational space onto an extended temporal space and translates them in temporal space, thus was named TBRm.

### 3.2. Model Structures

For each relation predicate $p$ in the triple, TransR sets a projection matrix $\mathbf{M}_p \in \mathbb{R}^{k \times d}$, which can project entities from entity space to relation space. Based on this idea, we extend the time dimension on the basis of relational space to form a time-space that is perpendicular to relational space. Define the Vertical projection matrix $\mathbf{V}_t \in \mathbb{R}^{d \times d}$, $\mathbf{V}_t$ can map the entity set and relation set to the time space, and reconstruct the triples under the limitation of the time dimension, so that the relation predicate contains it can be limited by duration. The overall structure of TBRm is shown in Figure 4.
The scoring function $f$ is used for the embedding method of KG completion. The function works on the embedding of the subject $e_s$, the object $e_o$, and the predicate $e_p$ of the triple in the time dimension, in order to represent the duration (time span) of a relational predicate, let $t = t_e - t_b$, it is worth noting that when the time represents the relational predicate in the form of timestamp, only $t_b$ is taken as the current time vector, that is, when $t_e = t_b$, $t = t_b$. The value of a scoring function is proportional to the probability that a triple is true; a classic example of a scoring function is:

1. **TransE** [10]:

\[
    f_t(s, p, o) = \|e_s + e_p - e_o\|_2
\]

where $e_s, e_o \in \mathbb{R}^d$ are the embeddings of subject and object entities, and $e_p \in \mathbb{R}^d$ are the embeddings of relational predicates. $\| \cdot \|_2$ represents the two-norm.

2. **TransR** [14]:

\[
    f_p(s, o) = \|s_p + p - o_p\|_2^2
\]

where $s_p = sM_p$, $o_p = oM_p$, and at the same time, the embedding and mapping matrices must be constrained, that is, $\|s\|_2 \leq 1$, $\|o\|_2 \leq 1$, $\|p\|_2 \leq 1$, $\|s_p\|_2 \leq 1$, $\|o_p\|_2 \leq 1$. These scoring functions do not consider temporal information.

To introduce the temporal information of relational predicates, TBRm uses the projection matrix $V_t$ to define the projection of relational predicates as:

\[
    p_t^e = pV_t, \quad p_t^o = pV_t
\]

where $p_t^e$ represents the projection of the entity (subject) set at the beginning of the relation predicate on the time dimension, and $p_t^o$ represents the projection of the entity (object) set at the end of the relation predicate on the time dimension.

The scoring function is correspondingly defined as:

\[
    f_t(s, p, o) = \|s_p + p_t^o - p_t^e - o_p\|_2^2
\]

where $s_p = s_pV_t$, $o_p = o_pV_t$. At the same time, the constraints of the projection matrix are satisfied, that is, $\|s_p\|_2 \leq 1$, $\|o_p\|_2 \leq 1$. 

Figure 4. The overall structure of the TBRm model.
3.3. Training Target

Predict (object) entities given a query (s, p, ?, t). The learning goal is to minimize all the cross-entropy loss functions $L$ of TKG snapshots that exist during training. By referring to the training method of the paper [29], we use the following margin-based scoring function as the training goal:

$$L = \sum_{(s, p, o) \in S} \left( \max(0, f_t(s, p, o) + \alpha - f_t(s', p', o')) \right)$$

(5)

where $\max(x, y)$ aims to obtain the maximum value between $x$ and $y$, $\alpha$ is the margin, $S$ is the set of correct triples, and $S'$ is the set of incorrect triples.

Existing knowledge graphs only contain correct triples. It is reasonable to destroy correct triples $(s, p, o) \in S$ by replacing entities, and construct incorrect triples $(s', p', o) \in S'$. When breaking triples, we follow [30–41] and assign different probabilities to head/tail entity replacement.

4. Experiment and Discussion

In this section, we demonstrate the effectiveness of TBRm using four public IIoT datasets. First of all, we will explain the experimental settings in detail, including a detailed introduction to the baseline and dataset. Then, we analyzed and discussed the experimental results. We also conducted a comparative study to evaluate the advantages of the TBRm method over other baseline methods. Finally, we put the proposed representation method into practice on the Edge-IoTset data set to prove the feasibility of TBRm on the Industrial Internet of Things. The specific code can be found at https://github.com/Dash69dash/temporalKG (accessed on 27 October 2022).

4.1. Dataset

We conducted experiments on the connection prediction of TBRm on four benchmark data sets related to the industrial Internet. At the same time, in order to see the performance of the proposed method on the Industrial Internet of Things, we used another Industrial Internet of Things data set to test the representation effect of TBRm. Table 2 summarizes the statistics of the dataset. The Bosch production line internal fault data set (BPLP) [42] describes the measurement results of parts as they move in the Bosch production line. Each part has a unique ID and contains a large number of anonymous features. Features are named according to the agreement that tells you the production line, the workstations on the production line, and the feature number. We separate the files according to the types of features contained in the files: numbers, categories, parts and workstations on the production line as subject entities and object entities, and fault types as relational predicates in the triplet. Finally, there is a file with a date feature, and the date function provides a timestamp of each measurement time. MOOC Platform User Behavior Data Set (MOOC-Ub) [43] includes the learning activities of all users on the school’s online platform from August 2015 to August 2017. User information is the information of users of the school online, including: gender, year of birth and education level. Course information includes the course start date, course end date, course category, and course type. Extract and filter the user ID as the subject entity, the course category and type as the object entity, and the user information and course information as the relational predicate. Combining the start and end times of the course, the model of the time series knowledge graph is used for experiments. Each json file contains user tracking logs for a specific period of time. The NFT Ethereum transaction data set (NFT) [44] contains the transaction activities of the Ethereum non-homogeneous currency (NFT) from 1 April 2021 to 25 September 2021. It is purely constructed from on-chain data and represents the activities of 9292 NFT smart contracts on the Ethereum blockchain during the period. The Information Exposure from Consumer IoT Devices (IE-IoTD) [45] dataset processed and analyzed the information leakage of 81 devices located in laboratories in the United States and the United Kingdom. We filtered out a total of 23,475 triplet relationships with timestamps.
Table 2. Summarizes the statistics of the dataset.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>BPLP</th>
<th>MOOC-Ub</th>
<th>NFT</th>
<th>IE-IoTD</th>
<th>Edge-IoTset</th>
</tr>
</thead>
<tbody>
<tr>
<td>#Entities</td>
<td>3024</td>
<td>24,100</td>
<td>5422</td>
<td>12,564</td>
<td>108,576</td>
</tr>
<tr>
<td>#Relation</td>
<td>145</td>
<td>274</td>
<td>186</td>
<td>245</td>
<td>14</td>
</tr>
<tr>
<td>#Training</td>
<td>7213</td>
<td>19,151</td>
<td>7274</td>
<td>18,780</td>
<td>24,301</td>
</tr>
<tr>
<td>#Validation</td>
<td>5327</td>
<td>7263</td>
<td>4263</td>
<td>4072</td>
<td>19,281</td>
</tr>
<tr>
<td>#Test</td>
<td>3348</td>
<td>2854</td>
<td>1000</td>
<td>2349</td>
<td>4820</td>
</tr>
</tbody>
</table>

The Edge-IoTset [46] data set identifies and analyzes 14 types of network attack methods. These attacks can be summarized as five threats, namely, DoS/DDoS attacks, Information gathering, Man in the middle attacks, Injection attacks, and Malware attacks. At the same time, the data set also indicates the start time and end time of an attack (or the time point of the attack). Edge-IoTset’s IOT data is generated from various IOT devices (more than 10 types), such as Low-cost digital sensors for sensing temperature and humidity, Ultrasonic sensor, Water level detection sensor, pH Sensor Meter, Soil Moisture sensor, Heart Rate Sensor, Flame Sensor, etc.). The Edge-IoTset dataset also records the IP address of the attacker and the sensor being attacked.

4.2. Experimental Setup

**Baseline Methods** We compare our proposed method with the static KGE and TKGE models in the previous study. The static KGE methods used in this article are TransE [10], DistMult [16], Temporal methods include TTransE [20], HyTE [21], TA-DistMult [5].

**Evaluation Protocol** According to the previous work [29], we filtered out the triplet data that meets our experimental requirements from these four Industrial Internet of Things data sets, and divided them into training sets, verification sets, and test sets according to the characteristics of the data, which are 80%/10%/10%, respectively [47–50]. We report the average countdown ranking (MRR) and Hits@1/3/10 (the proportion of correct test cases in the top 1/3/10) to measure the performance of our model and the comparison model [51–55]. The calculation formula of MRR is as follows:

\[
MRR = \frac{1}{|S|} \sum_{i=1}^{|S|} \frac{1}{\text{rank}_i} = \frac{1}{|S|} \left( \frac{1}{\text{rank}_1} + \frac{1}{\text{rank}_2} + \cdots + \frac{1}{\text{rank}_{|S|}} \right)
\]  

where \( S \) is the set of triples, \(|S|\) is the number of triples, and \(\llbracket \text{rank}_i \rrbracket \_j \) refers to the link prediction ranking of the \( i \)-th triplet.

HITS@n refers to the average proportion of triples ranked less than \( n \) in the link prediction [56,57]. The specific calculation method is as follows:

\[
\text{HITS@n} = \frac{1}{|S|} \sum_{i=1}^{|S|} \mathbb{I}(\text{rank}_i \leq n)
\]

where the symbols involved in the above formula are the same as those involved in the MRR calculation formula, and \( \mathbb{I}(\cdot) \) is the indicator function (if the condition is true, the function value is 1, otherwise it is 0). \( n \) usually takes the values of 1, 3, and 10.

**Model configuration** Exclude empty accounts that have not been used for a long time in the MOOC-Ub data set, and use the filtered triples and time for verification. The coefficient \( \alpha \) is adjusted from 0.1 to 0.9, with a step size of 0.1. The batch size is set to 1024. The training epoch is limited to 50, which is sufficient to converge in most cases. The embedding dimension is set to 200 to be consistent with the setting of Jin et al. [30]. The baseline results are also from Jin et al. [30].

**Selection of data in Edge-IoTset dataset** Since the Edge-IoTset dataset not only contains the data described in Section 4.1, it also contains 61 flow features and two new attributes, which are stored together in a CSV file. We use python programs to extract the data information we need from it, including: The IP addresses of the network attacker...
and the attacked sensor, the attack methods of different attackers on the sensor, and the time when the network attack started and ended. We regard the attacker’s IP address as the subject entity set, the attacked sensor (IP address) as the object entity set, and the attack method as the predicate relationship set. The above data sets are used to form triples of knowledge graphs. Due to the different time units of network attacks (milliseconds, seconds, minutes, and hours), therefore, the time data is dimensionless. Use the Z-score normalization method:

\[
t_{b}^{*} = \frac{t_{b} - \mu_{b}}{\sigma_{b}}, \quad t_{e}^{*} = \frac{t_{e} - \mu_{e}}{\sigma_{e}}
\]  

where \(\mu_{b}\) and \(\mu_{e}\) are the mean values of \(t_{b}\) and \(t_{e}\) of the same time unit category, respectively. \(\sigma_{b}\) and \(\sigma_{e}\) are the standard deviations of \(t_{b}\) and \(t_{e}\) in the same time unit category, respectively.

### 4.3. Experimental Results

Tables 3 and 4 report the link prediction results of the TBRm and baseline methods on the four IIoT data sets. On the data set in Table 3, the static KGE method lags far behind TA-DistMult or TTransE, because it cannot capture time dynamics. It can also be observed that the performance of all static KGE methods is usually better than that of HyTE. We believe that this is because HyTE slices the dynamic knowledge map with time into multiple static knowledge maps, which are represented independently on each static map, and lack coherence in time updates. Table 3 also shows that TBRm is significantly better than other baseline methods in BPLP and MOOC-Ub.

Table 3. Results (percentages) on the BPLP and MOOC-Ub datasets.

<table>
<thead>
<tr>
<th>Method</th>
<th>BPLP</th>
<th>MOOC-Ub</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MRR</td>
<td>Hits@1</td>
</tr>
<tr>
<td>TransE</td>
<td>16.47</td>
<td>4.62</td>
</tr>
<tr>
<td>DistMult</td>
<td>23.54</td>
<td>5.66</td>
</tr>
<tr>
<td>TTransE</td>
<td>10.63</td>
<td>13.14</td>
</tr>
<tr>
<td>TA-DistMult</td>
<td>9.61</td>
<td>7.25</td>
</tr>
<tr>
<td>TBRm</td>
<td>8.72</td>
<td>22.63</td>
</tr>
</tbody>
</table>

The best results are shown in bold.

Table 4. Results (percentages) on the NFT and IE-IoTD datasets.

<table>
<thead>
<tr>
<th>Method</th>
<th>NFT</th>
<th>IE-IoTD</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MRR</td>
<td>Hits@1</td>
</tr>
<tr>
<td>TransE</td>
<td>33.67</td>
<td>17.26</td>
</tr>
<tr>
<td>DistMult</td>
<td>32.53</td>
<td>17.64</td>
</tr>
<tr>
<td>TTransE</td>
<td>49.57</td>
<td>32.98</td>
</tr>
<tr>
<td>HyTE</td>
<td>45.39</td>
<td>26.35</td>
</tr>
<tr>
<td>TA-DistMult</td>
<td>50.26</td>
<td>40.28</td>
</tr>
<tr>
<td>TBRm</td>
<td><strong>30.04</strong></td>
<td>35.99</td>
</tr>
</tbody>
</table>

The best results are shown in bold.

We also observed in Table 4 that TBRm’s performance on NFT and IE-IoTD is not always the best, especially on NFT. In fact, this is due to the excessive concentration of time information participating in the NFT data set. The excessive concentration of time information makes the time information data interval carried smaller, making it more difficult to distinguish triples. However, the Hyte method of slicing the time series knowledge
map by time has a good performance on this data set. Although TBRm performs better on other data sets with a more balanced distribution of time information, how to solve this shortcoming of TBRm is a meaningful direction for further research.

Table 5 shows the percentage of TBRm’s four performance indicators on the Edge-IIoTset dataset. We can see that in addition to Hits@1, TBRm has achieved relatively high scores in the tests of the other three performance indicators. Analyzing the reasons, it was found that in the Edge-IIoTset data set, the number of subject entities of the network attacker is too large, however, there are only 14 kinds of attacks that act as predicates, and the gap between subject entity set and predicate relation set is too obvious. Therefore, in the process of link prediction, the probability of successful prediction for the first time will become very small, and the percentage value of Hits@1 will decrease accordingly.

Table 5. The performance of TBRm tested on the Industrial Internet of Things.

<table>
<thead>
<tr>
<th>Method</th>
<th>Edge-IIoTset</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MRR</td>
</tr>
<tr>
<td>TBRm</td>
<td>34.68</td>
</tr>
</tbody>
</table>

5. Conclusions

Describing and inferring knowledge graphs with time constraints is a challenging problem. In this paper, we exploit the mapping mechanism to address this problem, assuming that future facts can be predicted from historical facts. The proposed TBRm selects future facts based on known facts that appeared in the past. The results show that TBRm has good performance in predicting future facts in TKGs. We propose a time-constrained relational mapping method to learn temporal knowledge graph fact representations that can be used in conjunction with current link prediction scoring function methods. Experiments on five temporal knowledge graph data demonstrate the effectiveness of the method and its feasibility on the Industrial Internet of Things.

Author Contributions: Conceptualization, K.C. and C.Z.; methodology, K.C. and C.Z.; software, C.Z.; validation, C.Z.; formal analysis, C.Z.; investigation, C.Z.; resources, K.C.; data curation, K.C.; writing—original draft preparation, C.Z.; writing—review and editing, K.C. and C.Z.; visualization, K.C.; supervision, K.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Intelligent IOT and Integrated security of industrial information physics systems, Key projects of the National Natural Science Foundation of China, 2022-01-01 to 2026-12-31, grant number 62133014. And The APC was funded by National Natural Science Foundation of China.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References


34. Shao, P.; Zhang, D.; Yang, G.; Tao, J.; Che, F.; Liu, T. Tucker decomposition-based temporal knowledge graph completion. Knowl.-Based Syst. 2022, 238, 107841. [CrossRef]  


38. Tareq, I.; Elbagoury, B.M.; El‑Regaily, S.; El‑Horbaty, E.S.M. Analysis of ToN‑IoT, UNW‑NB15, and Edge‑IoT Datasets Using DL in Cybersecurity for IoT. Appl. Sci. 2022, 12, 9572. [CrossRef]


43. Gyrrard, A.; Boudaoud, K. Interdisciplinary IoT and Emotion Knowledge Graph-Based Recommendation System to Boost Mental Health. Appl. Sci. 2022, 12, 9712. [CrossRef]


45. Semmler, N. Data‑Driven Transfer Optimizations for Big Data in the Industrial Internet of Things; Technical University of Berlin: Berlin, Germany, 2022.


