Multilayer Networks: For Modeling and Analysis of Big Data

Abhishek Santra, Hafsa Billah and Sharma Chakravarthy
Information Technology Lab, CSE Department, University of Texas at Arlington, Texas, U.S.A.
{*abhishek.santra, uxb7123*}*@mavs.uta.edu, sharmac@cse.uta.edu*

Keywords: Multilayer Networks, Modeling, Analysis, Big Data.

Abstract: In this **position paper**, we make a case for the appropriateness, utility, and effectiveness of graph models for big data analysis focusing on Multilayer Networks (or MLNs) – a specific type of graph. MLNs have been shown to be more appropriate for modeling complex data compared to their traditional counterparts. MLNs have also been shown to be useful for diverse data types, such as videos and information integration. Further, MLNs have been shown to be flexible for computing analysis objectives from diverse application domains using extant and new algorithms. There is research for automating the modeling of MLNs using widely used EER (Enhanced/Extended Entity Relationship) or Unified Modeling Language (UML) approaches.

We start by discussing different graph models and their benefits and limitations. We demonstrate how MLNs can be effectively used to model applications with complex data. We also summarize the work on the use of EER models to generate MLNs in a principled manner. We elaborate on analysis alternatives provided by MLNs and their ability to match analysis needs. We show the use of MLNs for - i) traditional data analysis, ii) video content analysis, iii) complex data analysis, and iv) propose the use of MLNs for information integration or fusion. We show examples drawn from the literature of their modeling and analysis usage. We conclude that graphs, specifically MLNs provide a rich alternative to model and analyze big data. Of course, this certainly does not preclude newer data models that are likely to come along.

1 INTRODUCTION

Big data analytics is predicated upon our ability to model and analyze disparate, complex data sets and associated application objectives. Relational and object-oriented data models have served well for modeling and analyzing transactional data sets that need to be managed over long periods. NoSQL data models filled the gap in modeling and analysis for data sets for which earlier data models were not best suited. New data models including graph models are gaining importance due to the diverse types of social networks and other data types being used for mining, knowledge discovery, querying, and analysis.

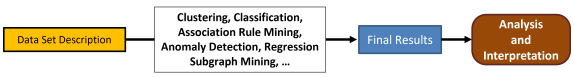


Figure 1: Life Cycle Flow Chart of Mining.

In this paper, we focus on the applicability and versatility of graphs, especially Multilayer Networks (MLNs) for moving towards modeling and analysis of big data. In contrast to the mining approach shown in Figure 1, big data analysis needs to be addressed using a life cycle starting from modeling to drill-down and visualization. Currently, graph models are generated *manually* for a given data set without using any principled approach. For many data sets, both modeling and analysis computations are quite different from the ones addressed in earlier data models. In this paper, instead of generating a schema, application requirements and data are transformed into different types of graphs including MLNs. Moreover, an analysis may require graph computations, such as shortest path, substructure discovery, community, centrality (e.g., hubs), or their combination. Once the chosen data model is generated and the objectives are mapped into appropriate computations, any available package/algorithm can be used. Finally, the analysis results need to be drilled down and visualized in multiple ways for decision-making and for taking action. We present several results from the literature to convince the reader that this workflow is needed. **Figure 2** shows our view of the big data analysis life cycle from *gathered application requirements* to *analysis of objectives* to *result drill-down with visualization*. Only graph and MLN models are shown. This workflow is iterative.

Drill-down of analysis results is critical, especially for diverse data that has both structure and se-

Santra, A., Billah, H. and Chakravarthy, S.
Multilayer Networks: For Modeling and Analysis of Big Data.
DOI: 10.5220/0012997200003838

In *Proceedings of the 16th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2024) - Volume 1: KDIR*, pages 383-390
ISBN: 978-989-758-716-0; ISSN: 2184-3228

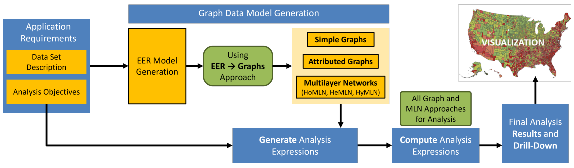


Figure 2: Life Cycle of Big Data Modeling and Analysis using Graphs and MLNs.

mantics. For example, it is not sufficient to know the objects in a community, but additional object details are needed, similarly, for a centrality hub. For graph and MLN models, we also need to know the edges within and across layers, if any. From a computation/efficiency perspective, minimal information needs to be used for analysis whereas the drill-down phase needs to expand upon to the desired extent. Visualization is not new either and there exists a wide variety of tools for visualizing base data, results, and drilled-down information in multiple ways. Several data visualization platforms are available (GeP, 2014, Samant et al., 2021). Due to space constraints, we will not discuss drill-down and visualization in this paper. The contributions of this paper are:

- **Complete Life cycle** for big data analytics in comparison with mining
- **Graph and MLN** models, and analysis alternatives
- Use and applicability of **MLNs for complex data**
- **Graphs and MLNs** applicability for **video data** analysis
- **MLNs Applicability** for information integration/fusion

The rest of the paper is organized as follows. Relevant literature and different graph models are discussed in Sec. 2 and Sec. 3 respectively. The use of MLNs to model and analyze complex data is summarized in Sec. 4. The use of MLN in lieu of graphs is discussed in Sec. 5. Graphs and MLNs applicability to model and analyze video data is summarized in Sec. 6. Finally, the use and applicability of MLNs for information integration/fusion are discussed in Sec. 7. We conclude and outline future work in Sec. 8.

2 RELATED WORK

We discuss here how different phases of the lifecycle have been addressed in the literature.

EER Modeling: Since the 70s, *EER model* (Chen, 1976) has served as a methodology for database design, by representing data and functionality requirements of real-world applications in a precise manner by identifying entities, attributes, and relationships among them. However, with the emergence of data sets with multiple entity types and relationships along with complex analysis requirements, such as shortest paths, important neighborhoods, dominant nodes (or groups of nodes), etc., the relational data model was not adequate for modeling and analysis. Recently, there has been some work in modeling graphs from EER diagrams but is limited to simple and attributed graphs only (Roy-Hubara et al., 2017, Angles, 2018).

Graph and MLN Models: When a graph is used as a data model, the choice of nodes, edges, and their labels becomes important. There are multiple ways of creating them depending on the analysis objectives. Further, creating edges needs similarity/proximity criteria which need to be specified/identified. There needs to be a systematic and configurable approach for converting raw data sets (.csv files, extracted video contents, etc.) to graphs or MLN layers. Only recently, there has been some work (Komar et al., 2020, Santra et al., 2022) on extending the EER approach to generate MLN models.

Graph and MLN Analysis: There is substantial work in the area of simple, attributed graphs and MLNs. For simple graphs, many algorithms have been developed for shortest paths, spanning trees, community detection, centrality measures, and cliques. The breadth and depth-first approaches are also used for many algorithms. For attributed graphs, substructure discovery (Holder et al., 1994, Padmanabhan and Chakravarthy, 2009, Yan and Han, 2002) for interesting exact and inexact or similar substructures, and graph search and querying (Das et al., 2020) have been developed. For MLNs, algorithms have been developed for homogeneous (HoMLN) and heterogeneous (HeMLN) MLNs. Community detection algorithms have been extended to HoMLNs (review: (Kim and Lee, 2015, Magnani et al., 2021)). Further, methods have been developed to determine *centrality measures* to identify highly influential nodes (Solé-Ribalta et al., 2014, Zhan et al., 2015). Recently developed decoupling-based approaches combine partial analysis results from individual layers systematically *in a loss-less manner* to compute communities (Santra et al., 2017) or centrality hubs (Pavel et al., 2023) for layer combinations. Majority of HeMLN work (reviews in (Shi et al., 2017, Sun and Han, 2013)) focuses on developing meta-path based methods for object similarity, object classification, missing link prediction, ranking/co-ranking, and recommendations. Few existing works generate clusters of entities (Melamed, 2014). Most of them concentrate mainly on inter-layer edges and not the networks themselves.

Graph Models for Video Analysis: Several custom approaches have been developed for modeling videos as scene graphs (Ji et al., 2020, Ou et al., 2022) by training deep learning algorithms and can perform fixed types of analysis (Billah et al., 2024). They need to be retrained or a new algorithm is required to perform a new type of analysis. Several frameworks are also available which models extracted video contents as attributed graphs (Yadav et al., 2020, Zhang et al., 2023) and perform analysis on them. However, they do not consider all the extracted video contents for modeling and only support simple analysis such as counting the number of objects. They cannot perform complex analyses (e.g., finding groups) on videos. Graphs and MLNs can be leveraged to model all the extracted video contents and new algorithms/operators need to be developed to perform interesting analysis on videos.

3 GRAPHS FOR BIG DATA ANALYSIS

Graphs capture relationships between entities in application data using nodes and edges. This representation allows us to perform various analyses based on the graph structure and relationships found in the data.

3.1 Graph Types Used as Data Models

A **simple graph** is defined as (V, E) where V is a set of vertices or nodes and E is a set of edges connecting two *distinct* vertices. E is a subset of V × V. The edges are assumed to be unweighted, either directed or undirected, and loops and multiple edges between nodes are not allowed. Typically, vertices have unique numbers, but labels of nodes and edges need not be unique. These graph models are widely used for modeling and analyzing applications.

An **attributed graph** (also called a multigraph) is defined as (V, E, ϕ) where V is a set of vertices or nodes, E is a set of edges connecting two distinct vertices, and ϕ is a function mapping of E to $\{\{x,y\} \mid x,y \in V \text{ and } x \neq y\}$. If the distinctness of nodes is removed, loops will be allowed as well. The main advantage of a multigraph or attributed graph from a modeling viewpoint is that it captures multiple entities and multiple relationships between entities. Multiple labels can be associated with nodes and entities. With the attributed graph model, it is possible to include relevant information from the data description as labels and hence is more expressive as a model than a simple graph model.

An **MLN** is a *network of simple graphs* (or forests). In this model, every layer represents a distinct relationship among entities with respect to a single (or combination of) feature(s). The sets of entities across layers, which may or may not be of the same type, can be related to each other too.

An MLN can be used to separate entities and corresponding relationships from an attributed graph into separate layers where each layer is a simple graph. This provides more clarity in understanding and processing. MLNs are widely used for modeling complex data sets with multiple types of entities and multiple relationships between the same types of entities. They can also capture relationships between different types of entities.

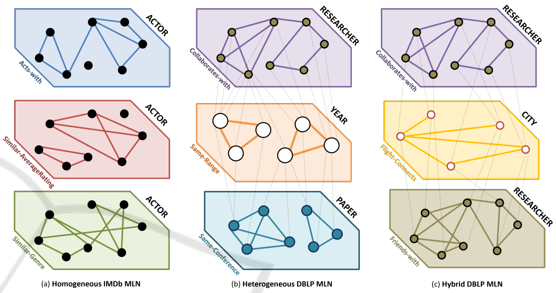


Figure 3: Multilayer Network Types.

Based on the type of relationships and entities, MLNs can be classified into three types. Layers of a **homogeneous MLN (HoMLN)** are used to model different relationships among the **same entity types** like movie actors who are linked based on co-acting (i.e., they act together in a movie) or have similar average rating or have worked in similar genres (Figure 3(a)). Thus, $V_1 = V_2 = \ldots = V_n$ and inter-layer edge sets are empty as no relations across layers are necessary. Relationships among **different types of entities** like researchers (connected by co-authorship), research papers (connected if published in the same conference), and year (related by pre-defined ranges/eras) are modeled through **heterogeneous MLN (HeMLN)** (Figure 3(b)). The inter-layer edges represent the relationship across layers like writes, published-in, and active-in. In addition to being collaborators, researchers may be social media friends. Thus, to model multi-feature data that capture **multiple relationships within and across different types of entity sets**, a combination of homogeneous and heterogeneous MLNs is used, termed **hybrid MLN (HyMLN)**, as shown in Figure 3(c). Here, the first and the third layer have the same node types (researchers) linked to the city nodes they reside in, which are in turn connected based on the flight network (second layer).

The above graph types and MLN variants provide alternatives for matching modeling and analysis needed for application data. Further, MLNs provide clarity in understanding the data set. Additionally, the availability of algorithms for a specific graph model also plays a key role in the choice of the graph model. For instance, there are not many algorithms available for attributed graphs in contrast to simple graphs. There is considerable ongoing research in developing algorithms for the MLNs (Boden et al., 2012, Santra et al., 2017) due to the clarity of the model. Hence, MLNs are preferred for modeling complex data sets.

3.2 EER Modeling Extensions

In contrast to the relational data model, a principled approach to convert application requirements into a chosen graph model (simple, attributed, or MLN) is lacking. However, recently there has been some work in this regard (Komar et al., 2020, Santra et al., 2022) leading to the wider use of MLNs. Broadly, the entities in the EER diagram dictate the formation of layers with the entity instances as layer nodes and the binary self relationship defining the intra-layer edges. The binary non-self relationships define the inter-layer edges. Some relationships are self-explanatory and can be easily mapped into edges like friendships, siblings, direct flights, and so on. However, some relationships are non-explicit like "two actors working in *similar* genre of movies" for which the EER model needs to have a *parameter* attribute for the relationship that defines the similarity metric and threshold. The value of these parameter attributes will be used to generate the edges in the MLN. Currently, we are developing algorithms for converting EER to any type of graph, not just MLN. More research is needed in this area to make analysis easier.

4 MLNs FOR BIG DATA ANALYSIS

Depending on the analysis requirements, the Google Knowledge Base (GKB) data set can be modeled as different types of MLNs. For instance, there exist multiple relationships among the same set of people - whether they are married to each other or have the same birth state or studied in the same university, and so on. This gives rise to a homogeneous GKB MLN with the same set of nodes being connected differently in each layer (Figure 4(a)). Similarly, Figure 4(b) shows an HeMLN where both layers have different sets of entities - person, and company.

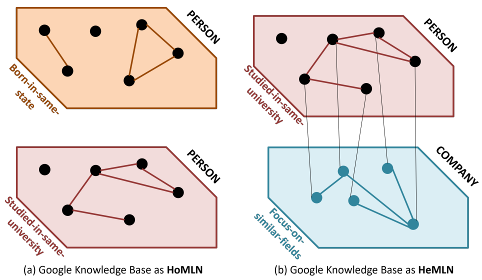


Figure 4: Google Knowledge Base modeled as MLNs.

The person nodes are connected if they studied in the same university, the company nodes are connected if they focus on similar fields, and the person nodes are connected to the company nodes that they founded/established through inter-layer edges. This may also be extended to Hybrid MLNs if two different person layers are connected to a company layer.

4.1 MLN: Multiple Analysis Choices

Figure 5 shows three MLN analysis alternatives. Figure 5(a) shows an MLN conflated into a simple graph by aggregating layers. These aggregation approaches, termed type-independent (Domenico et al., 2014) and projection-based (Berenstein et al., 2016), ignore type information. Hence, they do not support structure and semantics preservation without elaborate mappings as they aggregate or collapse layers into a simple graph in different ways. As observed in the literature, *without additional mappings*, currently-used aggregation approaches are likely to result in some information loss, distortion of properties, or hide the effect of different entity types and/or different intra- or inter-layer relationships (Kivelä et al., 2013, De Domenico et al., 2014). At the other end of the spectrum, Figure 5(c) shows the same MLN layers and result computation by traversing the MLN as is.

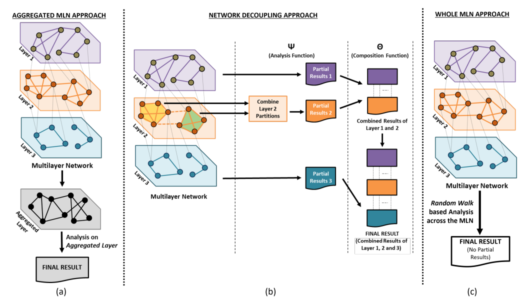


Figure 5: (a) Lossy Vs. (b) Decoupling Vs. (c) Whole MLN approaches (Santra et al., 2022).

Figure 5(b) on the other hand proposes an approach, termed **networking decoupling**, where network property for each layer is computed independently (possibly in parallel) in the analysis (Ψ) phase and compose them using a binary operator Θ. This approach has been shown to be effective and can be done using Boolean operations for HoMLNs and HeMLNs without losing type information. Furthermore, it is more efficient than the approaches shown in Figure 5(a) or (c). Finally, the clarity of modeling using MLNs is retained as well.

5 USE OF MLNs IN LIEU OF GRAPHS

Based on daily life interactions (education, social media platforms, restaurant check-ins, healthcare check-up appointments, etc.) different facts are available on the web. In terms of knowledge base, "different facts" about a person are captured in the GKB. Freebase captures such information for famous personalities: birth place and residence, education institutions attended, birth and death date (if available), companies worked in/founded, family-based relationships and so on (Bollacker et al., 2008). Here, people, universities, companies, and states are related to each other based on explicitly available interactions or relationships. Some interesting analysis objectives can be:

- *(GKB-O1)* Find frequently occurring patterns among states, based on university locations and place of company headquarters for the entrepreneurs.
- *(GKB-O2)* Find groups of people who were born in the same state and have studied in the same university.
- *(GKB-O3)* For each group of founders who have studied in the same university, find out the most popular focus field among the group of similar companies that they have founded.

Although objective *(GKB-O1)* can be computed using traditional graph models, MLNs are needed for objective *(GKB-O2)* and others similar to that. The HoMLN shown in Figure 4(a) is required to address *(GKB-O2)*. In this case, we need to "Find **groups** of people who were *born in the same state* **and** have *studied in the same university*". Here "grouping" keyword means that we need to compute communities among the people nodes, followed by AND composition (due to the "and" keyword). For AND composition, here the CE-AND composition algorithm is used that intersects the community edges, then perform a connected component analysis to obtain the group of nodes that are tightly connected in both the layers (Santra et al., 2022, Santra et al., 2017). Thus, the analysis expression based on the decoupling approach can be expressed as:

```
Expression: Ψ(PERSON-Born-in-same-state) Θ
Ψ(PERSON-Studied-in-same-university);
where Ψ = Community; Θ = CE-AND (composition)
```

Similarly, for *(GKB-O3)*, the HeMLN shown in Figure 4(b) is used. Here, "For each **group** of founders who have *studied in the same university*, we need to find out the **most popular** focus *field* among the **group** of *similar companies* that they have founded." Thus, communities need to be detected in both person and company layers, which become meta-nodes in the bipartite graph. The number of inter-layer edges between the constituent nodes of each pair of meta nodes will define the edge weight. Finally, maximal weighted matching (MWM) will give us the required optimal pairing of person and company communities (Santra et al., 2022). The analysis expression is as follows:

```
Expression: Ψ(PERSON-Studied-in-same-univer- sity)
Θ Ψ(COMPANY-Focus-on-similar-fields);
where Ψ = Community; Θ = MWM (bipartite maximum
weighted matching)
```

6 GRAPHS/MLNs FOR VIDEO ANALYSIS

Our goal, as part of big data analysis, is to handle different data types (4 Vs of big data) in the same way we handle structured and tabular data. If videos (or extracted contents) can be modeled using graphs/MLNs, the same life cycle approach can be applied for video analysis, enabling the modeling and analysis of video data alongside other data types. As discussed in Sec. 2, the existing custom approaches for video analysis require new software/algorithm/re-training to perform a new analysis. Hence, this approach does not lend itself to the holistic analysis required for big data. In contrast, if big data analysis were to include video analysis in mainstream data processing, a different approach would be needed.

Some works in the literature used graphs for video analysis as explained in Sec. 2. Recently, (Billah et al., 2024) proposed a novel approach for video analysis that has the potential to advance big data analysis to include videos. This approach is novel as video contents are extracted once (using existing Video Content Extraction (VCE) algorithms), modeled, and then analyzed to identify a variety of situations from them. This approach has several advantages: i) video contents are **extracted only once**, ii) it is possible to model these extracted contents completely, iii) several analysis expressions can be formulated and computed on them, iv) both "ad hoc" and "what if" analysis can be supported, and v) most importantly, this can be extended for **real-time analysis**.

A workflow of open-source VCE algorithms can be used for extracting object bounding boxes and class labels (with a confidence score) using object detection (YOLO (Wang et al., 2024)) algorithm, unique identifier (object_id) for each object and feature vectors using object tracking (Bot-sort (Aharon et al., 2022)) algorithm, and pose coordinates using pose estimation (HRNet (Wang et al., 2020)) algorithm.

Modeling of Extracted Video Contents: The different types of extracted video contents can be modeled in multiple ways. Two promising models that are being explored in the literature are the extended relational model (Billah and Chakravarthy, 2024) and the graph model (Billah et al., 2024). If it is modeled using an extended relational model, Continuous Query Language (CQL) (an extension of the widely-used Structured Query Language (SQL)) can be used. If the extracted contents are modeled as graphs, different graph analysis techniques can be used. The rationale for using multiple models is that some analysis may be easier in one model as compared to the other. For example, clustering of objects is easier using the graph model than the extended relational model. We will focus on the graph model as this paper is about the utility of graphs and MLNs for big data analysis.

To represent extracted video contents as graphs, nodes and edges need to be identified and other related information (e.g., the feature vectors, bounding boxes, etc.) needs to be associated properly for computation. Many analyses involve objects. Hence, objects are represented as nodes and *Object_id* as node id in the literature. There are multiple choices to create edges (e.g., the distance between objects, and their spatial relationship in a frame, etc.). Figure 6 shows a graph representation of a sample video frame with nodes with two labels: frame id (f_{id}) and object class label (O_l) and edges (based on the objects bounding box centroid distance).

A spectrum of alternatives exists for the graph representation, each with different advantages and disadvantages. It is possible to model the *entire video as one graph* (model M_1) using object_id for nodes (with a large amount of information with each node). It is also possible to create *a graph for each frame* (model M_F) (shown in Figure 6), where the number of graphs will be equal to the number of non-empty frames F in the video. Options in-between are also possible where a forest of g ($1 \leq g \leq F$) graphs (model M_g) can be generated by aggregating the consecutive frames into a graph based on some constraints, with varying numbers of graphs for different videos. The in-between alternatives allow us to compress node labels and edges in different ways reducing the storage required and can also reduce computational complexity as the graphs are generated in some logical manner.

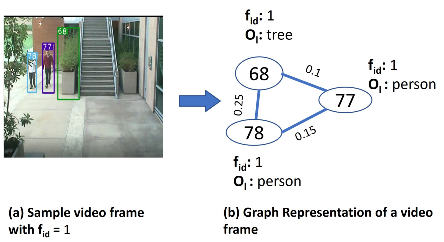


Figure 6: Graph representation of a sample video frame.

Video Content Analysis Using the Graph Models: Below, we indicate video analysis examples using graph models from the literature.

1. **Identifying Groups (Billah et al., 2024):** In assisted living environment videos, it is useful to identify isolated individuals (not participating in group discussions, etc.) This analysis has been reported in (Billah et al., 2024) to cluster individuals in video frames by leveraging K-Means clustering on model M_F where nodes are objects and edges are the object bounding box centroid distances.
2. **Identifying if a Parking Slot is Occupied (Yadav et al., 2020):** In surveillance videos, it is often important to know which parking spaces are occupied. This analysis has been reported in (Yadav et al., 2020) using model M_F, where nodes are objects and edges are spatial bounding box relationships (e.g., overlap, inside, etc.) between objects in a frame. Their proposed algorithm identifies a parking lot as occupied if the parking lot and a car's bounding box overlap over a threshold.

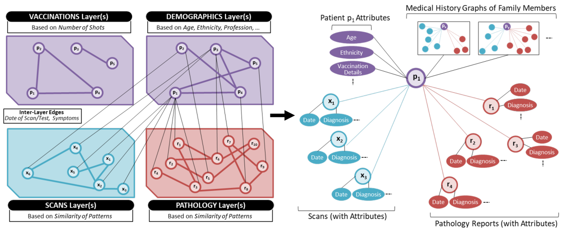
In summary, extracted video contents are shown to be modeled and analyzed using alternative graph models and analysis algorithms. MLNs come in handy to model multiple graphs (or videos) as different layers and perform combined analysis. HoMLNs can be used by connecting object_ids from different graphs generated from the same video or by connecting object_ids from different videos if their feature vectors match. Once modeled appropriately, interesting analysis (e.g., groups of objects entering and exiting a premise after n minutes of each other) can be performed using graphs/MLNs.

7 MLNs FOR INFORMATION FUSION

Analysis of a *single modality/data type* has been the major focus until now, be it structured (e.g., stream data processing (Barbieri et al., 2010)) or unstructured data (e.g., image and video analysis (Zhang et al., 2023, Yadav et al., 2020), text and natural language processing (Otter et al., 2020)). Yet, when all or a subset of these data types must be analyzed holistically, several challenges emerge. These problems have been categorized under various headings, such as data fusion, multi-modal data analysis, and others which are limited in scope and context (Atrey et al., 2010). The challenges originate due to the lack of approaches that can effectively perform information fusion both at the modeling and analysis stages. Therefore, the holistic approach needs to accommodate modeling, and analysis techniques for objectives for performing knowledge discovery. In our view, MLNs with their modeling and analysis advantages provide a path to explore information fusion. Many applications, such as cybersecurity, healthcare, and surveillance can benefit from this. We illustrate this with an example.

Sample Application – Healthcare: Patient data is collected in diverse formats by different specialists over time. This data constitutes the patient's medical records including demographics, hospital/doctor visits, vital signs, medications, progress notes, allergies, radiology images, and laboratory results, and can be further enriched by exercise data, etc. This data is both spatial and temporal. When all this data is accumulated, holistic knowledge discovery over an individual and the population is possible. This application with big data characteristics can be used for personalized care using querying, searching, and mining. MLNs can be used for effectively modeling this data and for flexible analysis. Layers that can be identified are: i) **Demographics Layer(s)**: Patient nodes are connected by edges based on demographics (age, ethnicity, profession, education level, etc.), ii) **Image/Video Layer(s):** Patient nodes are connected based on the similarity of patterns present in them (X-rays, MRIs, EKG, and CT Scans), iii) **Pathology Layer(s):** Patient nodes are connected based on the similarity of indicators (e.g., high sugar, high/low BP, etc.), iv) **Vaccination Layer(s):** Person nodes are connected based on the number of doses and type of shots. These layers can be generated for county/city/state as needed.

Figure 7(a) illustrates 4 possible layers of the hybrid healthcare MLN, with the inter-layer edges. For example, the demographics layer can be linked with scan/pathology layers based on whom the report belongs to with the test report date and symptoms as the label information. From this MLN, it is also possible to extract graphs for an individual or a select group for different types of analy-



(a) Hybrid Healthcare Multilayer Network (b) Extracted Medical History Graph of Patient p_1

Figure 7: Healthcare MLN.

sis (shown in Figure 7(b) for patient p_1 and his/her family). This model with the extracted graph(s) allows us to query, search, and analyze to discover knowledge using all or a subset of layers in various ways. Few examples are - using collective information of an individual and family, a physician can draw holistic inferences which may not be possible without a model that represents multi-source, multi-type data (**personalized/customized holistic diagnosis/inference**), find group(S) of people for a specific demographics who had lung problems and other co-morbidity (e.g. diabetes) and contracted Covid (**aggregate analysis using homogeneous and heterogeneous community detection on multiple layers**), people who did not have any history of lung issues but contracted Covid (**mining on a subset of layers using Boolean NOT operation**).

8 CONCLUSIONS

In this position paper, we argue for MLNs as **a** viable alternative for big data analytics. We have discussed the versatility of MLN models and their ability to model diverse data, the recent work on MLN model generation using the EER approach, and efficient MLN algorithm development for analysis. Based on MLN work in the literature, we have argued for their use for modeling and analyzing complex data sets including images, videos, and other data types (e.g., natural language). There is an ongoing effort to apply MLNs for information fusion/integration as well.

ACKNOWLEDGMENTS

This work was supported by NSF awards #1955798 and #1916084.

REFERENCES

(2014). Gephi - The Open Graph Viz Platform . http://gephi.org/.

Aharon, N., Orfaig, R., and Bobrovsky, B. (2022). Bot-sort: Robust associations multi-pedestrian tracking. *CoRR*, abs/2206.14651.

Angles, R. (2018). The property graph database model. In *AMW*.

Atrey, P. K., Hossain, M. A., El Saddik, A., and Kankanhalli, M. S. (2010). Multimodal fusion for multimedia analysis: a survey. *Multimedia Systems*, 16(6):345–379.

Barbieri, D. F., Braga, D., Ceri, S., VALLE, E. D., and Grossniklaus, M. (2010). C-sparql: a continuous query language for rdf data streams. *International Journal of Semantic Computing*, 4(01):3–25.

Berenstein, A., Magarinos, M. P., Chernomoretz, A., and Aguero, F. (2016). A multilayer network approach for guiding drug repositioning in neglected diseases. *PLOS*.

Billah, H. and Chakravarthy, S. (2024). Video situation monitoring using continuous queries. In *DEXA,2024*, volume 14911 of *LNCS*, pages 125–141. Springer.

Billah, H., Santra, A., and Chakravarthy, S. (2024). Leveraging video situation monitoring in assisted living environment. In *PETRA, 2024*, pages 307–315. ACM.

Boden, B., Günnemann, S., Hoffmann, H., and Seidl, T. (2012). Mining coherent subgraphs in multi-layer graphs with edge labels. KDD '12, pages 1258–1266.

Bollacker, K., Evans, C., Paritosh, P., Sturge, T., and Taylor, J. (2008). Freebase: a collaboratively created graph database for structuring human knowledge. SIGMOD '08, pages 1247–1250, New York, NY, USA. ACM.

Chen, P. P.-S. (1976). The entity-relationship model—toward a unified view of data. *ACM transactions on database systems (TODS)*, 1(1):9–36.

Das, S., Santra, A., Bodra, J., and Chakravarthy, S. (2020). Query processing on large graphs: Approaches to scalability and response time trade offs. *Data Knowl. Eng.*, 126:101736.

De Domenico, M., Solé-Ribalta, A., Gómez, S., and Arenas, A. (2014). Navigability of interconnected networks under random failures. *Proc. of Ntl. Acad. of Sciences*.

Domenico, M. D., Nicosia, V., Arenas, A., and Latora, V. (2014). Layer aggregation and reducibility of multilayer interconnected networks. *CoRR*, abs/1405.0425.

Holder, L. B., Cook, D. J., and Djoko, S. (1994). Substucture Discovery in the SUBDUE System. In *Knowledge Discovery and Data Mining*, pages 169–180.

Ji, J., Krishna, R., Fei-Fei, L., and Niebles, J. C. (2020). Action genome: Actions as compositions of spatio-temporal scene graphs. In *CVPR*, pages 10236–10247.

Kim, J. and Lee, J. (2015). Community detection in multi-layer graphs: A survey. *SIGMOD Record*, 44(3):37–48.

Kivelä, M., Arenas, A., Barthelemy, M., Gleeson, J. P., Moreno, Y., and Porter, M. A. (2013). Multilayer networks. *CoRR*, abs/1309.7233.

Komar, K. S., Santra, A., Bhowmick, S., and Chakravarthy, S. (2020). Eer→mln: EER approach for modeling, mapping, and analyzing complex data using multi-layer networks (mlns). In *ER 2020*, pages 555–572.

Magnani, M., Hanteer, O., Interdonato, R., Rossi, L., and Tagarelli, A. (2021). Community detection in multiplex networks. *ACM CS.*, 54(3):48:1–48:35.

Melamed, D. (2014). Community structures in bipartite networks: A dual-projection approach. *PloS one*, 9(5):e97823.

Otter, D. W., Medina, J. R., and Kalita, J. K. (2020). A survey of the usages of deep learning for natural language processing. *TNNLS*, 32(2):604–624.

Ou, Y., Mi, L., and Chen, Z. (2022). Object-Relation Reasoning Graph for Action Recognition. In *CVPR*, pages 20133–20142.

Padmanabhan, S. and Chakravarthy, S. (2009). HDB-Subdue: A Scalable Approach to Graph Mining. In *DaWaK*, pages 325–338.

Pavel, H. R., Roy, A., Santra, A., and Chakravarthy, S. (2023). Closeness centrality detection in homogeneous multilayer networks. In *IC3K 2023, KDIR*.

Roy-Hubara, N., Rokach, L., Shapira, B., and Shoval, P. (2017). Modeling graph database schema. *IT Professional*, 19(6):34–43.

Samant, K., Memeti, E., Santra, A., Karim, E., and Chakravarthy, S. (2021). Cowiz: Interactive covid-19 visualization based on multilayer network analysis. In *ICDE 2021*, pages 2665–2668. IEEE.

Santra, A., Bhowmick, S., and Chakravarthy, S. (2017). Efficient community re-creation in multilayer networks using boolean operations. In *ICCS 2017*, pages 58–67.

Santra, A., Komar, K., Bhowmick, S., and Chakravarthy, S. (2022). From base data to knowledge discovery–a life cycle approach–using multilayer networks. *DKE*, 141:102058.

Shi, C., Li, Y., Zhang, J., Sun, Y., and Philip, S. Y. (2017). A survey of heterogeneous information network analysis. *IEEE Trans. Knowl. Data Eng.*, 29(1):17–37.

Solé-Ribalta, A., De Domenico, M., Gómez, S., and Arenas, A. (2014). Centrality rankings in multiplex networks. In *Procds. of 2014 ACM conf. on Web science*, pages 149–155. ACM.

Sun, Y. and Han, J. (2013). Mining heterogeneous information networks: a structural analysis approach. *ACM SIGKDD Explorations Newsletter*, 14(2):20–28.

Wang, A., Chen, H., Liu, L., Chen, K., Lin, Z., Han, J., and Ding, G. (2024). Yolov10: Real-time end-to-end object detection. *CoRR*, abs/2405.14458.

Wang, J., Sun, K., Cheng, T., Jiang, B., Deng, C., Zhao, Y., Liu, D., Mu, Y., Tan, M., Wang, X., et al. (2020). Deep high-resolution representation learning for visual recognition. *PAMI*, 43(10):3349–3364.

Yadav, P., Salwala, D., Das, D. P., and Curry, E. (2020). Knowledge Graph Driven Approach to Represent Video Streams for Spatiotemporal Event Pattern Matching in Complex Event Processing. *IJSC*, 14(03):423–455.

Yan, X. and Han, J. (2002). gSpan: Graph-Based Substructure Pattern Mining. In *IEEE International Conference on Data Mining*, pages 721–724.

Zhan, Q., Zhang, J., Wang, S., Philip, S. Y., and Xie, J. (2015). Influence maximization across partially aligned heterogenous social networks. In *PAKDD (1)*, pages 58–69.

Zhang, E., Daum, M., He, D., Haynes, B., Krishna, R., and Balazinska, M. (2023). Equi-vocal: Synthesizing queries for compositional video events from limited user interactions. *VLDB*, 16(11):2714–2727.