JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 39, 1421-1436 (2023)
DOI: 10.6688/JISE.202311_39(6).0011

Inconsistency Detection for Spatiotemporal Knowledge Graph with Entity Semantics and Spatiotemporal Features

XIAO-WEN ZHANG[1,2], JING SHAN[2], WEN-YI TANG[1], LI YAN[2]
AND ZONGMIN MA[2,+]
[1]*State Key Laboratory of Air Traffic Management System and Technology*
Nanjing, 210007 P.R. China
[2]*College of Computer Science and Technology*
Nanjing University of Aeronautics and Astronautics
Nanjing, 211106 P.R. China

Knowledge graph (KG) can model and manage the massive metadata, have received a lot of attention in recent years. Information in the real world is not always static, aim to model and manage the dynamic information (*e.g.*, time interval, location), some research works for spatiotemporal KG have been proposed. Due to the spatiotemporal knowledge is constantly changing, data operations will be more frequent in the process of spatiotemporal KG construction and management, therefore, inconsistency may exist in spatiotemporal KG. The current work on handling the inconsistencies in spatiotemporal KG mainly focuses on providing consistency constraints and fixing rules when operating the spatiotemporal KG, and no work on actively detects the existing inconsistency in spatiotemporal KG. In this paper, we discuss and summarize the inconsistency in spatiotemporal KG firstly. Then we design algorithms to extract inconsistency semantic feature in spatiotemporal KG, and finally we propose a spatiotemporal KG inconsistency detection model. The experimental results show that our method is scientific and effective.

Keywords: spatiotemporal knowledge graph, entity semantics, inconsistency detection, multi-classification, feature extraction

1. INTRODUCTION

With the development of big data intelligence, to modeling and management the massive metadata, knowledge graph (KG) has been proposed and received a lot of attention from researchers. KG describes knowledge in the form of triple (s, p, o), where s and o denote *subject* and *object*, p represents *predicate* and linked *subject* to *object*. However, knowledge in the real world is not always static, the modeling and management of dynamic knowledge (*e.g.*, time interval and spatial location) is still a challenge. Spatiotemporal knowledge describes the temporal evolution of spatial objects over time [1], such as an aircraft in a certain spatial position at a certain time point. Spatiotemporal KG contains massive information of time interval and location, modeling and management spatiotemporal knowledge with static KG data model will leads to a large number of fragmented temporal and spatial knowledge in KG [2]. Therefore, spatiotemporal KG has been regarded by researchers as a special kind of KG different from the static KG, and some work related to spatiotemporal KG has been proposed (including spatiotemporal KG representation [2-5], spatiotemporal KG storage and querying [6-8], spatiotemporal KG datasets and tools [9-12]).

Received June 23, 2022; revised October 11 & December 12, 2022; accepted January 26, 2023.
Communicated by De-Nian Yang.
[+] Corresponding author: zongminma@nuaa.edu.cn

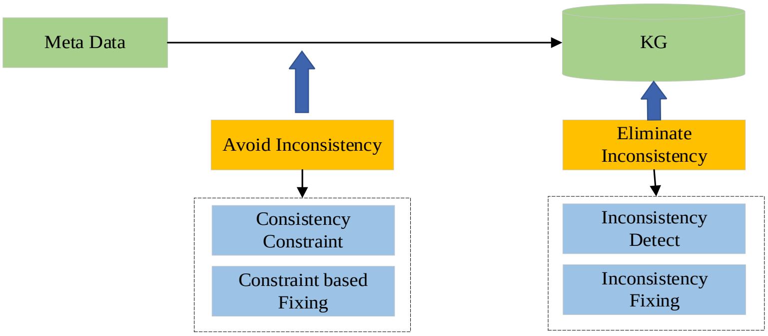


Fig. 1. Handle inconsistencies in KG.

For the popularization and application of large-scale KG in the industry, there have been a lot of work for the KG construction. Early KG construction was mostly based on manual annotation and expert system [13], this method cannot handle the massive metadata. We are in an era of information explosion, due to the multi-source heterogeneity of data sources when construct the KG, inconsistency is inevitable in KG. Inconsistencies in KG will damage the downstream tasks based on KG. The current work of handling inconsistencies in KG mainly focuses on two parts: (i) reduce the possible inconsistency in data operation or construction of KG; (ii) actively eliminate the existing inconsistencies in KG, and as shown in Fig. 1.

There has been some relevant work on dealing with inconsistencies in static KG [14-16], including minimizing the generation of inconsistencies and actively eliminating inconsistencies in KG. To deal with inconsistencies in spatiotemporal KG, several efforts work on detection and repair of inconsistent constraint during data operations [19, 24, 25]. To the best of our knowledge, there is no research work on detecting the inconsistencies that exist in the constructed spatiotemporal KG, rather than the inconsistencies occurred in data processing. This paper tries to fill this gap. We first discuss the inconsistencies of spatiotemporal KG and summarize them into three categories. We then propose an entity semantic density calculation method and two feature extraction algorithms to mine inconsistent semantic features. We finally design a multi-classifier to identify inconsistencies in spatiotemporal KG. We conducted experiments on datasets containing spatiotemporal information, and the experimental results prove the effectiveness of the proposed method.

The organization of this paper is as follows. Section 2 introduces the related work on inconsistency detection and spatiotemporal KG. Section 3 introduces the spatiotemporal KG model and discusses its inconsistency. Section 4 introduces the spatiotemporal KG inconsistency detection method. Section 5 provides details of the experimental and analyzes the experimental results. Section 6 concludes the paper.

2. RELATED WORK

This paper's work is related to spatiotemporal KG and inconsistency detection. Spatiotemporal knowledge represents the temporal and spatial changes of spatiotemporal entities. Temporal knowledge can be discrete or continuous, and spatial knowledge can be a place or a route. For spatiotemporal knowledge representation, [2] proposed a spatiotemporal data modeling method called STT. STT designs an ontology knowledge framework of spatial, which can associate temporal tuple (s, p, o):[t] and spatial information with ontology syntax, where t can be *time point* or *time interval*. [3] further integrates spatiotemporal information into static triple, and proposed spatiotemporal quadruple (s, p, o, τ) named stRDF. stRDF represents spatiotemporal objects as quantifier-free formulas in first-order logic of *linear constraints*. In this method, the temporal constraint $\tau$of spatiotemporal quadruple represents the *time point*, and uses a conjunction of *linear constraints* to represent the *location*. [4] proposes the spatiotemporal data model g^{st}-Store and the tuple formal is (s, p, o, l, t), where l and t represent *location* and *time interval* respectively. [5] proposed to fix spatiotemporal information to *predicate*, and proposed a spatiotemporal tuple (s, p: <t, l>, o) named stRDFS. stRDFS additionally provides the syntax of spatiotemporal RDF and spatiotemporal algebraic operations. Based on the research of spatiotemporal KG data model, other research work related to spatiotemporal KG has been proposed: spatiotemporal KG storage and querying [6-8], spatiotemporal KG datasets and tools [9-12], *etc.*

For static KG inconsistency detection, some work is based on rich schema information [21, 22] and others are based on entity relationship paths [14-16, 23]. Furthermore, inconsistency detection based on representation learning has gradually gained some attention. TransE [26] is a representative model based on translation, which designs the objective function of $s + p = o$. CKRL [20] introduces confidence in the triple learning process, and learns confidence to enhance the model's ability to learn KG structural information. For spatiotemporal KG inconsistency detection, the current work focuses on reducing the generation of inconsistencies by providing consistency constraints and immediately repairing inconsistencies. [25] proposed spatiotemporal semantic dependency constraints described by formal language, and designs an inconsistency checking algorithm. In order to model spatiotemporal knowledge more accurately and flexibly, [24] proposed to model spatiotemporal information based on RDF and formally proposed spatiotemporal RDF data model, and then presented corresponding consistency constraints and repair methods. [19] proposed a new spatiotemporal KG data model and designs corresponding consistency constraints for data operations on spatiotemporal KG.

At present, the handling inconsistency of spatiotemporal KG is still focused on the consistency constraint and repair, and there is no work to actively detect inconsistencies in spatiotemporal KG. This paper attempts to solve this problem. In this paper, we first discuss the inconsistency of spatiotemporal KG, and propose a method to detect the inconsistency in spatiotemporal KG.

3. INCONSISTENCY OF SPATIOTEMPORAL KNOWLEDGE GRAPH

In this section, we first give the details of spatiotemporal data model, then we discuss the inconsistencies of spatiotemporal KG.

3.1 Spatiotemporal Information in Knowledge Graph

Spatiotemporal knowledge is dynamic and related to temporal and spatial changes. Considering that the expression ability of *time interval* is stronger than that of *time point*, we decide to describe temporal knowledge in the form of *time interval*. For spatial knowledge, previous research work [3-5] uses (*longitude*, *latitude*) or *MBR* to represent *location*, where *MBR* denotes a rectangular area. This form ignores altitude information. In this paper, we use spatiotemporal tuple (s, p [t: <t_s, t_e>, l: <l_{lon}, l_{lat}, l_{alt}>], o) to describe spatiotemporal instance and give the definition as follows:

Definition 1 (Spatiotemporal data model): Spatiotemporal data is represented in the formal of STKG = (s, p [t: <t_s, t_e>, l: <l_{lon}, l_{lat}, l_{alt}>], o). Here,

1) s and o represent *subject* and *object*.
2) p represents *predicate* with spatiotemporal information.
3) t: <t_s, t_e> represents the *time interval*, t_s denotes the time start and t_e denotes the time end. If $t_s \neq t_e$, the t_s must be less than t_e and if $t_s = t_e$, the t can be seen as a *time point*.
4) l: [<l_{lon}, l_{lat}, l_{alt}>] represents the spatial information of spatiotemporal tuple, and l_{lon} denotes the longitude, l_{lat} denotes the latitude, l_{alt} denotes the altitude.

Such a spatiotemporal data model can be seen as a labeled and directed graph. The *subject* and *object* exist as vertices in KG, and the directed edge *predicate* points from the *subject* to the *object*. Each *subject*→*predicate*→*object* describes a spatiotemporal KG triple. The spatiotemporal information is assigned to the *predicate*. Now we give two examples of spatiotemporal KG (Meteorological monitoring and Air traffic) in Figs. 2 and 3, to intuitively display the details of spatiotemporal KG.

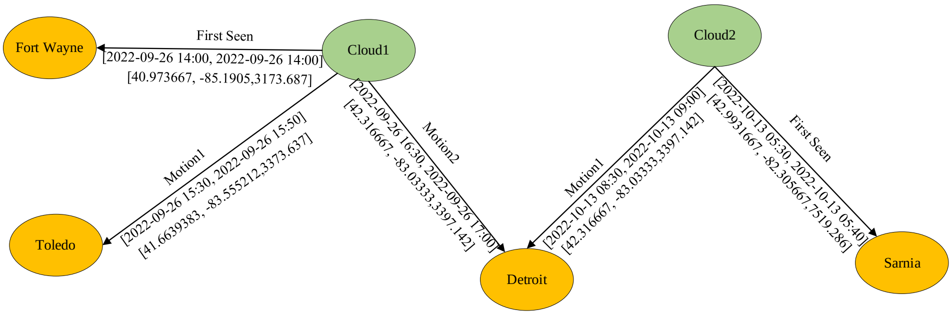


Fig. 2. Example of spatiotemporal KG (Meteorological monitoring).

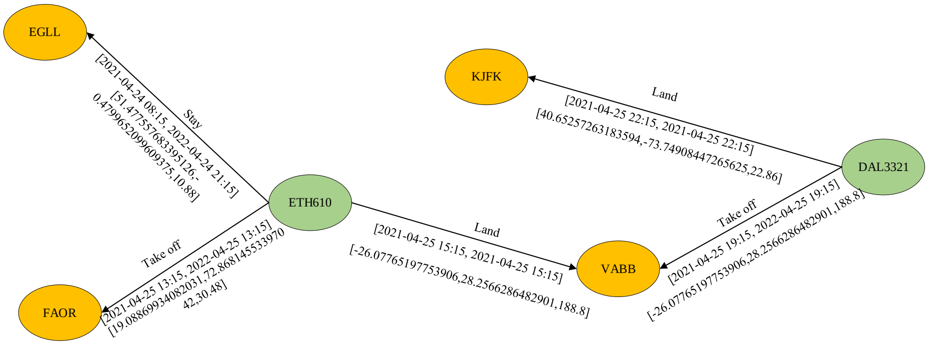


Fig. 3. Example of spatiotemporal KG (Air traffic).

Example 1: Meteorological information is naturally suitable for modeling with spatiotemporal KG. One of the main observation targets of meteorological monitoring is cloud movement. In Fig. 1, there are two clouds in the picture: *cloud1* and *cloud2*. *Cloud1* is first monitored in Fort Wayne City, and the monitoring time is 2022-09-26 14:00, coordinate is [40.97366, –85.1905], cloud height is 3173.687km. Following the motion tracks of *motion1* and *motion2*, *cloud1* successively passes through Toledo City and Detroit City. Furthermore, the monitoring information of *cloud2* is also shown in this figure.

Example 2: Air traffic is a representative spatiotemporal data intensive field. In Fig. 3, we take flight information as an example. For ETH610 takes off instance, representing flight ETH610 take off from the airport FAOR at 2021-04-25 13:15, and the longitude of FAOR is 19.08869934082031, latitude is 72.86814553397042, altitude is 30.48. ETH610 stay instance represents flight ETH610 stay at the airport EGLL from 2021-04-24 08:15 to 2021-04-24 21:15. On this basis, we further discuss the inconsistency of this spatiotemporal KG based on the air traffic field with more intensive spatiotemporal information.

3.2 Inconsistency of Spatiotemporal Knowledge Graph

Spatiotemporal KG includes massive dynamic knowledge. Spatiotemporal KG consists of two parts: entity and spatiotemporal edge. Discuss their inconsistencies separately, for an entity, when the entity label is wrong, there may be multiple vertices that look different but refer to the same entity actually. For the spatiotemporal edge, inconsistency is mainly caused by time interval conflict and spatiotemporal conflict. An entity cannot appear in different places at a certain time, there may occur time interval inconsistency between the same *subject*. Correspondingly, two entities cannot exist in the same location at the same time, there may be spatiotemporal inconsistency between the two spatiotemporal instances.

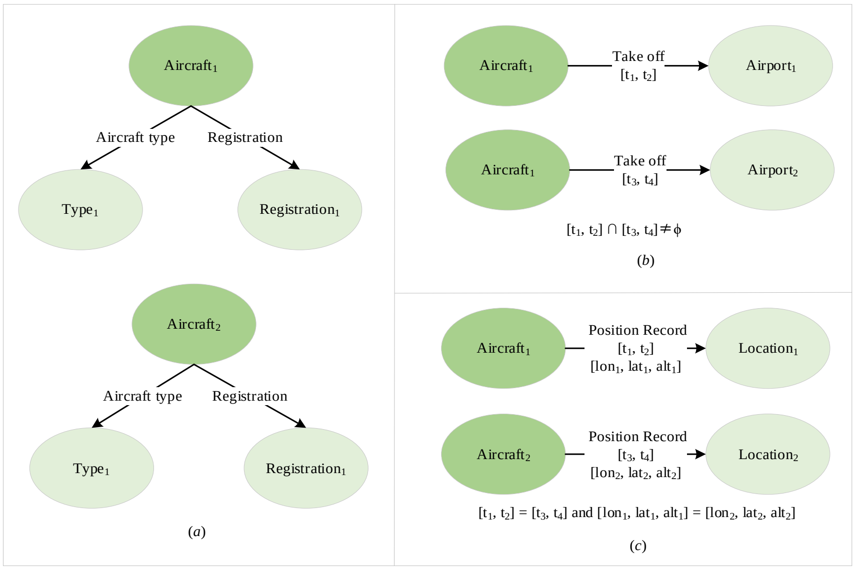


Fig. 4. Three types inconsistencies in spatiotemporal KG.

Based on the above considerations, we divide the inconsistencies of spatiotemporal KG into three types: (i) entity inconsistency; (ii) time interval inconsistency; and (iii) spatiotemporal inconsistency. The graphical illustration of three kinds of inconsistency is as follows.

Fig. 4 (a) illustrates the entity inconsistency. Two different aircrafts in spatiotemporal KG have the same aircraft type and registration, where the registration is the aircraft tail number and shall be unique, so these two different aircrafts in spatiotemporal KG actually refer to the same aircraft in the real world. Fig. 4 (b) describes two take off instances of an aircraft, but $[t_1, t_2] \cap [t_3, t_4] \neq \phi$. Actually, it is impossible for an aircraft to take off at two different airports at the same time, so there has time interval inconsistency. Fig. 4 (c) shows two instances of position recording during the flight, the temporal information and spatial information of two aircrafts are equal. This situation is obviously incorrect, so we believe that there has spatiotemporal inconsistency for two position record instances.

4. INCONSISTENCY DETECTION OF SPATIOTEMPORAL KNOWLEDGE GRAPH

In this section, we introduce the method of inconsistency detection of spatiotemporal KG and give technical details.

4.1 Multiple Classifiers of Inconsistency Detection

The goal of inconsistency detection is to identify inconsistent instances from spatiotemporal KG and recognize the type of inconsistency. The entities and spatiotemporal relations in spatiotemporal KG contain rich semantic information, which can help us judge the correctness of spatiotemporal tuples. Based on the above considerations, we treat inconsistency detection of spatiotemporal KG as a multi classification problem and design a multiple classifiers of inconsistency detection. The structure of the multiple classifiers is shown as follows.

Fig. 5 illustrates the details of multiple classifiers. We use different strategies to mine the entity feature and spatiotemporal relation feature in spatiotemporal KG: entity semantic density calculation method to obtain the semantic density of entities; time interval feature extraction algorithm and spatiotemporal feature extraction algorithm to obtain the spatio-

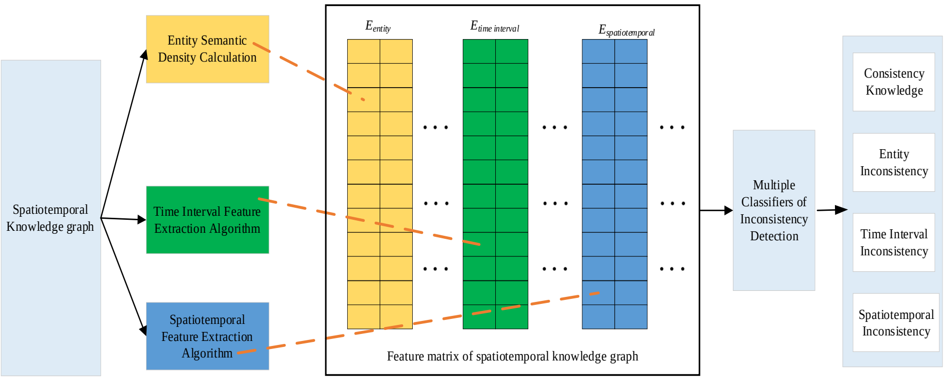


Fig. 5. Multiple classifiers of inconsistency detection.

temporal feature of spatiotemporal relations. Then we get the feature matrix of entities and spatiotemporal relations. We use classification strategies as the classifier model, and output the classification results of inconsistency including: (i) consistency knowledge; (ii) entity inconsistency; (iii) time interval inconsistency; (iv) spatiotemporal inconsistency.

For entity feature, based on the graph structure of spatiotemporal KG, the entities association strength is related to the resource flow between different vertices. This can be the basis for judging whether the entity is correct. For time interval feature and spatiotemporal feature catch, must define time interval and spatiotemporal conflict rules firstly, and then design the corresponding feature extraction algorithms to vectorize the spatiotemporal relations.

4.2 Entity Semantic Density Calculation

In spatiotemporal KG, entity (*subject* and *object*) exists as a vertex, *predict* exists as a directed edge, and the spatiotemporal KG can be seen as a directed graph. In such a directed graph, the number of relations that enters or leaves vertex is countable, the process of relations leaving the *subject* and entering the *object* is called resource value flow. Resource value indicates the number of paths between two vertices, so the resource value of *subject* or *object* is fixed. Entity semantic information is obtained by the resource value flowing from *subject* to *object*. In spatiotemporal tuple (s, p [t: <t_s, t_e>, l: <l_{lon}, l_{lat}, l_{alt}>], o), vertex s can diverge to vertex o through multiple relation paths, and named the resource value of s is semantic density of entity pair (s, o). Then we give the formal definition of semantic density.

Definition 2 (Semantic Density): Given two non-adjacent entities in spatiotemporal KG: subject s to object o, the multiple paths from s to o are expressed as (pa_1, pa_2, …, pa_l), each path pa_i is expressed as $s \xrightarrow{p_1} E_1 \xrightarrow{p_2} \cdots \xrightarrow{p_{n-1}} E_{n-1} \xrightarrow{p_k} o$, where E_i represents the entity set at the i^{th} step, k is the number of relations on the path pa_i. For each entity e (including subject s to object o) on the path pa_i, the semantic density $S(e)$ can be described as follows:

$$S(e) = (1-\theta) \sum_{e' \in E_{i-1}(\cdot, e)} \frac{S(e') \cdot W_{e'e}}{|E_i(e', \cdot)|} + \theta. \tag{1}$$

In Definition 2, $E_{i-1}(\cdot, e)$ represents the forward vertex of edge p_i entering vertex e, $E_i(e', \cdot)$ represents the backward vertex of edge p_i leaving vertex e, $W_{e'e}$ represents the weight of e' to e and depends on the number of relations between e' and e. To avoid the infinite path iterations caused by paths closed loops, we set the probability parameter θ to control the resource flow from e' to e. At the beginning of calculation, all vertices are given the same resource value, and after n-step iteration, the semantic density $S(o)$ is considered as the semantic information implied in the entity pair (s, o). Fig. 6 shows the semantic density calculation process of entity pair. There are multiple paths from A to E, so the semantic density of entity pair (A, E) is high. In contrast, no path can be observed from F to G, so the semantic density of entity pair (F, G) is almost 0. Specifically, the semantic density calculation of entity pair (s, o) utilizes path constraint resource allocation (PCRA) method. PCRA is inspired by resource allocation and considered that the number of resource value of vertex s is certain, and can be allocated to o through multiple paths connected to o. The resource value flowing from the *subject* to the *object* is semantic density of entity pair (*subject*, *object*).

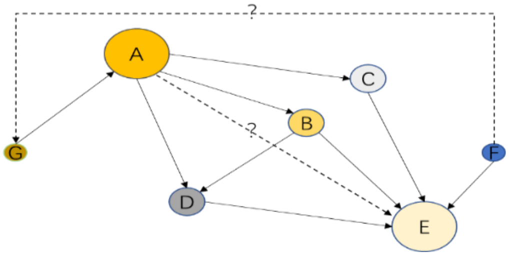


Fig. 6. Semantic density calculation process of entity pair.

4.3 Time Interval Feature Extraction Algorithm

For spatiotemporal tuple (s, p [t: <t_s, t_e>, l: <l_{lon}, l_{lat}, l_{alt}>], o), the obvious difference from static triple (s, p, o) is the spatiotemporal tuple has timeliness. The lifetime of spatiotemporal tuple is < t_s, t_e >. The time interval of different spatiotemporal tuples lacks direct correlation, so the catch of time interval feature must consider the semantic correlation between entities. Then, we design the logical rule of time interval conflict as follows:

Definition 3 (Logical rule of time interval conflict)**:** Given two spatiotemporal tuples (s_1, p_1 [t: <t_{s1}, t_{e1}>, l: <l_{lon1}, l_{lat1}, l_{alt1}>], o_1) and (s_2, p_1 [t: <t_{s2}, t_{e2}>, l: <l_{lon1}, l_{lat1}, l_{alt1}>], o_1), equals (s_1, s_2) $\models$ ([t_{s1}, t_{e1}] $\cap$ [t_{s2}, t_{e2}] = ϕ).

Definition 3 represents that for given two spatiotemporal tuples, when their *subject* is equal, their time interval should not cross. This rule is for linear recording in the generalized time interval, and limits the disorganization of spatiotemporal instances in time interval granularity. Then we give the time interval feature extraction algorithm in Algorithm 1.

The input of Algorithm 1 is spatiotemporal knowledge graph *KG* and spatiotemporal tuple *tup*. At the beginning, *KG* is initialized and loaded all tuples in *KG* into memory. Then, circularly count the mutually exclusive information between each spatiotemporal tuple and *tup*, and add it to the set *time*. Finally, according to the set *time*, determined whether there has spatiotemporal tuple mutually exclusive with *tup* in *KG*, and the result v is returned in the form of vector. Analyze the time and space complexity of Algorithm 1, if the spatiotemporal KG contains n spatiotemporal triples, m edges with spatiotemporal information and k time interval inconsistent spatiotemporal triples. Then the time complexity of Algorithm 1 is $n + 2m + k$, the space complexity of algorithm 1 is $n + m + k$.

For example, to get the time interval feature of spatiotemporal tuple *TIIT*, lines [1-2] first construct spatiotemporal tuple set T, then extract time interval information t_s, t_e, and finally construct time interval triplet *En* with *subject*. Lines [3-8], according to Definition 3, check whether there are any time interval conflicts in *En*. Finally, the time interval feature of *TIIT* is returned by lines [9-10].

Algorithm 1: Extract the time interval feature of spatiotemporal knowledge graph

```
Input: spatiotemporal knowledge graph KG, spatiotemporal tuple tup
Output: time interval feature of tup v
1:  $T \leftarrow ReadAllTriples(KG)$//Load the spatiotemporal KG as the triple set.
2:  $En(s, t_s, t_e) \leftarrow ExtractTime(T)$//Extract the time start, time end and form a triple with s.
```

```
***Calculate the time interval mutual exclusion information of each subject***
for $tup \in En(e)$ do
    $val \leftarrow GetRules(tup)$ // Statistics of mutually exclusive information
    for $r \in val$ do
        $time.append(r)$
    end for
vector $v \leftarrow Convert(time)$ // Returns the time interval feature of tup.
return $v$
```

4.4 Spatiotemporal Feature Extraction Algorithm

The location of different spatiotemporal tuples has no direct connection. Considering that location conflicts always occur in the intersected time intervals, we think that the spatial conflict detection should be based on time interval. Moreover, for two spatiotemporal tuples with spatiotemporal conflict, the locations are not necessarily identical. For example, when the distance is less than the threshold during the flight of two aircrafts, it is called dangerous flight, and the position record instances of two aircrafts will have spatiotemporal conflict. Based on the above considerations, we propose a spatiotemporal conflict logic rule with spatiotemporal threshold as follows:

Definition 4 (Logical rule of spatiotemporal conflict): Given two spatiotemporal tuples $(s_1, p_1\ [t: <t_{s1}, t_{e1}>, l: <l_{lon1}, l_{lat1}, l_{alt1}>], o_1)$ and $(s_2, p_2\ [t: <t_{s2}, t_{e2}>, l: <l_{lon2}, l_{lat2}, l_{alt2}>], o_2)$, $(abs\,(t_{s2} - t_{s1}) < \alpha_s \,\|\, abs\,(t_{e2} - t_{e1}) < \alpha_e \,\|\, abs\,(l_{lon2} - l_{lon1}) > \delta \,\|\, abs\,(l_{lat2} - l_{lat1}) < \gamma \,\|\, abs\,(l_{alt2} - l_{alt1}) < \tau) \models consistent\,(s_1, s_2)$.

In Definition 4, α_s and α_e are the time interval thresholds for start time and end time, δ is the longitude threshold, γ is the latitude threshold, τ is the altitude threshold. If the rule equation about several thresholds is satisfied, there is no spatiotemporal conflict between spatiotemporal tuples. Moreover, various thresholds can be flexibly adjusted or even 0 as required. Then we give the spatiotemporal feature extraction algorithm as follows:

```
Algorithm 2: Extract the spatiotemporal feature of spatiotemporal knowledge graph
Input: spatiotemporal knowledge graph $KG$, spatiotemporal tuple $tup$, time interval threshold $\alpha_s$
    and $\alpha_e$, longitude threshold $\delta$, latitude threshold $\gamma$, altitude threshold $\tau$
Output: spatiotemporal feature of $tup$ $v$
$T \leftarrow ReadAllTriples(KG)$ // Load the spatiotemporal KG as the triple set.
$En(s, t_s, t_e, l_{lon}, l_{lat}, l_{alt}) \leftarrow ExtractSpatiotemporal(T)$ // Extract the time start, time end, lon-
    gitude, latitude and altitude form a tuple with s.
***Calculate the spatiotemporal mutual exclusion information of each subject***
For $tup \in En(e)$ do
    $val \leftarrow GetRules(tup)$ // Statistics of mutually exclusive information
    if $abs(e\text{-}val) < (\alpha_s, \alpha_e, \delta, \gamma, \tau)$ do
        $spatiotemporal.append(r)$
    end for
vector $v \leftarrow Covert(spatiotemporal)$ // Returns the spatiotemporal conflict tuples.
return $v$
```

The input of Algorithm 2 includes spatiotemporal knowledge graph *KG*, spatiotemporal tuple *tup*, longitude threshold δ, latitude threshold γ, altitude threshold τ, time interval threshold α_s and α_e. At the beginning, *KG* is initialized and loaded all tuples in *KG* into memory. Next, circularly count the mutually exclusive information between each spatiotemporal tuple and *tup*, and add it to the set *spatiotemporal*. Finally, determined whether there has spatiotemporal tuple mutually exclusive with *tup* in *KG*, and the result *v* is returned in the form of vector. Analyze the time and space complexity of Algorithm 2, if the spatiotemporal KG contains *n* spatiotemporal triples, *m* edges with spatiotemporal information and *k* spatiotemporal inconsistent spatiotemporal triples. Then the time complexity of algorithm 1 is $n + m\ (m + 1 - k)$, the space complexity of algorithm 1 is $n + m + k$.

For example, to get the spatiotemporal feature of spatiotemporal tuple *STIT*, line [1-2] first construct spatiotemporal tuple set *T* and spatiotemporal triplet *En*. Lines [4-8] check whether there are any spatiotemporal inconsistencies in *En*. Finally, lines [9-10] return spatiotemporal feature of *TIIT*.

5. EXPERIMENT

In this section, we design experiments to verify the method proposed in this paper and analyze the experimental results.

5.1 Datasets

The datasets used in the experiments are OpenSky and YAGO2S. OpenSky is a global flight information dataset, which contains a large number of flight data (*e.g.*, flight number, takeoff time, longitude). We randomly choose 20,000 of entity to construct the OpenSky-20K. Because OpenSky dataset has no inconsistent information, we mark some inconsistent data to supplement it. The final version of the dataset OpenSky-20K is named OpenSky-20K-STI. In addition, YAGO2S includes the data extracted from Wikipedia, which are combined with the taxonomy of WordNet and attached relevant facts and entities in spatiotemporal dimensions. YAGO2S with 244799610 facts is a spatiotemporal dataset expanded from the static dataset YAGO1. Note that, however, YAGO2S does not contain inconsistent information. Following the inconsistent dimensioning protocol in [21], we label 1000 inconsistent tuples for each of the three types of inconsistencies.

We use two datasets OPENSKY-20K-STI and YAGO2S-8K-STI and the statistics are shown in Table 1.

Table 1. Statistics of datasets.

Dataset	#Ent	#Rel (type)	Consistency	Entity Inconsistency	Time Interval Inconsistency	Spatiotemporal Inconsistency
OPENSKY-20K-STI	20,000	16	80,000	10,000	20,000	20,000
YAGO2S-8K-STI	7870	13	6000	1000	1000	1000

5.2 Experimental Settings

The prototype runs on Windows 10 professional system. In addition, the configuration of CPU and RAM are i7-9700 3.0GHz and 32G. All algorithms designed in this paper are implemented in Python 3.6.

We choose TransE [26], TransR [27] and CKRL [20] as the baselines. CKRL has achieved great performance and TransE has always been a classical baseline for error detection task, TransR optimizes the defect that TransE cannot efficiently model complex relations. We choose classification strategies *SoftMax*, *Random Forest* (RF) and Gradient Boosting Decision Tree (GBDT) to carry out follow-up experiments on the best classification strategy. For *SoftMax*, the option is multinomial and the optimization solver is *newton-cg*. The tolerance for stopping criteria is set among {1e-4, 1e-5, 1e-6}. For RF, the estimator number of train process is {50, 100, 150} and *max_features* = 5. For GBDT, the max depth of tree is 8, the max feature is sqrt, the estimator is 80, and the learning rate is 0.001. In the process of entity semantic training, we set θ to express the probability that the resource randomly flows to the successor of a node. We set $\theta = 0.15$ according to the ResourceRank algorithm.

For experimental task, we first evaluate the performance of multiple classifier based on different classification strategies, and the sub classifiers including: (i) for *consistency*; (ii) for *entity inconsistency*; (iii) for *time interval inconsistency*; and (iv) for *spatiotemporal inconsistency*. We select the commonly used evaluation indicators in the classification task: Precision, Recall and F1-Score as the evaluation indicators. To more intuitively observe the performance of the classifier in each class of inconsistency detection, we generate and analyze the *confusion matrix* of the classifier. We use *Receiver Operating Characteristic Curve* (*ROC*) to measure the generalization ability of the model.

5.3 Experimental Results

We expect to find a classification strategy that best fits our method, and can effectively implement the inconsistency detection task of spatiotemporal KG. For this purpose, we designed four-classifier for inconsistency classification, and carried out comparative experiments with different classification strategies.

Table 2. Evaluation results with several classification strategies.

Datasets	OpenSky-20K-STI			YAGO2S-8K-STI		
Metric	Precision	Recall	F1Score	Precision	Recall	Score
TransE	0.804	0.776	0.790	0.758	0.749	0.753
TransR	0.816	0.795	0.805	0.763	0.752	0.757
CKRL	0.846	0.814	0.830	0.769	0.745	0.757
RF	0.936	0.914	0.924	0.899	0.854	0.876
SoftMax	0.954	0.915	0.932	0.917	0.852	0.883
GBDT	**0.955**	**0.917**	**0.934**	**0.924**	**0.870**	**0.896**

Experimental results are shown in Table 2. It is shown that the four-classifiers based on GBDT have achieved the best performance, which is 1-3 percentage points higher than each indicator of RDF and SoftMax. The Precision of the four-classifier based on GBDT in dataset OpenSky-20K-STI has reached 95.5% and the Recall has reached 91.7%, it directly proves the ability of entity semantic and spatiotemporal feature in the inconsistency detection of spatiotemporal KG. It should be noted that the indicators of all models in OpenSky-20K-STI are lower than those in YAGO2S-8K-STI. This is because OpenSky-

20K-STI has more abundant spatiotemporal semantic information. On the other hand, the inconsistent tuple size also affects the final indicators. Thus, the next experiments are mainly based on OpenSky-20K-STI to ensure the persuasiveness of the experimental results. Considering that GBDT is more adaptable to entity semantic and spatiotemporal features, we use GBDT as a classification strategy to further test the recognition ability of the four-classifier for each inconsistency.

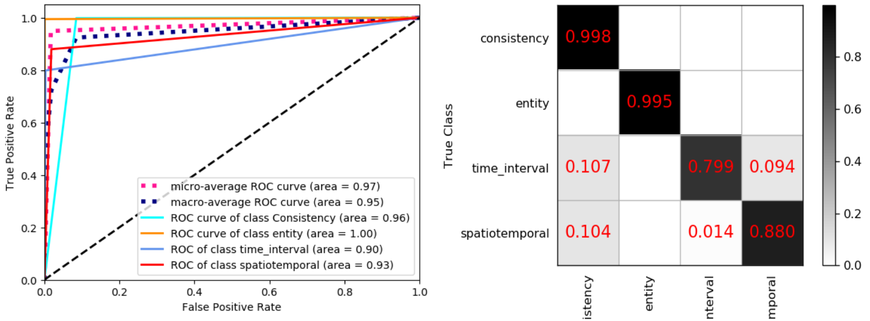
Table 3. Evaluation results of sub classifier with classification strategy GBDT.

Classifier	Precision	Recall	F1-Score
Sub classifier of consistency	0.95	1.00	0.97
Sub classifier of entity inconsistency	0.99	0.99	0.99
Sub classifier of Time interval Inconsistency	0.99	0.78	0.87
Sub classifier of spatiotemporal inconsistency	0.89	0.89	0.89
Complete classifier	**0.9556**	**0.9179**	**0.9342**

To verify the recognition ability of the semantic density calculation and two feature extraction algorithms for three types of inconsistencies, we have carried out an independent experiment of sub classifier in four-classifier. The experimental results are shown in Table 3.

We can observe that each indicator of the sub classifier of consistency has reached more than 95%, which shows that the inconsistency features catcher is effective, and will not lead to the misjudgment of consistency knowledge as inconsistency knowledge. Each indicator of entity inconsistency sub classifier has reached 99%, which proves the importance of entity semantic information in inconsistency detection, and verifies the effectiveness. The Precision of time interval inconsistency sub classifier has reached 99%, but the Recall and F1-score are 78% and 87% respectively, indicating that some temporal conflicts have not been detected. We believe that all temporal conflicts cannot be effectively detected only by time interval conflict logic rule, therefore, we consider further increasing the types of temporal conflict logic rules in future work. The indicators of spatiotemporal inconsistency sub classifier are 89%, and there is room for further improvement. We believe that by adjusting the several thresholds in the spatiotemporal conflict logic rule, we can flexibly identify the spatiotemporal inconsistency.

To more intuitively observe the recognition performance of the four-classifier for various inconsistencies, we generate the Roc and confusion matrix of the four-classifier based on GBDT. The experimental results are shown in Fig. 7.



(a) ROC. (b) Confusion matrix.

Fig. 7. ROC and confusion matrix of the four-classifier based on GBDT.

In Fig. 7 (a), we observe that our model achieves a higher false positive rate when detecting time interval inconsistency. In Fig. 7 (b), we find that some time interval conflicts are incorrectly identified as spatiotemporal conflicts. This is because some inconsistent knowledge satisfies both time interval conflict logic rule and spatiotemporal conflict logic rule. However, we only find a single class label when we label the dataset. In the follow-up work, we consider using multiple labels and multiple classifiers method to detect the inconsistency of spatiotemporal KG. The same situation can be observed on the ROC of spatiotemporal inconsistency, the false positive rate of spatiotemporal inconsistency is only lower than that of time interval inconsistency. Combined with Fig. 7 (b), furthermore, some spatiotemporal inconsistency instances are incorrectly identified as time interval inconsistency.

6. CONCLUSIONS

In this paper, for inconsistency detection of spatiotemporal KG, we first discuss the inconsistencies of spatiotemporal KG and identify them into three categories: (i) entity inconsistency; (ii) time interval inconsistency; and (iii) spatiotemporal inconsistency. Then, we design entity semantic density calculation to capture entity semantic information. We propose the time interval conflict logic rule and the spatiotemporal conflict logic rule, then we design the time interval feature extraction algorithm and the spatiotemporal feature extraction algorithm. In addition, we define the problem of inconsistency detection of spatiotemporal KG as the problem of multi classification of inconsistent knowledge, and design a four-classifier for (i) consistency knowledge; (ii) entity inconsistency; (iii) time interval inconsistency; and (iv) spatiotemporal inconsistency. As far as we know, this is the first attempt to detect the inconsistency of spatiotemporal KG.

The experimental results prove the effectiveness of the proposed method. Our model achieves 95% Precision and 91% Recall. It shows the importance of entity semantic information, time interval features and spatiotemporal features for inconsistency detection of spatiotemporal KG. The experimental results of ROC and confusion matrix prove that the entity semantic density calculation and feature extraction algorithms can effectively capture the inconsistent information in the spatiotemporal KG, and also point out the next research direction for us. In future work, we will consider using an adaptive method to optimize the spatiotemporal conflict logic rule, and try to use multiple labels and multiple classifiers method to strengthen the detection ability to the inconsistency of complex spatiotemporal knowledge.

ACKNOWLEDGMENT

The work was supported in part by the State Key Laboratory of Air Traffic Management System and Technology (SKLATM202104) and the National Natural Science Foundation of China (62176121).

REFERENCES

1. F. Zhang, Q. Z. Lu, Z. J. Du, X. Chen, and C. Cao, "A comprehensive overview of RDF for spatial and spatiotemporal data management," *The Knowledge Engineering*

Review, Vol. 36, 2021, p. e10.

2. A. Sheth and M. Perry, "Traveling the semantic web through space, time, and theme," *IEEE Internet Computing*, Vol. 12, 2008, pp. 81-86.
3. M. Koubarakis and K. Kyzirakos, "Modeling and querying metadata in the semantic sensor web: The model stRDF and the query language stSPARQL," in *Proceedings of the 7th Extended Semantic Web Conference on Research and Applications*, 2010, pp. 425-439.
4. D. Wang, L. Zou, and D. Y. Zhao, "GST-store: querying large spatiotemporal RDF graphs," *Data and Information Management*, Vol. 1, 2017, pp. 84-103.
5. L. Zhu, N. Li, and L. Y. Bai, "Algebraic operations on spatiotemporal data based on rdf," *ISPRS International Journal of Geo-Information*, Vol. 9, 2020, p. 80.
6. M. Perry, A. Estrada, S. Das, and J. Banerjee, "Developing GeoSPARQL applications with oracle spatial and graph," in *SSN-TC/OrdRing@ISWC*, 2015, pp. 57-61.
7. P. Nikitopoulos, A. Vlachou, C. Doulkeridis, and G. A. Vouros, "Parallel and scalable processing of spatio-temporal rdf queries using spark," *GeoInformatica*, Vol. 25, 2021, pp. 623-653.
8. D. M. Wu, H. Zhou, J. M. Shi, and N. Mamoulis, "Top-k relevant semantic place retrieval on spatiotemporal rdf data," *The VLDB Journal*, Vol. 29, 2020, pp. 893-917.
9. J. Hoffart, F. M. Suchanek, K. Berberich, and G. Weikum, "Yago2: A spatially and temporally enhanced knowledge base from wikipedia," *Artificial Intelligence*, Vol. 194, 2013, pp. 28-61.
10. B.-H. Tran, N. Aussenac-Gilles, C. Comparot, and C. Trojahn, "Semantic integration of raster data for earth observation: An RDF dataset of territorial unit versions with their land cover," *ISPRS International Journal of Geo-Information*, Vol. 9, 2020, p. 503.
11. A. Vaisman and K. Chentout, "Mapping spatiotemporal data to rdf: A sparql endpoint for brussels," *ISPRS International Journal of Geo-Information*, Vol. 8, 2019, p. 353.
12. G. M. Santipantakis, A. Glenis, K. Patroumpas, A. Vlachou, C. Doulkeridis, G. A. Vouros, N. Pelekis, and Y. Theodoridis, "Spartan: Semantic integration of big spatio-temporal data from streaming and archival sources," *Future Generation Computer Systems*, Vol. 110, 2020, pp. 540-555.
13. H. Paulheim, "Knowledge graph refinement: A survey of approaches and evaluation methods," *Semantic Web*, Vol. 8, 2017, pp. 489-508.
14. A. Melo and H. Paulheim, "Detection of relation assertion errors in knowledge graphs," in *Proceedings of Knowledge Capture Conference*, Vol. 22, 2017, pp. 1-8.
15. P. Lin, Q. Song, Y. H. Wu, and J. X. Pi, "Discovering patterns for fact checking in knowledge graphs," *Journal of Data and Information Quality*, Vol. 11, 2019, pp. 1-27.
16. S. B. Jia, Y. Xiang, X. J. Chen, and K. Wang, "Triple trustworthiness measurement for knowledge graph," in *Proceedings of World Wide Web Conference*, 2019, pp. 2865-2871.
17. O. Udrea, D. R. Recupero, and V. Subrahmanian, "Annotated RDF," *ACM Transactions on Computational Logic*, Vol. 11, 2010, pp. 1-41.
18. Z. M. Ma, L. Y. Bai, Y. Ishikawa, and L. Yan, "Consistencies of fuzzy spatiotemporal data in xml documents," *Fuzzy Sets and Systems*, Vol. 343, 2018, pp. 97-125.
19. L. Y. Bai, J. Y. Wang, X. F. Di, and N. Li, "Fixing the inconsistencies in fuzzy spatiotemporal rdf graph," *Information Sciences*, Vol. 578, 2021, pp. 166-180.

20. R. B. Xie, Z. Y. Liu, F. Lin, and L. Y. Lin, "Does William Shakespeare really write Hamlet? knowledge representation learning with confidence," in *Proceedings of AAAI Conference on Artificial Intelligence*, Vol. 32, 2018, pp. 4954-4961.
21. H. Paulheim and C. Bizer, "Improving the quality of linked data using statistical distributions," *International Journal on Semantic Web and Information Systems*, Vol. 10, 2014, pp. 63-86.
22. Z. H. Syed, M. Röder, and A.-C. N. Ngomo, "Unsupervised discovery of corroborative paths for fact validation," in *Proceedings of the 18th International Semantic Web Conference*, 2019, pp. 630-646.
23. E. Munoz, "On learnability of constraints from rdf data," in *Proceedings of the 13th International Conference on Semantic Web: Latest Advances and New Domains*, 2016, pp. 834-844.
24. J. Y. Wang, X. F. Di, J. M. Liu, and L. Y. Bai, "A constraint framework for uncertain spatiotemporal data in rdf graphs," in *Advances in Natural Computation, Fuzzy Systems and Knowledge Discovery*, Vol. 1, 2020, pp. 727-735.
25. G. del Mondo, M. A. Rodríguez, C. Claramunt, L. Bravo, and R. Thibaud, "Modeling consistency of spatio-temporal graphs," *Data and Knowledge Engineering*, Vol. 84, 2013, pp. 59-80.
26. A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," *Advances in Neural Information Processing Systems*, Vol. 26, 2013, pp. 2787-2795.
27. Y. K. Lin, Z. Y. Liu, M. S. Sun, Y. Liu, and X. Zhu, "Learning entity and relation embeddings for knowledge graph completion," in *Proceedings of AAAI Conference on Artificial Intelligence*, Vol. 29, 2015, pp. 2181-2187.

Xiao-Wen Zhang is working toward a Ph.D. degree in the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, China. His research interests include spatiotemporal knowledge representation and reasoning, and fuzzy knowledge fusion.

Jing Shan is a Research Associate at Nanjing University of Aeronautics and Astronautics, China, engaging in intelligent system for task allocation, optimization algorithm and application of intelligent transportation system.

Wen-Yi Tang is currently an Engineer at State Key Laboratory of Air Traffic Management System and Technology, Nanjing, China. His research interest locates in the cross-cutting area between air traffic control and deep learning.

Li Yan is a Full Professor at Nanjing University of Aeronautics and Astronautics, China. Her research interests mainly include big data processing, knowledge graph, spatiotemporal data management, and fuzzy data modeling. She has published more than fifty papers on these topics. She is the author of three monographs published by Springer.

Zongmin Ma is a Full Professor at Nanjing University of Aeronautics and Astronautics, China. His research interests include big data, knowledge engineering, and computational intelligence. He has published more than two hundred papers and five monographs with Springer on these topics. He is a Fellow of the IFSA.