A Theoretical Framework for Temporal Graph Warehousing with Applications

Annie Y.H. Chou[1], Frank S.C. Tseng[2]
Dept. Computer and Information Science, ROC Military Academy, Kaohsiung, Taiwan, ROC[1]
Dept. Information Management, National Kaohsiung Univ. of Science and Technology, Kaohsiung, Taiwan, ROC[2]

***Abstract*—The evolution of data management systems has witnessed a paradigm shift towards dynamic and temporal representations of relationships. Graph databases, positioned as key players in managing highly-connected data with a fundamental requirement for relationship analysis, have recognized the need for incorporating temporal features. These features are crucial for capturing the temporal dynamics inherent in various applications, offering a more comprehensive understanding of relationships over time. This theoretical exploration emphasizes the importance of incorporating temporal dimensions into graph data warehousing for contemporary applications. Temporal features introduce a dynamic dimension to graph data, enabling a more nuanced understanding of relationships and patterns over time. The integration of temporal features in graph data management and analysis not only addresses the dynamic nature of contemporary applications but also contributes to enhanced modeling and analytical capabilities.**

Keywords—Data warehousing; graph database; graph warehousing; social computing; temporal data

I. Introduction

The pervasive expansion of social media platforms has led to the establishment of a global conduit, as explored in study [8], amalgamating a plethora of data, insights, and principles concerning product details, societal norms, and leisure suggestions. Consequently, this proliferation has facilitated the emergence of influential online personalities within the digital community. Comprehending the dynamics of this cyber-community is imperative for addressing societal challenges such as counter-terrorism, cyber warfare, and cyberbullying, while concurrently fostering adept management practices conducive to human welfare. Hence, an internal framework for delineating social networks is indispensable for advancing scholarly understanding and practical engagement with the complexities of the digital realm [10]. The evolution of data management systems has seen a paradigm shift towards dynamic and temporal representations of relationships.

Graph databases are aimed at dealing with highly-connected data that comes with an intrinsic need for relationship analysis [1] [2]. Being a prominent player in this landscape, they have increasingly recognized the need for temporal features to capture the temporal dynamics inherent in various applications. When we have a specific starting point or at least a set of points to start with (nodes with the same label), they are well equipped to traverse relationships. And the obtained graph structures have given rise to numerous business opportunities and applications leveraging the networking infrastructure [14] [18]. Instances comprise customer relationship management (CRM), cloud computing and its services, enterprise resource planning (ERP), supply chain management (SCM), and business intelligence (BI).

In the realm of contemporary data management, the inclusion of temporal features has emerged as a critical aspect [7][9][11][12][15][16][19][22][25][30][31][32], especially in the context of graph data [34][35]. Incorporating temporal features in graph data management and analysis allows for a more accurate modeling of influence dynamics, capturing changes in social structures and information dissemination patterns. We give some examples to explain this point:

1) Social networks and influence dynamics: Social networks are inherently temporal, with relationships evolving over time. Incorporating temporal features in graph data allows for a more accurate modeling of influence dynamics, capturing changes in social structures and information dissemination patterns.

2) Financial systems and transactional analysis: In financial applications, understanding the temporal aspects of transactions is crucial. Temporal features enable the identification of patterns related to fraudulent activities, market trends, and the evaluation of investment strategies.

3) Healthcare and patient journey analysis: Temporal features play a pivotal role in healthcare analytics by providing insights into the temporal progression of diseases, treatment effectiveness, and patient outcomes. This temporal perspective enhances the precision of predictive modeling and decision support systems.

This theoretical exploration delves into the importance of incorporating temporal dimensions into graph data warehousing for contemporary applications. Temporal features provide a dynamic dimension to graph data, enabling a more nuanced understanding of relationships and patterns over time. This paper discusses the implications of temporal features for various domains, outlines challenges in their integration, and highlights potential benefits for on-line analytical processing.

II. Related Works

The primary distinction between relational and graph databases lies in their respective methodologies for storing relationships among entities or objects. Traditionally, relational databases use predefined relationship type structures (i.e., by relationship table definitions) to store relationships, while in a graph database, relationships are stored at the individual object level [33]. The data in a relational database can be deduced to create a graph database as we have discussed in [29]. Commercial graph database vendors, like Neo4j (http://neo4j.com), also offer ETL (Extraction, Transformation, and Loading) tools for transforming relational database into their graph database products (e.g., https://neo4j.com/labs/etl-tool/1.5.0/) [17].

For example, in Fig. 1(a), there is a relational database modeled by the traditional Entity-Relationship Model (E-R Model), which concerns the relationship type <Enroll> (in the relation Enroll) between entity types Student and Course.

Conversely, in a graph database, relationships between any two instances of objects can be dynamically encoded. Illustrated in Fig. 1(b), beyond the relationships encompassed within the <Enroll> relationship type, a graph database permits users to introduce relationships such as "*roommate*" between John and Alex, "*classmate*" between Tom and Joe, "*friend*" between Mary and Alex, and a directed relationship labeled "*prerequisite*" from Algorithm to Fintech.

The relationships depicted in Fig. 1(a) exhibit a greater degree of "staticity," as they remain unchanged throughout the entirety of the semester. In contrast, Fig. 1(b) encompasses a greater diversity of "dynamic" relationships, reflecting various aspects that may evolve or change over time. Based on Fig. 1(b), suppose the relationships have been extended with temporal features and transformed into the temporal graph as shown in Fig. 2, where Tom studied the Fintech at t_3 (a course offered by Dr. Liu), but Alex studied the course at t_8 (offered by Dr. Li). Rigorously speaking, the system should not deduce Tom and Alex as classmates even though they have studied the same course Fintech, as they did not meet each other in their classes (different class timings).

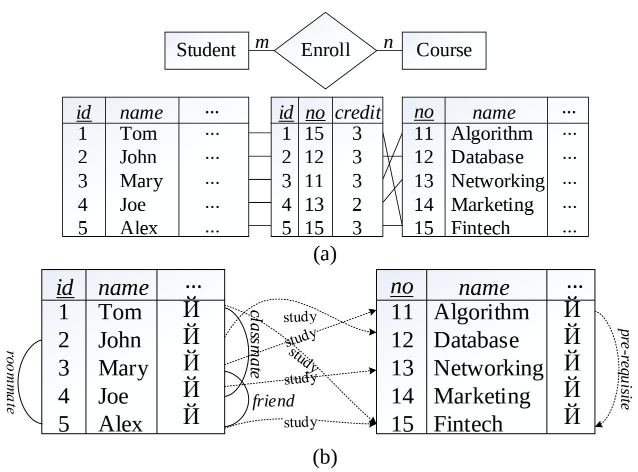

Fig. 1. (a) Conventional relational database models relationships between entity types. (b) Graph database models relationships between object instances.

A graph consists of two fundamental components:

1) A set of vertices (also known as nodes): Representing selected objects.

2) A set of edges (also called links): Representing relationships between objects. Directed edges are represented as *arcs*, while undirected edges are depicted as *edges*.

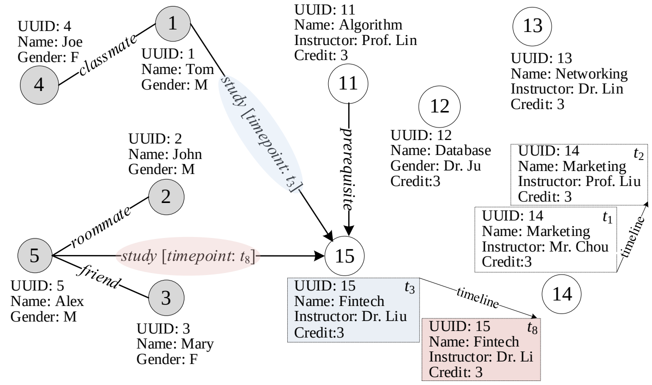

Fig. 2. A temporal graph database used to model the relationships between object instances.

Traditionally, social networks are depicted using a graph structure that encompasses data relating to all participating objects and their interconnections. These foundational concepts establish the framework for describing the structure of a social network using a graph. Additional information pertaining to vertices and edges, acquired through assignment or derived from measurement, is referred to as *properties* (or *attributes*). Properties can be assigned by humans or calculated from the graph or other properties.

As posited by Vicknair *et al.* (2010) [33] and Ho, Wu, & Liu (2012) [15], the interactions and engagements of netizens within social networks can be effectively represented within established graph database management systems. When cyber-communities are structured within a graph database framework, the identification of opinion leaders across various domains or cyber warriors can be facilitated through analytical inquiries, comparing their interstitial behaviors and relationships. Moreover, given the maturity of relational data warehousing technologies, synergizing these methodologies enables the exploration of business intelligence inherent within social networks. This convergence, known as *social business intelligence*, harnesses the combined capabilities of graph databases and relational data warehousing technologies to conduct online analytical processing (OLAP) and unveil hidden insights within social networks.

Sahu, *et al*. (2019) [20] performed an extensive survey study on how graphs are used in practice, and revealed surprising facts of the increasing prevalence across many application domains. Zhao *et al*. (2011)[36] introduced the concept of *graph cube*, and presented a novel data warehousing model designed to facilitate OLAP (On-Line Analytical Processing) queries on extensive multidimensional networks. Sakr *et al*. (2021) [21] even posited that the future of data processing is a big graph. Through the incorporation of temporal features into graph databases, we posit that graph-related systems will emerge as highly potent tools for managing interconnected data in modern applications.

Different from the traditional model of social networks, we assume there are *temporal properties* in some vertices and lines, and then propose a rigorous model for the challenge of consistent graph management and graph data warehousing by the following observations:

1) Temporal graph data are indispensable in social substance and related contexts [29], as social networks form and evolve gradually through the operations of social processes along the timeline of every participant [3]. The contextual frameworks in which social networks are formed play a pivotal role in comprehending the dynamics of network inception and the ensuing implications for individuals, groups, and organizations. For example, Brodka *et al.* (2011, 2013) [5] [6] utilize temporal data to devise a methodology aimed at elucidating the evolutionary trajectories of groups within social networks. This approach enables the discernment of group dynamics encompassing formation, growing, splitting, shrinking, continuing, merging, and dissolution.

2) Currently, most of the research on social networks has been studied without considering time and the dynamically changing status of objects (and their relationships).

3) Since there is already well-developed temporal data management in contemporary relational database management systems, it is the right time to embark on the study of business intelligence hidden in *temporal graphs*.

Graph warehouses, distinct from graph database management systems, encompass rich semantics pertaining to object relationships, facilitate efficient feature extraction and indexing functionalities, and offer flexible summarization approaches from diverse perspectives. This enables the grouping or clustering of subgraphs, thereby providing nuanced access to business intelligence pertaining to network types. Through a comprehensive integration of attribute aggregation and structural summarization within multidimensional networks, their approach yields insights and structure-enriched aggregate networks across varied multidimensional spaces.

However, despite the advancements in relational temporal technologies, their formal extension or adaptation to the domain of graph databases remains largely unexplored. Consequently, temporal considerations have been largely overlooked in the majority of related studies.

In this paper, we elucidate several scenarios to underscore the potential for imprecise analytical outcomes stemming from the absence of temporal considerations. Subsequently, we endeavor to present a formal definition for a temporal graph database model, extending the temporal data warehousing concept into the domain of temporal graph warehousing. Additionally, we provide formal definitions for the fundamental elements of a temporal graph warehouse.

Although temporal graph databases are powerful for dealing with interconnected data, they are not suited for traversing the whole graph when it contains tremendous vertices and relationships from performance viewpoints. Therefore, when we need to analyze temporal graph data based on some criteria, a network or graph is usually generated (for a specific time slot or interval) through a graph query posed on the temporal graph database.

Our principal focus lies in delving into temporal graphs beyond mere structural delineations, aiming instead to encapsulate their evolutionary trajectories over time and discern the underlying social dynamics propelling their transformations. This endeavor is geared towards comprehending the temporal evolution of social networks and elucidating the processes involved in their formation, persistence, and dissolution.

III. A Formal Definition of the Temporal Graph Model

By conceptualizing vertices and edges (including directed *arcs* and undirected *edges*) as entities and connections respectively, objects sharing similar characteristics can be grouped into *object types*, and their connections of similar nature into *relationship types*. If the historical evolution of object types can be meticulously recorded and managed, these object types can be elevated to *temporal object types*. Through such temporal object types, it is envisaged that precise online analytical processing of interlinked object relationships can be facilitated.

Furthermore, given the potential existence of multiple relationships between two objects within a temporal graph, the model ought to manifest as a labeled multi-digraph. Each label corresponds to a distinct relationship imbued with associated attribute values, while each object possesses its own set of attribute values. Labels of identical nature can be categorized under the same relationship types, a framework that can be established through an extension of prior research endeavors [28].

Definition 1: A *temporal graph* is a multi-digraph with labeled vertices and lines (including directed *arcs* and undirected *edges*). Formally, it is an 8-tuple $G_T = (\Sigma_O, \Sigma_R, T, \mathcal{O}, \mathcal{R}, f_s, f_t, \Psi)$, where

1) Σ_O is a set of vertices, representing universal unique identifiers (UUIDs) of all the object instances in G_T.

2) Σ_R is a set of lines, generally denoted (p, q) (or $l(p, q)$ with label l), which include directed *arcs* of UUID pairs $(p \rightarrow q)$ and undirected *edges* of UUID pairs $(p–q)$ of every instance of relationship, such that $p, q \in \Sigma_O$.

3) $T = \{t_1, t_2, \ldots, t_i, \ldots\}$ is a set of time points.

4) $O = \{ O_1^{t_j}(\mathbb{A}_1), O_2^{t_j}(\mathbb{A}_2), \ldots, O_i^{t_j}(\mathbb{A}_i), \ldots, O_n^{t_j}(\mathbb{A}_n)\}$ represents a set of *temporal object types* $O_i^{t_j}$ with schema $\mathbb{A}_i = (A_{i,1}, A_{i,2}, \ldots, A_{i,d(i)})$ of *degree* $d(i)$, such that each $O_i^{t_j}(\mathbb{A}_i) = \{o_{i,1}^{t_j}, o_{i,2}^{t_j}, \ldots, o_{i,k}^{t_j}\}$ contains a set of *temporal objects* of type $O_i^{t_j}$, where $o_{i,k}^{t_j} = (a_{i,1}, a_{i,2}, \ldots, a_{i,d(i)})$ represents an object instance of type $O_{i,}$ at time t_j with the universal unique identifier (UUID) k. Practically, object instances of the same type in a graph database are allowed to have different schemas.

5) $\mathcal{R} = \{R_1(\mathbb{B}_{R_1}), R_2(\mathbb{B}_{R_2}), \ldots, R_i(\mathbb{B}_{R_i}), \ldots, R_m(\mathbb{B}_{R_m})\}$ represents a collection of *relationship types* R_i with a schema $\mathbb{B}_{R_i} = (B_{i,1}, B_{i,2}, \ldots, B_{i,e(i)})$ of degree $e(i)$, such that each $R_i(\mathbb{B}_{R_i}) = \bigcup_{p,q \in \Sigma_O}\{r_{(p,q)}\}$ denotes a set of relationships of type R_i and $r_{(p,q)} = (p, q, b_{i,1}, b_{i,2}, \ldots, b_{i,e(i)})$, $b_{i,k} \in B_{i,k}$, is a relationship instance for a pair of UUIDs (p, q), p and $q \in \Sigma_O$. $\mathbb{B}_{R_i}$ can be empty, which implies that R_i lacks attributes and can also be represented as $R_i(\emptyset)$. For temporal applications, some $\mathbb{B}_{R_i}$ may have at least one attribute, e.g., *time-points*, used to store the active time points of a (p, q) relationship.

6) f_s: $\Sigma_R \rightarrow \Sigma_O$ and f_t: $\Sigma_R \rightarrow \Sigma_O$ are two mappings indicating the *source* and *target* object UUIDs of a relationship UUID pair.

7) $\Psi = \{\psi_{R_1}, \psi_{R_2}, \ldots, \psi_{R_i}, \ldots, \psi_{R_m}\}$ represents a set of mappings, such that ψ_{R_i}: $\Sigma_R \rightarrow \mathbb{B}_{R_i}$ is a mapping returning the tuple of attribute values $(b_{i,1}, b_{i,2}, \ldots, b_{i,e(i)})$ of a relationship (p, q) in Σ_R.

In a *temporal* graph, the presence of vertices and edges can vary over time. A vertex $v \in \Sigma_O$ and an edge $l \in \Sigma_R$ are not necessarily active in all time points. Additionally, a strict consistency condition must be upheld: If an edge $l(p, q)$, which may be directed $l(p \rightarrow q)$ or undirected $l(p - q)$, is active at time point t, then its endpoints p and q should be also active at time t. Formally this is expressed as $t(l(p \rightarrow q)) \subseteq t(p) \cap t(q)$ and $t(l(p - q)) \subseteq t(p) \cap t(q)$, where $t(e)$ denotes a function returning the active set of time points for e.

In this definition, a *temporal graph*, encompasses a variety of objects (e.g., individuals or affiliations) and their multilateral relationships of diverse relationship types (e.g., friendships or spouse relationships). Both objects and relationships may possess varying numbers of attributes. To facilitate graph or network analytics, modern comprehensive graph database management systems (GDBMSs) are furnished with specialized query language constructs for extracting attributes, relationships, or even transitive closures within networks. Users can formulate query statements by effortlessly expressing pattern matching or multi-hop navigation in social networks [8] [13]. However, these GDBMSs do not inherently support functionalities for temporal features. Therefore, our aim is to investigate such temporal features based on formally-defined characteristics and explore methods for simulating these functionalities through time-oriented relations.

To exemplify the concept of our temporal graph data model, we present the temporal graph, denoted as $G_T = (\Sigma_O, \Sigma_R, T, \mathcal{O}, \mathcal{R}, f_s, f_t, \Psi)$ in Fig. 3, where,

1) $\Sigma_O = \{2, 3, 5, 6, 7, 8, 101\}$, means there are 7 UUIDs for six persons and one conference.

2) $\Sigma_R = \{(2, 3), (2, 101), (3, 5), (3, 101), (5, 8), (5, 101), (7, 8), (7, 101), (8, 101), (6, 7), (6, 101)\}$. There are 12 UUID pairs for the relationships between persons and the conference.

3) $T = \{t_1, t_2, \ldots, t_{888}\}$.

4) $\mathcal{O} = \{O_1(\mathbb{A}_1), O_2(\mathbb{A}_2)\} = \{$*Person*(*UUID*, *Name*, *Gender*, *City*, *Affiliation*, *Degree*), *Conference*(*UUID*, *Name*, *Start_Date*, *End_Date*, *City*)$\}$, where *Person*(*UUID*, *Name*, *Gender*, *City*, *Affiliation*, *Degree*) = {{(2, Luna, F, Taipei County, NTPU, MS)$_{t_1}$, (2, Luna, F, New Taipei, NTPU, MS)$_{t_2}$}, (3, Lora, F, Taipei, NTU, PhD)$_{t_2}$, {(5, Tom, M, Changhua, NCUE, MS)$_{t_1}$, (5, Tom, M, Changhua, NCUE, PhD)$_{t_2}$}, (6, May, F, Tainan, NCKU, MS)$_{t_2}$, (7, Ling, F, Tainan, NCKU, PhD)$_{t_2}$, (8, Ren, M, Kaohsiung, NKUST, PhD)$_{t_2}$} is an object type of degree 6, and *Conference*(*UUID*, *Name*, *Start_Date*, *End_Date*, *City*) = {{(101, 2020 ICDE Conference, 2020/04/20, 2020/04/24, 'Dallas, TX')$_{t_1}$, (101, 2021 ICDE Conference, 2021/04/19, 2020/04/22, 'Chania, Crete, Greece')$_{t_2}$} is an object type of degree 5, containing 1 object.

5) $\mathcal{R} = \{R_1(\mathbb{B}_{R_1}), R_2(\mathbb{B}_{R_2})\} = \{$*participate*(*timepoints*), *Friend*($\varnothing$)$\}$, where *participate*(*timepoints*) = {(2, 101, [2020/04/20-2020/04/23]), (3, 101, [2020/04/20-2020/04/23]), (5, 101, [2020/04/22-2020/04/24]), (6, 101, [2020/04/22-2020/04/24]), (7, 101, [2020/04/20-2020/04/24]), (8, 101, [2020/04/21-2020/04/23])}; *Friend*($\varnothing$) = {(2, 3), (3, 5), (5, 8), (6, 7), (7, 8)}.

6) f_s: $\Sigma_R \rightarrow \Sigma_O$ *and* f_t: $\Sigma_R \rightarrow \Sigma_O$ are two mappings indicating the *source* and *target* objects of a relationship. For example, we may obtain $f_s((2, 101)) = 2$ and $f_t((2, 101)) = 101$.

7) $\Psi = \{\psi_{Participate}, \psi_{Friend}\}$ represents a set of mappings, where $\psi_{Participate}((2, 101))$ = ([2020/04/20-2020/04/23]), $\psi_{Participate}((3, 101))$ = ([2020/04/20-2020/04/23]), $\psi_{Participate}((5, 101))$ = ([2020/04/22-2020/04/24]), $\psi_{Participate}((6, 101))$ = ([2020/04/22-2020/04/24]), $\psi_{Participate}((7, 101))$ = ([2020/04/20-2020/04/24]), $\psi_{Participate}((8, 101))$ = ([2020/04/21-2020/04/23]), $\psi_{Friend}((2, 3)) = \varnothing$, $\psi_{Friend}((3, 5)) = \varnothing$, $\psi_{Friend}((6, 7)) = \varnothing$, $\psi_{Friend}((5, 8)) = \varnothing$, $\psi_{Friend}((7, 8)) = \varnothing$.

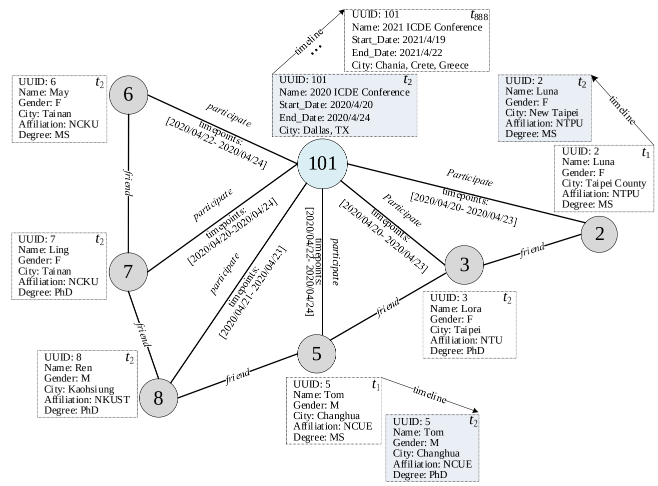

Fig. 3. A temporal graph representing ICDE 2020 Conference and the participants.

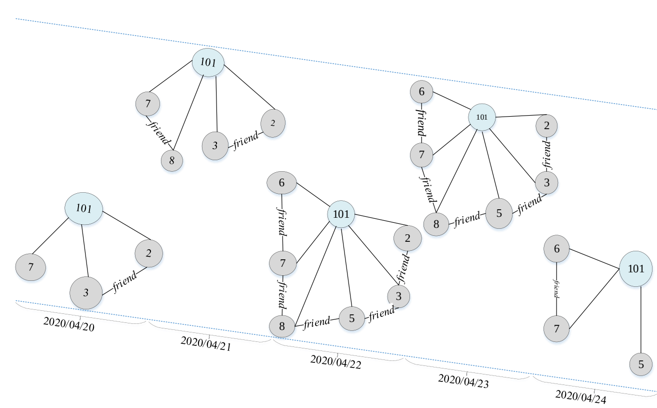

Fig. 4. The graphs of participants in each day of ICDE 2020 Conference.

Based on the *Start_Date* and *End_Date* of object 101, the graph of participants in each day of the *ICDE 2020 Conference* can be illustrated in Fig. 4. These subgraphs can be further summarized as Fig. 5 depicts, where the top belt contains the participants' total aggregation, the middle and bottom belts respectively derive the participants' aggregation by their degree and gender analysis for every conference day. The number in a vertex denotes the number of participants, and the number beside an edge represents the number of relationships. Notice that the vertex with UUID = 5 (Name: Tom) participated in the 2020 ICDE Conference (held at t_2) from 2020/04/22 to 2020/04/24, and the degree of Tom at t_2 is PhD instead of MS. Besides, the city of UUID = 2 (Name: Luna) is 'New Taipei' instead of 'Taipei County' at t_2. Therefore, if the *City* is regarded as a temporal dimension, then 'New Taipei' should be used at t_2 for analytical processing.

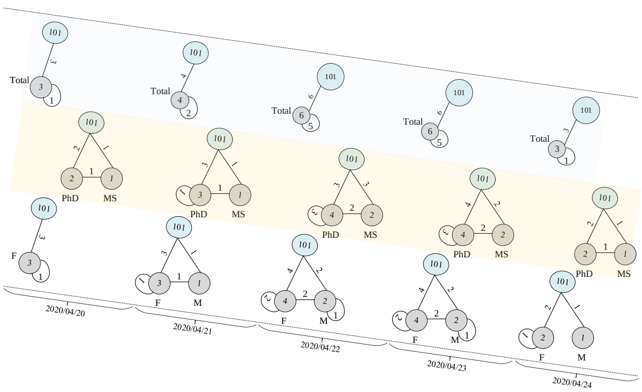

Fig. 5. Three graph summarizations of participants in each day of ICDE 2020 Conference.

When analyzing a large graph, displaying all its details in a single pass becomes impractical. Two overarching strategies exist for examining large graphs:

1) Generating summary descriptions of graphs using statistical methods.

2) Extracting smaller subgraphs based on interesting criteria, such that more sophisticated methods can be conducted.

While both strategies are considered in our study, greater emphasis is placed on the second approach. This involves addressing large graph structures through a divide-and-conquer strategy. In this method, a large graph is partitioned into smaller segments. If a segment remains sizable, it can be further subdivided into subsegments. This iterative process continues until the subsegments become sufficiently small for the application of more intricate methodologies. Descriptions of these smaller graphs prove valuable, offering insights into graph structures. Importantly, these descriptions can be amalgamated to yield a comprehensive understanding of graph structures.

Fig. 6 illustrates various options within the divide-and-conquer strategy. A sample graph is depicted, with different-colored areas denoting regions from which segments can be extracted. The most detailed partition contains vertices within the yellow area. One approach involves extracting graph *parts* to closely examine their interrelationships. Additionally, a decomposition process can be executed at a chosen level, employing vertex or edge partitions based on attribute values, thereby forming a *hierarchy*. When vertices are consolidated into a single vertex, a *reduction* of the graph is achieved.

In Fig. 7, we demonstrate a series of status changes of temporal graph reductions along a timeline when temporal features are introduced and considered for analytical processing. Such temporal tracking graph capability is helpful for group evolution discovery in social networks (e.g., like the work conducted by Bródka *et al.* (2011, 2013) [5] [6]).

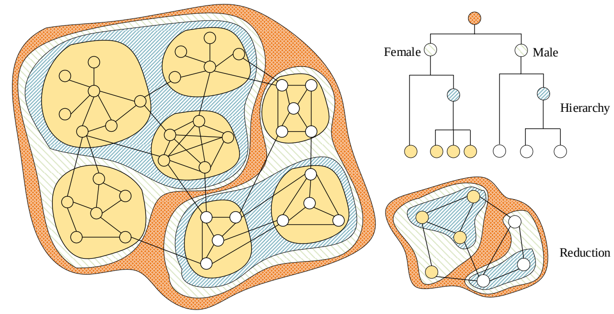

Fig. 6. Graph summarizations by hierarchy decomposition and reduction.

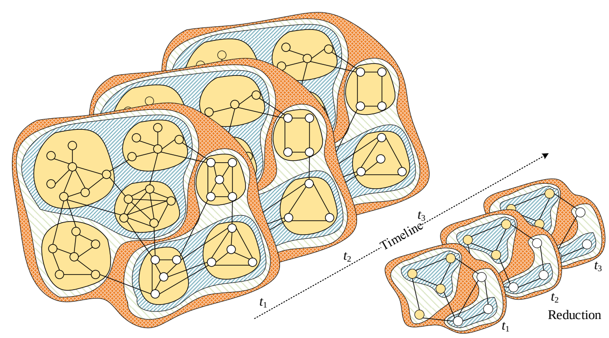

Fig. 7. A series of status changes of graph reduction along a timeline.

IV. Temporal Graph Warehouse Modeling

Since the data in a relational database can be deduced to create a graph database, the basic elements defined in Definitions 1 to 6 can also be employed for both the temporal relational data warehouse and temporal graph warehouse. For example, a temporal dimension can be constructed from attributes of a vertex type, and used for building a temporal graph cube later.

Recall that a line $l(p, q)$, including $l(p \rightarrow q)$ or $l(p\text{–}q)$, can be active in time t, only when two end-vertices p and q are active in time t. Therefore, we do not introduce temporal concepts into relationship types, and the dimensions constructed from attributes of relationship types are treated as ordinary dimensions in our model. As a relationship represents an event or action, the attributes of a line record the history itself. We use the time points contained in relationship types to construct an ordinary *time* dimension, which can be regarded as a counterpart corresponding to the *time* dimension constructed from attribute $\underline{T}$ of the fact table in relational temporal data warehousing (as discussed in Section IV).

In Fig. 8, we draw the temporal dimensions R^{t_1} and R^{t_2} for Taiwan, and their aggregated temporal dimension R^T in Fig. 9. Fig. 10 also depicts another ordinary dimension C for representing a categorization of computer, communication, and consumer electronic products.

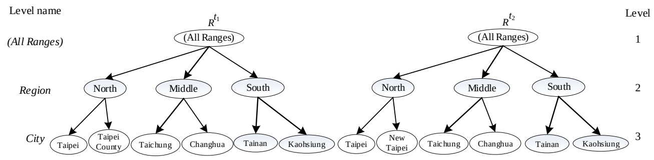

Fig. 8. The temporal dimensions R^{t_1} and R^{t_2} about *Taiwan*.

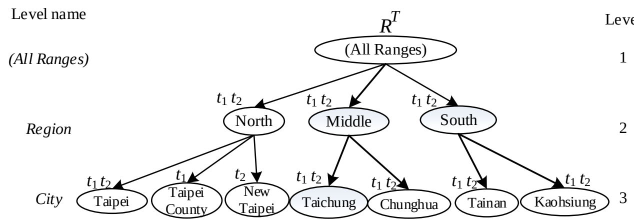

Fig. 9. The aggregated temporal dimension R^T.

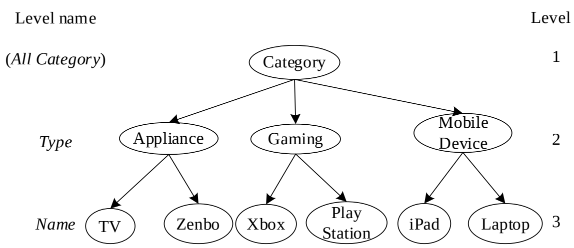

Fig. 10. The dimension C of a categorization of general electronic products.

In the following, we define the basic elements for temporal graph warehousing. These definitions are modified from our previous work of graph data warehousing [28]. Primarily, for facilitating multi-dimensional indexing in temporal graph warehousing, each temporal graph and its subgraphs are all allocated with a distinct identifier.

We establish the fundamental unit of a temporal graph cube, termed a *t-cell*, as follows.

Definition 2: A *t-cell* of time t, denoted $c^t = (t, K, x)$, defined on

- a temporal graph $G_T = (\Sigma_O, \Sigma_R, T, \mathcal{O}, \mathcal{R}, f_s, f_t, \Psi)$,
- a time point $t \in T$, and
- n aggregated temporal dimensions ($D_1^T, D_2^T, \ldots, D_i^T, \ldots, D_n^T$), such that each D_i^T is a hierarchy of keywords, derived from some attribute values of $O_i^t(A_i) \in \mathcal{O}$, $1 \le i \le n$, is a subgraph of G_T, denoted $G_t = (\Sigma^t{}_O, \Sigma^t{}_R, T, \mathcal{O}, \mathcal{R}, f_s, f_t, \Psi)$, pointed (indexed) by a unique identifier x, with the following conditions hold:
  - $K = (K_1, K_2, \ldots, K_i, \ldots, K_n)$, such that $K_i \cap (D_i^t(0) \cup \{‘*’\}) \neq \emptyset$, $1 \le i \le n$,
  - $\Sigma^t{}_O$ ($\Sigma^t{}_O \subseteq \Sigma_O$) contains vertices of type O_i^t, which have attribute values $c_i \in K_i$ at time t.
  - $\Sigma^t{}_R$ ($\Sigma^t{}_R \subseteq \Sigma_R$) contains all relationships (p, q) in G_t, such that these relationships have an attribute *timepoint* containing time t and $p, q \in \Sigma^t{}_O$.

In essence, a t-cell c^t contains a subgraph of G_T, generated through a graph query rooted in $(K_1, K_2, \ldots, K_n)$ for time t, with the subgraph referenced by x.

Example 1: Based on the graph depicted in Fig. 3, an example *t-cell* of time ‘2020/04/23’, denoted c^t = (2020/04/23, ({2020 ICDE Conference}, {F, M}, {*}), x), defined on three aggregated temporal dimensions (C, S, R^T), where C is a dimension of conference names, S is a dimension of gender, and R^T is the dimension depicted in Fig. 9. The subgraph with their multilateral relationships (e.g., friend relationship) pointed by x can be illustrated in Fig. 11.

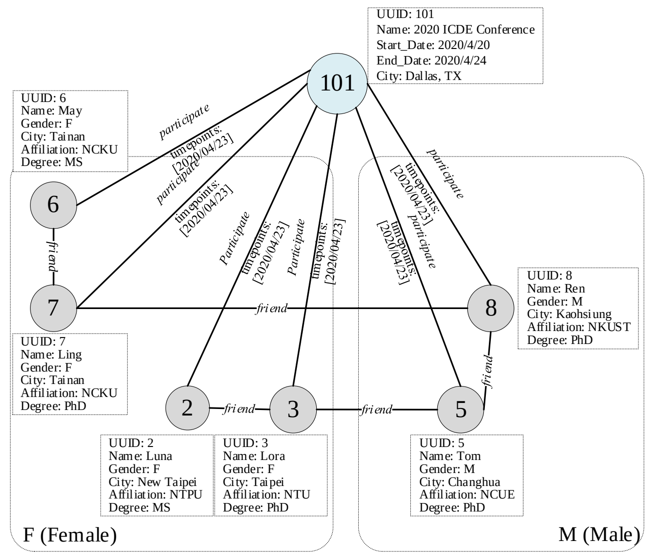

Fig. 11. A *t-cell* c^t = (2020/04/23, ({2020 ICDE Conference}, {F, M}, {*}), x) of time ‘2020/04/23’.

Definition 3: A *t-cell* $c^t = (t, K, x)$, defined over n aggregated temporal dimensions ($D_1^T, D_2^T, \ldots, D_i^T, \ldots, D_n^T$) is termed an m-d *t-cell*, $0 \le m \le n$, if and only if there exist exactly m non-summary members K_i (i.e., $K_i \neq \{*\}$). When $m = n$ and $c_i \in D_i^t(h_i)$, where h_i denotes the height of D_i^t, for all $1 \le i \le n$, then c^t is referred as a *base t-cell*; otherwise, c^t is termed a *non-base t-cell*.

Definition 4: An n-dimensional i-d t-cell $a^t = (t, (a_1, a_2, \ldots, a_n), x_a)$ serves as a *parent* to another n-dimensional j-d t-cell $b^t = (t, (b_1, b_2, \ldots, b_n), x_b)$, if and only if the following conditions are met:

1) $i = j - 1$,

2) There exists exactly one index k, such that a_k is the parent of b_k in D_k^t and $a_l = b_l$ for all $l \neq k$, $1 \le l \le n$.

3) The graph indexed by x_b is a subgraph of the graph indexed by x_a.

Definition 5: A *temporal graph cube* $GC_T = (T, G_T, (D_1^T, D_2^T, \ldots, D_i^T, \ldots, D_n^T))$ for $G_T = (\Sigma_O, \Sigma_R, T, \mathcal{O}, \mathcal{R}, f_s, f_t, \Psi)$ defined over n aggregated temporal dimensions ($D_1^T, D_2^T, \ldots, D_i^T, \ldots, D_n^T$), is a cube composed of all t-cells in $\{c^t{}_i = (t, K, x_i) \mid t \in T(0), c^t{}_i \in T(0) \times (\times_{1 \le i \le n} D_i^T(0))$, the subgraph indexed by x_i is a subgraph of $G_T\}$.

The main difference between a temporal relational cube and a graph temporal cube is the data contained in a *t*-cell. In a relational temporal cube, a *t*-cell is a tuple of values returned by respectively applying aggregate functions $f_j(C, M_j)$ on each measure in $M = \{M_1, M_2, \ldots, M_j, \ldots, M_k\}$, using each c_i of $C = (c_1, c_2, \ldots, c_i, \ldots, c_n)$, $c_i \in D_i^T(0) \cup \{‘*’\}$, $1 \leq i \leq n$, as the filter of D_i^T . However, in a temporal graph cube, a *t*-cell is conceptually a subgraph of the original graph defined by a graph query using K as the slice condition (using K_i to slice the aggregated temporal dimension D_i^t).

An example depiction of a temporal graph cube $GC_T = (T, G_T, (R^T, C))$ is presented in Fig. 12, with T representing the *Time* dimension, and R^T and C representing the dimensions as depicted in Fig. 9 and Fig. 10, respectively.

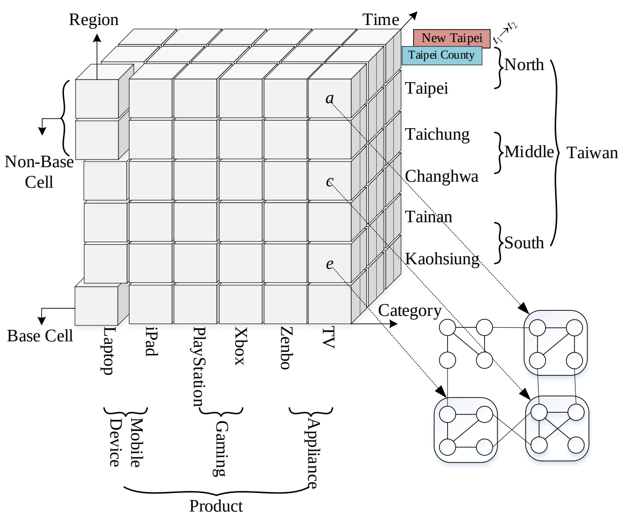

Fig. 12. An example of multi-dimensional temporal graph cube.

Each *t*-cell in Fig. 12 corresponds to a subgraph containing objects and their relationships defined by the intersected dimension members across all engaged aggregated temporal dimensions. For instance, *t*-cells *a*, *c*, and *e* respectively reference three subgraphs concerning the social network netizens residing in {(Taipei County)$_{t_1}$, (New Taipei)$_{t_2}$} (in *North* Taiwan), {*Taichung*} (in *Middle* Taiwan), and {*Tainan*} (in *South* Taiwan), who purchased *TVs* on the same day (e.g., ‘2021/08/08’ in the *Time* dimension), with their *friendship* relationships. If the purchase date precedes t_2, then *t*-cell *a* corresponds to the dimension keyword ‘Taipei County’; otherwise, it corresponds to the dimension keyword ‘New Taipei’. By selecting t = ‘2021/08/08’, the system can generate these subgraphs for users respectively using the tuples (t, {New Taipei}, {TV}), (t, {Taichung}, {TV}) (t, {Tainan}, {TV}) as filters on the temporal graph cube. In contrast, in a traditional temporal relational cube structure, the cells just respectively store three numbers regarding the amounts of TVs bought by customers located in {(New Taipei)$_{t_2}$}, {Taichung} and {Tainan} at time t_2.

V. Visualization and Summarization of Graph Cubes

Following the establishment of a temporal graph cube, all temporal dimensions or attributes associated with vertices and relationships can be leveraged to generate a summarization of all subgraphs defined by t-cells facilitating on-line analytical processing in social networking. For instance, if the vertices of Person (i.e., the participants of the 2020 ICDE Conference) in Fig. 3 are expanded and used to construct two dimensions, one for Degree (e.g., {MS, PhD}), and the other for Gender (i.e., {Female, Male}) as shown in Fig. 13. Then, any subgraph indexed by a t-cell defined on these dimensions can be used to derive the summarization of participants based on their degrees and genders, respectively (as shown in Fig. 14).

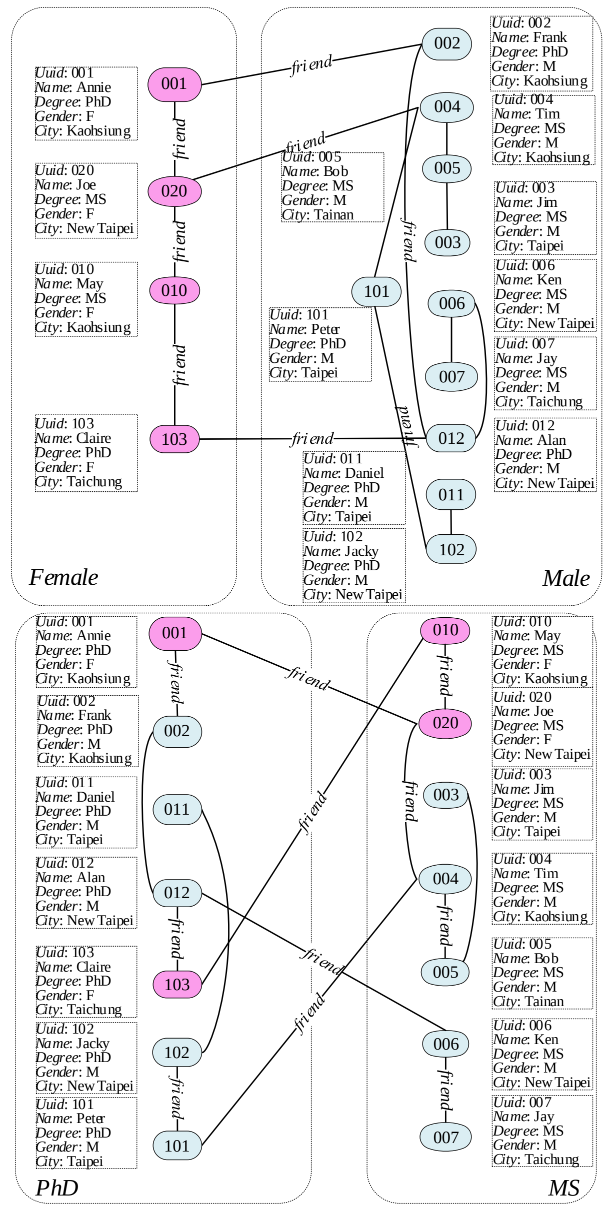

Fig. 13. Two views of the friendships for the dimensions Gender and Degree of Person at time *t*.

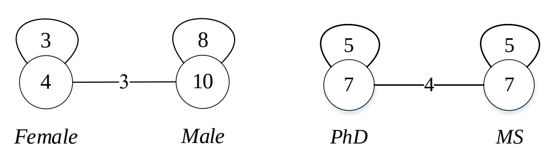

Fig. 14. Two summarizations of the *friendship* of Fig. 13 at time *t*.

Furthermore, it also offers the summarization of composite dimensions. For example, by regarding Gender × Degree = {*Female*, *Male*} × {*MS*, *PhD*} = {(*Female*, *MS*), (*Female*, *PhD*), (*Male*, *MS*), (*Male*, *PhD*)}) as a composite dimension, the summarization result at some time *t* can be derived as Fig. 15 illustrates. The intricate relationships depicted in Fig. 15 can be internally stored within the system, and utilized to execute an operation akin to the traditional DRILL-THROUGH operation in multi-dimensional query language like MDX [23] or MD^2X [26]. This functionality enables users to navigate from the summarized results shown in Fig. 15 to access the detailed information pertaining to each engaged vertex.

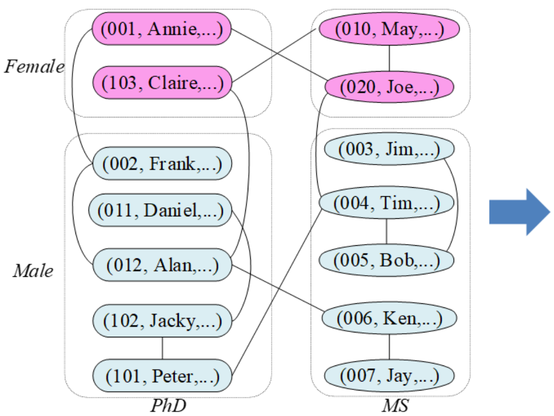

Fig. 15. The summarizations of Gender × Degree at time *t*.

Another intricate perspective concerning the cities of Person with Degree (i.e., {*PhD*, *MS*}) are situated at a certain time *t* is illustrated in Fig. 16. This view aids in computing the summarization for the Degree-*City* relationships at time *t* (as depicted in Fig. 17). Additionally, the perspective regarding the cities where individuals of different genders are located is provided in Fig. 18. This view assists in calculating the summarization for the *Gender-City* relationships at time *t* (as shown in Fig. 19).

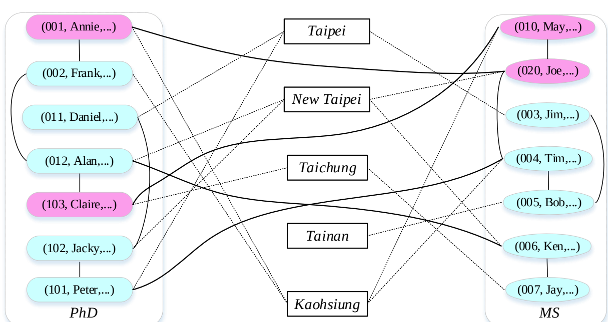

Fig. 16. A view of the Degree-City relationships at time *t*.

The drill-down and roll-up operations outlined in Definitions 3 and 4 can also be seamlessly executed within temporal graph cubes. For instance, Fig. 20 illustrates two subgraphs obtained by rolling up one level (along the dimension R^T) from Fig. 17 and 19, respectively.

In contemporary times, numerous fan pages proliferate across social networks, such as Facebook or Instagram, serving as platforms for gathering stakeholders' feedback, disseminating promotion content, or conducting sentiment analysis on valuable customers alongside their friends or followers. These shared comments or resources can be processed, structured and integrated into graph cube frameworks for social network analytics, leveraging the principles of temporal graph data warehousing [36] to generate insights for short-term analysis or long-term strategizing. Such features offer a wealth of opportunities for users to extract social business intelligence from graph databases substantially. The insights derived can be systematically harnessed for internal knowledge management and disseminated to relevant users with value-added feedback, thus perpetuating a virtuous cycle of information exchange and refinement [4] [24] [27].

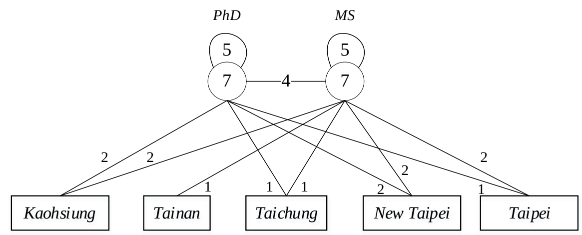

Fig. 17. The Summarization of Degree-City Relationships at time *t*.

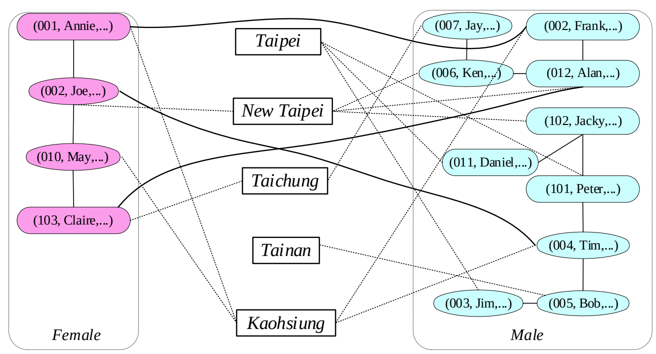

Fig. 18. A view of the Gender-City relationships at time *t*.

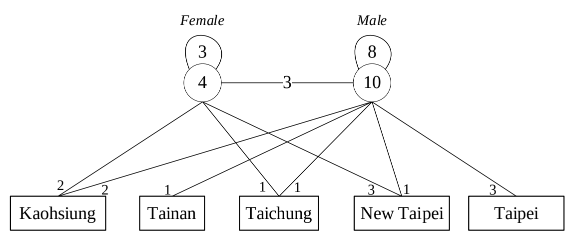

Fig. 19. The summarizations of Gender × City at time *t*.

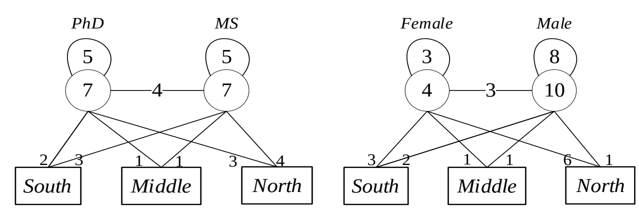

Fig. 20. One level rolling up (along the dimension R^T) for Fig. 17 and Fig. 19 at time *t*.

VI. Summary and Future Directions

We are currently observing an unparalleled expansion of interconnected data, emphasizing the crucial significance of graph processing in our society. We recognize that big graph processing systems integrating temporal features, alongside their associated data warehousing capabilities, are now fundamental components within numerous emerging data management ecosystems across various domains of societal relevance [21]. Our temporal graph model can be used to model many practical applications with abstractions. The incorporation of temporal features in graph data management and warehousing is indispensable for contemporary applications across diverse domains. By providing a dynamic perspective on relationships and patterns, temporal features enhance the analytical capabilities of graph databases, contributing to more informed decision-making processes. While challenges exist, ongoing research and technological advancements are addressing these issues, ensuring that the integration of temporal features continues to be a forefront consideration in the evolution of graph data management systems.

We hopefully expect the following benefits can be obtained through temporal graph warehousing:

1) Improved predictive analytics: Temporal features contribute to more accurate predictive models, allowing for the anticipation of future trends and events. This is particularly valuable in applications where timely decisions are paramount.

2) Enhanced pattern recognition: Temporal graph data facilitates the identification of recurring patterns and anomalies. This is valuable in diverse domains, including cybersecurity, where detecting temporal patterns of malicious activities is critical.

3) Temporal graph warehousing for historical analysis: Temporal graph warehousing enables the retrospective analysis of data, fostering a deeper understanding of historical trends and facilitating informed decision-making based on the evolution of relationships over time.

In Fig. 21, we depict an IoT network consisting of different sensors, where blue vertices are used for detecting water levels in the underpasses of a mega city, and gray vertices are used to monitor hill landslides of some geolocations. Assuming their status can be divided into *normal*, *warning*, and *dangerous*. When their status changes continuously along the timeline, a graph warehousing system can be built by deriving the t-cells of a temporal graph cube for each t moment, such that the number of dangerous spots can be visualized and calculated instantly for administrative decision makings. If the number of dangerous spots runs over a threshold (e.g., there are respectively 4 and 2 dangerous places with landslide and high-water levels detected in Fig. 22), then by drilling through to target the dangerous sensors, the city government can activate the alarm system for traffic control or an emergency procedure for possible evacuation.

Through integration with location-based service facilitated by mobile devices and leveraging a resource multiplexer, diverse multi-dimensional analyses for various types of networking business intelligence can be seamlessly conducted immediately following the integration of the temporal data stream. For instance, to prevent the Covid-19 pandemic, each vertex in Fig. 21 can also be regarded as an instance of type Branch to represent a branch of some chain stores, and customers entering a branch can also be represented by vertices of type Customer. By gathering the cellphone check-in information (arriving at irregular intervals) of all customers in a branch, our framework can help enterprise administrators derive the status of each branch, to grasp the number of customers at different timestamps. If a customer (with the cellphone number) is notified as suspected of being infected, then the temporal graph together with their multi-dimensional summarization result may effectively help administrators make a correct decision to fit the official epidemic prevention policy.

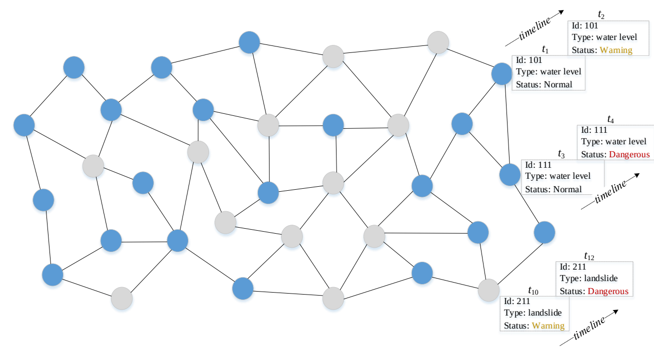

Fig. 21. An example IoT network consisting of different sensors.

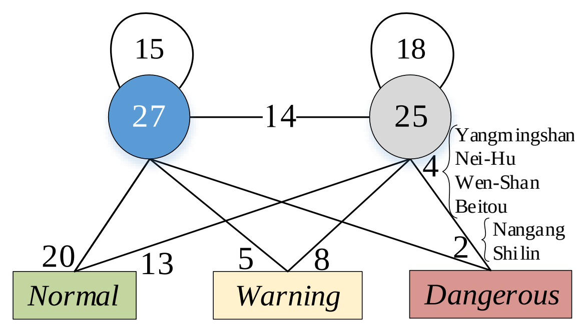

Fig. 22. Aggregations of sensor monitoring.

Based on our model, we also intend to build a temporal graph database based on our roadmap to record the inter-relationships between publicly traded companies in Taiwan and their business partnerships in related countries. Each company will be represented as a vertex with some important attributes announced in the Taiwan Stock Exchange (TWSE) or the corresponding affiliations in their native countries. Attribute values may be changed unpredictably, but should be reported in the official administration websites (e.g., the TWSE website in Taiwan). Therefore, we need to develop a crawler that periodically retrieves the official news of every company and, based on the following identified conditions, adjusts the attribute values, or adds new relationships with other mentioned companies:

1) If the content of the news talks about the adjustment of the company's status, then record the *new attribute values* as another version according to the time point of the announcement.

2) If the news talks about a cooperation with another company, then add a *new relationship edge* between both companies with an edge attribute *start_time* for recording the starting time.

3) If the news talks about a strategic alliance of many companies, then add a *new relationship edge* between every pair of the engaged companies, such that each edge contains an attribute *start_time* for recording the starting time.

4) If the news content talks about the future vision or prediction of an entire industry concerning a lot of companies (which is called *concept stocks* in Taiwan), then add the concept term (like "EV (Electric Vehicle)" or "Metaverse") into an attribute named *concept*, which can be represented as a multi-valued attribute (in JSON format), since a stock can be regarded as involved in many *concept*s.

Graphs serve as ubiquitous abstractions that offer reusable tools for graph processing with applications spanning diverse domains. We believe that the temporal graph framework harbors boundless potential for the development of applications and business opportunities. In our future study, we will step forward to focus on the algebraic frameworks and the query language design for our model to help users create a core of temporal graph processing ecosystems for various applications.

Acknowledgment

This research was partially supported by the National Science and Technology Council, Taiwan, ROC, under contract No. NSTC 112-2410-H-992-022.

References

[1] R. Angles, and C. Gutierrez, "Survey of graph database models," *ACM Computing Survey*, vol. 40, no. 1, 2008, pp. 1-39.

[2] R. Angles, "A comparison of current graph database models," *Proc. IEEE 28th International Conference on Data Engineering Workshops*, 2012, pp. 171-177.

[3] S. Asur, S. Parthasarathy, and D. Ucar, "An event-based framework for characterizing the evolutionary behavior of interaction graphs," *ACM Transactions on Knowledge Discovery from Data*, vol. 3, no. 4, Article 16, 2009, pp. 16:1-16:36.

[4] V. Batagelj, P. Doreian, A. Ferligoj and N. Kejzar, Understanding Large Temporal Networks and Spatial Networks: Exploration, Pattern Searching, Visualization and Network Evolution, John Wiley & Sons, 2014.

[5] P. Bródka and P. Kazienko, "Group evolution discovery in social networks," *Proc. Advances in Social Networks Analysis and Mining* (*ASONAM*), 2011, pp. 247-253.

[6] P. Bródka, S. Saganowski and P. Kazienko, "GED: the method for group evolution discovery in social networks," *Social Network Analysis and Mining*, Vol. 3, 2013, pp.1-14.

[7] C.J. Date, H. Darwen and N. Lorentzos, *Time and Relational Theory: Temporal Databases in the Relational Model and SQL*, The Morgan Kaufmann Series in Data Management Systems, Morgan Kaufmann, 2014.

[8] D. Easley and J. Kleinberg, *Networks, Crowds, and Markets: Reasoning about a Highly Connected World*, Chapter 5, Cambridge University Press, 2010.

[9] S. Faisal, M. Sarwar, K. Shahzad, S. Sarwar, W. Jeffry and M.M. Yousaf, *Temporal and Evolving Data Warehouse Design*, Scientific Programming, Volume 2017, Article ID 7392349.

[10] L.C. Freeman, The development of social network analysis: a study in the sociology of science, ΣP Empirical Press, Vancouver, BC, 2004.

[11] G. Garani, G.K. Adam and D. Ventzas, "Temporal data warehouse logical modelling", *International Journal on Data Mining, Modelling and Management*, vol. 8, no. 2, 2016, pp.144-159.

[12] M. Golfarelli and S. Rizzi, "A survey on temporal data warehousing," *International Journal of Data Warehousing and Mining*, vol. 5, no. 1, 2009, pp. 1-17.

[13] W. Hai, Z. Zeshui, H. Jujita and L. Shousheng, "Towards felicitous decision making: An overview on challenges and trends of Big Data," *Information Sciences*, vol. 367-368, 2016, pp. 747-765.

[14] W. Hamilton, R. Ying and J. Leskovec, "Inductive representation learning on large graphs", *Proc. the 31st Conference on Neural Information Processing Systems* (*NIPS 2017*), 2017, pp. 1024-1034.

[15] L.Y. Ho, J.J. Wu and P. Liu, "Distributed Graph Database for Large-Scale Social Computing," *Proc. the 5th International Conference on Cloud Computing*, 2012.

[16] K. Kulkarni and J.-E. Michels, "Temporal features in SQL: 2011," *ACM SIGMOD Record*, vol. 41, no. 3, 2012, pp. 34-43.

[17] J.J. Miller, "Graph database applications and concepts with Neo4j," *Proc. the southern association for information systems conference, Atlanta, GA, USA*, vol. 2324, no. 36, 2013.

[18] M. Needham and A.E. Hodler, Graph Algorithms: Practical Examples in Apache Spark and Neo4j, O'Reilly Media Inc., 2019.

[19] G. Ozsoyoglu and R. Snodgrass, "Temporal and real-time databases: a survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 7, no. 4, 1995, pp. 513-532.

[20] S. Sahu, A. Mhedhbi, S. Salihoglu, J. Lin and M.T. Özsu, "The ubiquity of large graphs and surprising challenges of graph processing: extended survey," *The VLDB Journal*, vol. 29, no. 2-3, 2019, pp. 595–618.

[21] S. Sakr, A. Bonifati, H. Voigt and A. Isoup, "The future is big graphs: A community view on graph processing systems," *Communications of the ACM*, vol. 64, no. 9, 2021, pp. 62-71.

[22] R.T. Snodgrass, *Developing Time-Oriented Database Applications in SQL*, Morgan Kaufmann Publishers, 2000.

[23] G. Spofford, S. Harinath, C. Webb, D.H. Huang, F. Civardi, *MDX Solutions—With Microsoft SQL Server Analysis Services 2005 and Hyperion Essbase*, 2nd ed., Wiley, 2006.

[24] W. Tan, M.B. Blake, I. Saleh and S. Dustdar, "Social-network-sourced big data analytics," *IEEE Internet Computing*, vol. 17, no. 5, 2013, pp. 62-69.

[25] A.U. Tansel, J. Clifford, S. Gadia, *Temporal Databases: Theory, Design, and Implementation*, Benjamin-Cummings Publishing, 1993.

[26] F.S.C. Tseng, "Design of a multi-dimensional query expression for document warehouses," *Information Sciences*, vol. 174, no. 1-2, 2005, pp. 55-79.

[27] F.S.C. Tseng and A.Y.H. Chou, "Click-and-mortar social networking with business intelligence," *The 5th Annual Conference on Engineering and Information Technology* (*ACEAIT 2018*), Kyoto, March 27-29, 2018, pp. 43-58.

[28] F.S.C. Tseng and A.Y.H. Chou, "Formalizing graph database and graph warehouse for on-line analytical processing in social networks," *Proc. the Future Technology Conference*, vol. 2, San Francisco, CA, USA, Oct. 24-25, 2019, pp. 605-618.

[29] F.S.C. Tseng and A.Y.H. Chou, "Generating social relationships from relational databases for graph database creation and social business intelligence management," *Proc. the Computing Conference*, London, July 10-12, 2018, pp. 220-229. Also published in *Advances in Intelligent Systems and Computing*, Kacprzyk, J. Ed., Springer International Publishing AG, ISSN:2194-5357, 2018, pp. 372-393.

[30] F.S.C. Tseng and A.Y.H. Chou, "Spatiotemporal data warehousing for event Tracking Applications," *Journal of Information Science and Engineering*, vol. 38, no. 6, 2023, pp. 1213-1241.

[31] F.S.C. Tseng and A.Y.H. Chou, "Temporal data processing: problems, realities and perspectives − a practical look," *Proc. the 2024 8th International Conference on Medical and Health Informatics (ICMHI 2024)*, Yokohama, Japan, 2024, pp. 17-19.

[32] A. Vaisman and E. Zimanyi, "What is spatiotemporal data warehousing," *Proc. the 11th International Conference on Data Warehousing and Knowledge Discovery* (*DaWaK 2009*), Linz, Austria, Aug. 31-Sep. 2, Lecture Notes in Computer Science, Springer, Vol. 5691, 2009, pp. 9-23.

[33] C. Vicknair, M. Macias, Z. Zhao, X. Nan, Y. Chen and D. Wilkins, "A comparison of a graph database and a relational database: a data provenance perspective," *Proc. the 48th ACM SE Annual Southeast Regional Conference*, 2010, pp. 42-48.

[34] T. Washio and H. Motoda, "State of the art of graph-based data mining," *ACM SIGKDD Explorations Newsletter*, vol. 5, no. 1, 2003, pp. 59-68.

[35] D. Xu, C. Ruan, E. Korpeoglu, S. Kumar and K. Achan, "Inductive representation learning on temporal graphs," *Proc. The International Conference on Learning Representations* (*ICLR 2020*), 2020.

[36] P. Zhao, X. Li, D. Xin and J. Han, "Graph cube: on warehousing and OLAP multidimensional networks," *Proc. the ACM-SIGMOD Special International Conference on Management of Data*, June 12-16, 2011, pp. 853-864.