`MLN-geeWhiz`: A Dashboard for Supporting Complete Life-Cycle of Complex Data Analysis using Multilayer Networks

Amey Shinde
ITLab & CSE Dept., UT Arlington
amey.shinde@mavs.uta.edu

Viraj Sabhaya
ITLab & CSE Dept., UT Arlington
virajvipinbhai.sabhaya@mavs.uta.edu

Kevin Farokhrouz
ITLab & CSE Dept., UT Arlington
kaf2886@mavs.uta.edu

Fariba Irany
University of North Texas
FaribaAfrinIrany@my.unt.edu

Ali Khan
University of North Texas
AliKhan@my.unt.edu

Sanjukta Bhowmick
University of North Texas
sanjukta.bhowmick@unt.edu

Abhishek Santra
ITLab & CSE Dept., UT Arlington
abhishek.santra@uta.edu

Sharma Chakravarthy
ITLab & CSE Dept., UT Arlington
sharmac@cse.uta.edu

ABSTRACT

Over the last few decades, simple graphs have been extensively used for studying complex systems of interacting entities from diverse disciplines, such as social networks, transportation, epidemiology, etc. However, when studying data with multiple types of entities, relationships, and features, simple (or even attributed) graphs are not always sufficient. For example, to study accident patterns to take mitigating actions, one needs to explore accident patterns based on factors like weather (rain, sunny, sleet, etc.), light, and road surface conditions in different geographical regions. As another example, to find individuals who are influential across multiple social media, a single graph approach is not well-suited. Indeed, to model such multiple relationships, multiple related graphs are useful. This can be done using multilayer networks (MLNs).

Any complex data analysis can immensely benefit from interactive graphic tools rather than working with raw data in command prompt mode. This is especially true as data and models become increasingly complex. To interpret and understand the results of analysis, drill-down, and visualization become critical. The MLN-Dashboard (called `MLN-geeWhiz`) presented in this demo paper aims to facilitate all aspects of MLN layer generation, analysis, and visualization through an intuitive, interactive web-based dashboard. In this paper, we discuss the dashboard, its architecture, the functionality currently supported, and some use cases.

PVLDB Reference Format:
Amey Shinde, Viraj Sabhaya, Kevin Farokhrouz, Fariba Irany, Ali Khan, Sanjukta Bhowmick, Abhishek Santra, and Sharma Chakravarthy. `MLN-geeWhiz`: Multilayer Network Dashboard. PVLDB, 18(12): 5323 - 5326, 2025.
doi:10.14778/3750601.3750662

Proceedings of the VLDB Endowment, Vol. 18, No. 12 ISSN 2150-8097.
doi:10.14778/3750601.3750662

1 INTRODUCTION

MLNs find use in diverse disciplines that require modeling complex interactions, including drug design [7], understanding collaboration patterns [5], understanding the interaction between spoken and written language [10], comparing the evolution of species [23], and identifying illegal activities via social interactions [4].

MLN modeling and analysis is a new frontier for big data analytics. Analysis algorithms, data structures to represent MLNs, and even the definitions of properties, such as community, centrality, substructure, and k-core are still evolving. Thus, a platform where *researchers* can access the latest algorithms and apply them to *their data sets* is pivotal for advancing the field. Although the analysis of multilayer networks is a fast-growing area, there is yet *no tool for the generation, analysis, drill-down, and visualization of MLNs*, let alone sharing and community interaction.

Existing dashboards (Py3Plex [21], MultiNetx [13], Muxviz [6], and Pymnet [14]) are limited in their ability to analyze MLNs and mainly focus on visualizing the results as graphs. In contrast, the focus of this dashboard is to facilitate each user in performing the analysis appropriate for their application data through a flexible configuration file strategy used for specifications. Layer generation can be done using multiple metrics and is easy to specify. Similarly, MLN analysis, whether simple or complex, can be specified by composing known operators (e.g., community, substructure, centrality, etc.) and precedence for evaluation. The dashboard transparently handles the input and output formats. Diverse visualizations and their side-by-side comparisons are supported. The ability to create files and directories for proper organization and management of data and results is provided.

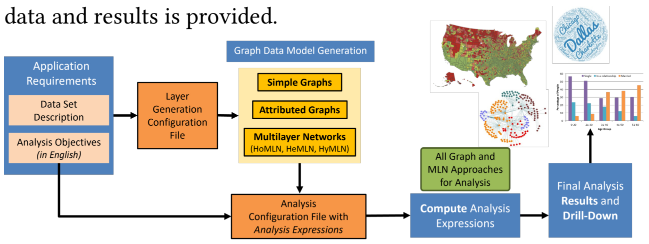
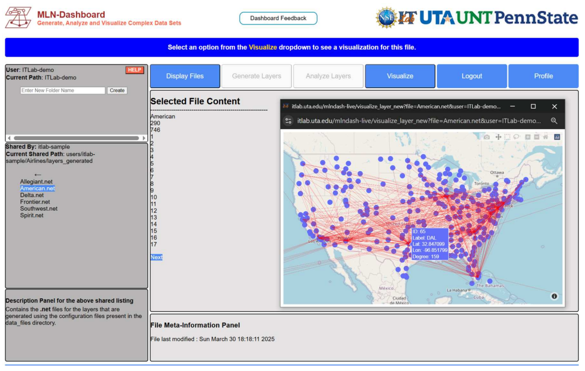

Application Requirements
Data Set Description
Analysis Objectives (in English)
Layer Generation Configuration File
Graph Data Model Generation
Simple Graphs
Attributed Graphs
Multilayer Networks (HoMLN, HeMLN, HyMLN)
Analysis Configuration File with Analysis Expressions
All Graph and MLN Approaches for Analysis
Compute Analysis Expressions
Final Analysis Results and Drill-Down
Chicago
Dallas
Charlotte
Age Group
Percentage of People
Single
In a relationship
Married

Figure 1: MLN Analysis Lifecycle Supported by Dashboard

To support the entire lifecycle (shown in figure 1) from the generation of MLNs using complex raw data to the final drill-down and visualization of results, we have developed a multi-user, web-based dashboard whose interface is shown in figure 2. The initial version has been released for public usage (**Dashboard URL**: https://itlab.uta.edu/mln-geewhiz/, **YouTube video**: https://youtu.be/uA1OD2PqHlY) and supports the following functionalities:

(1) *User Registration and Login*: Multiple users can register, securely log in, and concurrently use the dashboard. Each user has a private workspace and their work persists across sessions. Updating the password is also supported.
(2) *Data Sets*: Several sample data files for applications from diverse domains are available in the shared space (including configuration files for layer generation and analysis), so the user can exercise all the supported functionalities.
(3) *Graph and MLN Layer Generation*: Configuration files (with .gen extension) are used to generate graphs and MLN layers. Many similarity metrics (e.g., Haversine, Euclidean) are supported for relationships (i.e., edges of the graph).
(4) *Graph and MLN Analysis*: Configuration files (with .ana extension) are used to analyze generated layers. Several MLN (both homogeneous and heterogeneous) analysis algorithms have been developed and are already integrated.
(5) *Visualization*: Interactive visualization alternatives, such as networks, word cloud, bar charts, and geographical maps (see Figure 1) are currently supported.



Select an option from the Visualize dropdown to see a visualization for this file.

User: ITLab-demo
Current Path: ITLab-demo
Enter New Folder Name

Shared By: itlab-sample
Current Shared Path: users/itlab-sample/Airlines/layers_generated

←
Allegiant.net
American.net
Delta.net
Frontier.net
Southwest.net
Spirit.net

Description Panel for the above shared listing
Contains the **.net** files for the layers that are generated using the configuration files present in the data_files directory.

Selected File Content

```
American
290
746
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
```

itlab.uta.edu/mlndash-live/visualize_layer_new?file=American.net&user=ITLab-demo...

ID: 65
Label: DAL
Lat: 32.847099
Lon: -96.851799
Degree: 159

File Meta-Information Panel
File last modified : Sun March 30 18:18:11 2025

Figure 2: MLN Dashboard Interface

In addition, any text file can be displayed in a scrollable manner. All user-created directories and files are always shown and re-navigable. New directories can be created. Functional buttons become active based on user interaction.

2 MULTILAYER NETWORKS: AN OVERVIEW

Multilayer networks (or MLNs) have been proposed in the literature as an important alternative for *flexible, expressive, and structure-preserving* modeling as well as for efficient analysis of complex data sets based on different combinations of features [8, 9, 19, 20]. MLN is a *network of networks.* In an MLN, every layer represents a distinct relationship among entities with respect to a feature (single or combined). The sets of entities across layers, which may or may not be of the same type, can be related to each other as well.

Formally, a **multilayer network**, $MLN(G, X)$, is defined by two sets of graphs: i) The set $G = \{G_1, G_2, \ldots, G_N\}$ contains graphs of N individual layers, where $G_i(V_i, E_i)$ is defined by a set of vertices, V_i and a set of edges, E_i. An edge $e(v, u) \in E_i$, connects vertices v and u, where $v, u \in V_i$ and ii) A set $X = \{X_{1,2}, X_{1,3}, \ldots, X_{N-1,N}\}$ of bipartite graphs. Each bipartite graph $X_{i,j}(V_i, V_j, L_{i,j})$ is defined by two sets of vertices V_i and V_j, and a set of edges (also called links or inter-layer edges) $L_{i,j}$, such that for every link $l(a, b) \in L_{i,j}$, $a \in V_i$ and $b \in V_j$, where V_i (V_j) is the vertex set of graph G_i (G_j.)

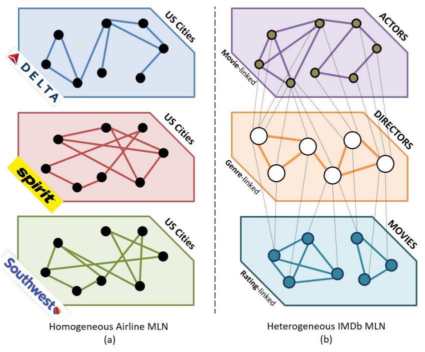


Figure 3: Homogeneous and Heterogeneous MLNs

Based on relationship and entity types, MLNs are classified into different types. **Homogeneous MLN (or HoMLN)** is used to model the diverse relationships that exist among the **same entity types**. For example, US cities are linked if a direct flight exists between them operated by a *specific airline* (Figure 3 (a)). Relationships among **different entity types** like actors (connected based on co-acting), directors (connected if they direct movies of similar genres), and movies (related by pre-defined average rating ranges) are modeled through **heterogeneous MLN (or HeMLN)** (Figure 3 (b)). The inter-layer edges represent the relationship across layers, such as directs-movie, directs-actor, and acts-in-movie (not illustrated). **Hybrid MLN (or HyMLN)** is a combination of both types.

2.1 Decoupling-Based MLN Analysis

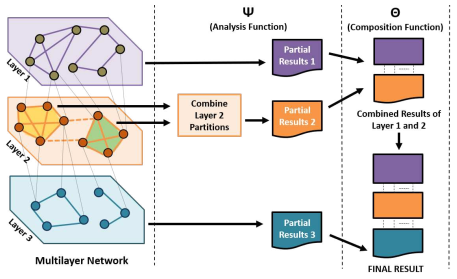


Figure 4: Decoupling-based Approach

We have proposed a novel decoupling approach for efficient detection of various graph metrics (e.g., community, centrality, etc.) This uses the equivalent of "divide and conquer" for MLNs ([16, 17, 22]). Decoupling-based analysis uses individual layers as partitions. The partial (layer-wise or intermediate) analysis results are combined using a composition function Θ as shown in Figure 4 to produce final results. Substantial work has been done to identify the composition function (called Θ) that is appropriate for efficient community, centrality, and substructure detection (called Ψ) for MLNs.

3 DASHBOARD COMPONENT DETAILS

The dashboard has been developed with extensibility in mind. Here, we briefly discuss the functionality and architecture of each module.

3.1 MLN Graph and Layer Generation

Typically, raw application data is in the form of a text file which needs to be used for MLN modeling and analysis. Modeling is done using a config file (with .gen extension). Different MLN layers can be generated for the same application data by specifying node attributes and how an edge is created. Many similarity metrics for edge generation (equality, Euclidean, Haversine, Jaccard, cosine, and many more) are supported for different attribute types (nominal, numeric, lat/long, date, time, set, etc.). Multiple layers (both homogeneous and heterogeneous) and MLNs can be generated using one or more .gen config files. Layers are generated (with .net extension) and stored in user-specified directories using given names. Relevant information is entered into a hash table for efficient lookup during analysis. For drill down, a mapping of labels associated with nodes and edges is also generated (during layer generation) for use by the visualization module.

3.2 Analysis of Graphs and MLNs

Once the layers are generated, they are available for analysis. A configuration file (with .ana extension) is used to specify infix analysis expressions (e.g., community on single or multiple layers). Specified infix expressions with precedence are converted into post-fix format for evaluation, preserving the precedence. For HeMLNs inter-layer graph generation is also done as part of layer generation and is used for analysis. Community detection results in two types of files: nodes in the community (as .vcom files) and edge-based community (as .ecom files). They are also stored in specified directories.

3.3 Alternative Visualizations

Text files can be chosen for display on the window with scrolling. Any generated graph (or layer) or computed analysis results can be visualized using available alternatives - (i) **Bokeh**: dynamic network visualization, (ii) **Plotly Network**: handles graphs with 100K+ nodes, (iii) **Plotly Map**: for geographical data, (iv) **Pyvis**: interactive graph visualization with zooming and toggling nodes, (v) **Word Cloud**: highlights communities based on their size, and (vi) **Bar Graph:** indicates the number and size of generated communities.

3.4 Architecture and Implementation Details

Figure 5 shows the control flow of the dashboard. The modular architecture allows easy extensibility of existing and the addition of new modules, as well as facilitates parallel implementation.

The left panel of the dashboard displays the directories and files (see Figure 2.) *Tabs/buttons of the dashboard are enabled for operations based on the file type chosen.* The user is proactively prevented from making mistakes. Meaningful error messages are displayed. When a chosen file can be visualized, a drop-down menu indicates the compatible visualization options available. Multiple visualizations of the same file are generated as pop-up windows that can be arranged side by side for comparison and understanding.

SQLite3 database is used for the storage of user profiles, encrypted passwords, and other user information collected during registration. Proper unique user and session identifiers are generated to guarantee isolated concurrent usage to ensure a smooth user experience. Information needed across the modules and sessions is stored in a hash table (Python dictionary) and persisted. This is used to: (i) avoid repeating the same computation based on the file timestamp, (ii) fast lookup of files used for analysis, and (iii) not regenerate visualization if it has already been done (as it takes considerable response time based on file size). Previously computed analysis results are re-used in multiple expressions. Additional performance improvements (e.g., use of computed sub-expressions for analysis) will be implemented as the dashboard is extended.

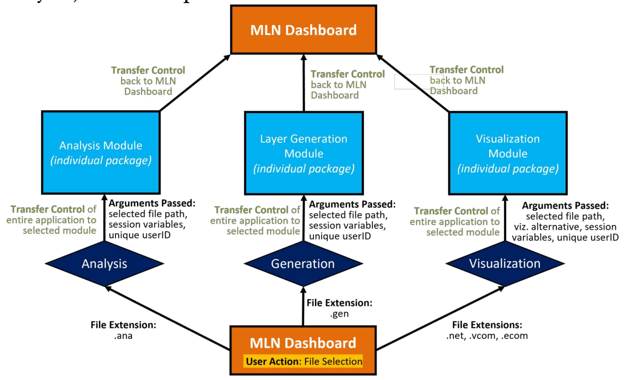

MLN Dashboard
Transfer Control back to MLN Dashboard
Transfer Control back to MLN Dashboard
Transfer Control back to MLN Dashboard
Analysis Module (individual package)
Layer Generation Module (individual package)
Visualization Module (individual package)
Transfer Control of entire application to selected module
Arguments Passed: selected file path, session variables, unique userID
Transfer Control of entire application to selected module
Arguments Passed: selected file path, session variables, unique userID
Transfer Control of entire application to selected module
Arguments Passed: selected file path, viz. alternative, session variables, unique userID
Analysis
Generation
Visualization
File Extension: .ana
File Extension: .gen
File Extensions: .net, .vcom, .ecom
MLN Dashboard
User Action: File Selection

Figure 5: **User Control Flow (*bottom to top*)**

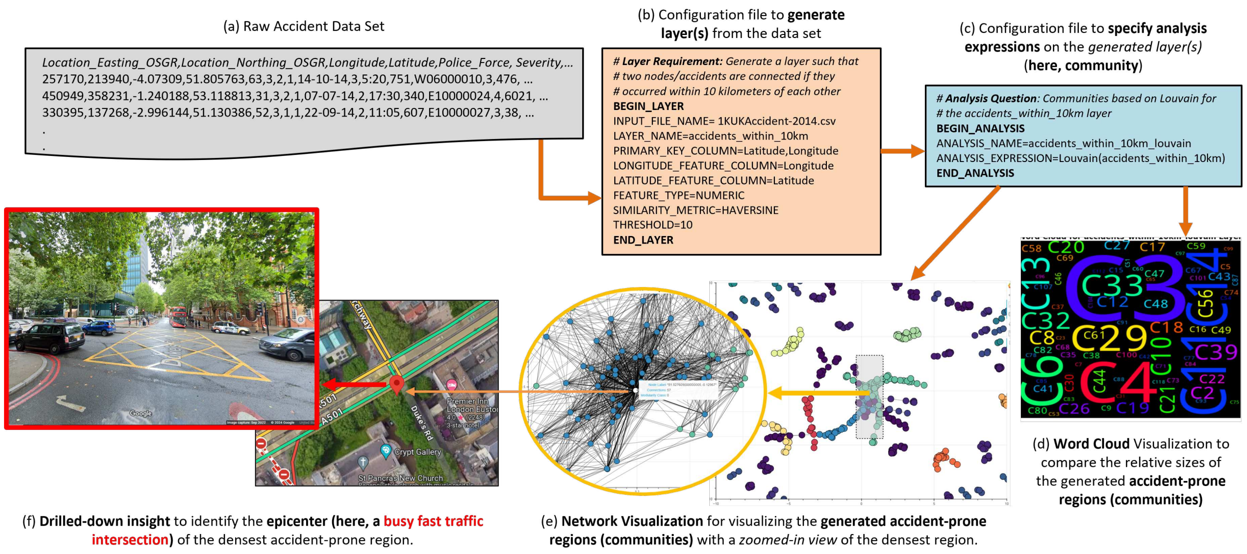
The dashboard is hosted on an Nginx Web Server 1.20.1 on a Linux machine. The technical stack used to implement the various components - (i) **Flask:** for the back-end server, (ii) **JavaScript:** for personalized scripts like conditional loading, (iii) **SQLite3:** for persistent storage of user information, and (iv) **Python and C++** packages: for different components of the three modules.

Currently, in the analysis module, community detection is supported using six packages: **Louvain, Infomap, Fast Greedy, Walktrap, Multilevel**, and **Leading Eigenvector**. **HoMLN community detection** algorithms based on Boolean compositions developed by us [16, 18] have been integrated, and other developed algorithms for HoMLN and HeMLN community [17], centrality [11, 12, 15], and substructure [3] analysis are being integrated.

4 SAMPLE MLN-DASHBOARD USE CASES

Here, we discuss how the dashboard can help users obtain deeper insight into their complex data sets. Some useful insights for an accident data set that can help city planners to take safety precautions are to find the (i) accident-prone regions, (ii) days of the week and time periods that witness an increased number of accidents, (iii) location types that become the epicenter of accidents, etc.

Figure 6 shows the workflow from a *raw accident data set* to *the final drilled-down visualized analysis* insight, supported by the interface shown in figure 2. **Figure 6 (a)** shows the UK Accident data set [1] snapshot, which for every accident occurrence captures the location (latitude, longitude), light conditions, weather conditions, date of occurrence, and so on. The user-specified configuration file in **Figure 6 (b)** connects two accident nodes if they have occurred within 10 km of each other based on Haversine distance to generate a layer. Then, the configuration file in **Figure 6 (c)** detects communities (i.e., in this case accident-prone regions) using the Louvain algorithm ([2]) with the output stored as a text file containing *(edge, communityID)* pairs. These accident-prone regions (communities) have been visualized using a word cloud to understand the relative severity of the regions in **figure 6 (d)**. Alternatively, network visualizations of the accident-prone regions with zoom-in, hover, and


(a) Raw Accident Data Set
(b) Configuration file to **generate layer(s)** from the data set
(c) Configuration file to **specify analysis expressions** on the *generated layer(s)* (**here, community**)
(d) **Word Cloud** Visualization to compare the relative sizes of the generated **accident-prone regions (communities)**
(e) **Network Visualization** for visualizing the **generated accident-prone regions (communities)** with a *zoomed-in view* of the densest region.
(f) **Drilled-down insight** to identify the **epicenter (here, a busy fast traffic intersection)** of the densest accident-prone region.

```
Location_Easting_OSGR,Location_Northing_OSGR,Longitude,Latitude,Police_Force, Severity,...
257170,213940,-4.07309,51.805763,63,3,2,1,14-10-14,3,5:20,751,W06000010,3,476, …
450949,358231,-1.240188,53.118813,31,3,2,1,07-07-14,2,17:30,340,E10000024,4,6021, …
330395,137268,-2.996144,51.130386,52,3,1,1,22-09-14,2,11:05,607,E10000027,3,38, …
.
.
```

```
# Layer Requirement: Generate a layer such that
# two nodes/accidents are connected if they
# occurred within 10 kilometers of each other
BEGIN_LAYER
INPUT_FILE_NAME= 1KUKAccident-2014.csv
LAYER_NAME=accidents_within_10km
PRIMARY_KEY_COLUMN=Latitude,Longitude
LONGITUDE_FEATURE_COLUMN=Longitude
LATITUDE_FEATURE_COLUMN=Latitude
FEATURE_TYPE=NUMERIC
SIMILARITY_METRIC=HAVERSINE
THRESHOLD=10
END_LAYER
```

```
# Analysis Question: Communities based on Louvain for
# the accidents_within_10km layer
BEGIN_ANALYSIS
ANALYSIS_NAME=accidents_within_10km_louvain
ANALYSIS_EXPRESSION=Louvain(accidents_within_10km)
END_ANALYSIS
```

Figure 6: Showcasing the use of the MLN-Dashboard to model, analyze, drill-down and visualize a raw accident data set.

pan facility help us understand the relationship strength within the *densest* communities (**Figure 6 (e)**). Finally, **figure 6 (f)** gives us drilled-down insight to identify that the epicenter of the densest accident-prone region (i.e., node with the highest degree in the densest community) is a fast traffic intersection on the Google Maps using the latitude, longitude node label.

The MLN-Dashboard has also been used to analyze other raw data sets, such as *IMDb* to find the groups of most similar genre-based movies and the highly rated actors who have never worked together; *US Airline* data to identify airlines hubs or cities from which you can reach other cities with least number of hops using closeness centrality analysis, and so on.

5 CONCLUSIONS AND FUTURE PLAN

Our MLN dashboard offers a web-based, interactive approach for performing complex data analysis and visualizing results. In addition to layer generation and analysis, multiple visualization options are provided to the user, enhancing their understanding of the results. We are in the process of extending the modules by adding more analysis algorithms and visualization alternatives, as well as additional features like selective data and result sharing.

ACKNOWLEDGMENTS

This work was supported by NSF award #1955798 and #1956373.

REFERENCES

[1] 2014. Road Safety - Accidents 2014. https://data.gov.uk/dataset/road-accidents-safety-data/resource/1ae84544-6b06-425d-ad62-c85716a80022
[2] Vincent D. Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of community hierarchies in large networks. *CoRR* abs/0803.0476 (2008).
[3] Kiran Bolaj, Abhishek Santra, and Sharma Chakravarthy. 2024. Substructure Discovery in Heterogeneous Multilayer Networks. In *IEEE International Conference on Knowledge Graph, ICKG 2023, Shanghai, China, December 1-2, 2023.* IEEE, 1–8.
[4] Antoni Calvó-Armengol and Yves Zenou. 2004. Social networks and crime decisions: The role of social structure in facilitating delinquent behavior. *International Economic Review* 45, 3 (2004), 939–958.
[5] Manlio De Domenico, Andrea Lancichinetti, Alex Arenas, and Martin Rosvall. 2015. Identifying modular flows on multilayer networks reveals highly overlapping organization in interconnected systems. *Physical Review X* 5, 1 (2015).
[6] M. De Domenico, M. A. Porter, and A. Arenas. 2014. MuxViz: a tool for multilayer analysis and visualization of networks. *Journal of Complex Networks* (2014).
[7] Andrew L Hopkins. 2008. Network pharmacology: the next paradigm in drug discovery. *Nature chemical biology* 4, 11 (2008), 682.
[8] M. Kivelä, A. Arenas, M. Barthelemy, J. P. Gleeson, Y. Moreno, and M. A. Porter. 2013. Multilayer Networks. *CoRR* abs/1309.7233 (2013).
[9] Kanthi Komar, Abhishek Santra, Sanjukta Bhowmick, and Sharma Chakravarthy. 2020. EER→MLN: EER Approach For Modeling, Mapping, And Analyzing Complex Data Using Multilayer Networks (MLNs). In *ER 2020*.
[10] Pablo Lara-Martínez, Bibiana Obregón-Quintana, CF Reyes-Manzano, Irene López-Rodríguez, and Lev Guzmán-Vargas. 2022. A multiplex analysis of phonological and orthographic networks. *Plos one* 17, 9 (2022), e0274617.
[11] Kiran Mukunda, Anamitra Roy, Abhishek Santra, and Sharma Chakravarthy. 2023. Stress Centrality in Heterogeneous Multilayer Networks: Heuristics-Based Detection. In *BigDataService 2023, Athens, Greece, July 17-20*. IEEE.
[12] Kiran Mukunda, Abhishek Santra, and Sharma Chakravarthy. 2023. The Challenge of Finding Degree Centrality Nodes in Heterogeneous Multilayer Networks. In *Proceedings of SEBD 2023, Galzignano Terme, Italy, July 2-5, 2023.*
[13] MultinetX 2019. multiNetX github Page. https://github.com/nkoub/multinetx/blob/master/README.md.
[14] Tarmo Nurmi, Arash Badie Modiri, Corinna Coupette, and Mikko Kivelä. 2024. pymnet: A python library for multilayer networks. *Journal of Open Source Software* 9, 99 (2024), 6930.
[15] Hamza Reza Pavel, Anamitra Roy, Abhishek Santra, and Sharma Chakravarthy. 2023. Closeness Centrality Detection in Homogeneous Multilayer Networks. In *Proceedings of IC3K 2023, KDIR, Rome, Italy, November 13-15, 2023.*
[16] A. Santra, S. Bhowmick, and S. Chakravarthy. 2017. Efficient Community Re-creation in Multilayer Networks Using Boolean Operations. In *Int. Conf. on Computational Science, Zurich, Switzerland.* 58–67.
[17] Abhishek Santra, Sanjukta Bhowmick, and Sharma Chakravarthy. 2025. CommPlex: Community in MultiPlexes - Definition and a Suite of Algorithms for Analysis. In *ICCS 2025*, Vol. 15904. Springer, 3–18.
[18] Abhishek Santra, Sanjukta Bhowmick, Fariba Afrin Irany, Kamesh Madduri, and Sharma Chakravarthy. 2023. Efficient Community Detection in Multilayer Networks using Boolean Compositions. *Frontiers in Big Data* (2023).
[19] Abhishek Santra, Kanthi Komar, Sanjukta Bhowmick, and Sharma Chakravarthy. 2022. From base data to knowledge discovery – A life cycle approach – Using multilayer networks. *Data Knowledge Engineering* (2022), 102058.
[20] Chuan Shi, Yitong Li, Jiawei Zhang, Yizhou Sun, and S Yu Philip. 2017. A survey of heterogeneous information network analysis. *IEEE Trans. Knowl. Data Eng.* 29, 1 (2017), 17–37.
[21] Blaz Skrlj, Jan Kralj, and Nada Lavrac. 2019. Py3plex toolkit for visualization and analysis of multilayer networks. *Appl. Netw. Sci.* 4, 1 (2019), 94:1–94:24.
[22] Xuan-Son Vu, Abhishek Santra, Sharma Chakravarthy, and Lili Jiang. 2019. Generic Multilayer Network Data Analysis with the Fusion of Content and Structure. In *CICLing 2019, La Rochelle, France.*
[23] Zhen Wang, Lin Wang, Attila Szolnoki, and Matjaž Perc. 2015. Evolutionary games on multilayer networks: a colloquium. *The European physical journal B* 88, 5 (2015), 124.