A Visual Multivariate Dynamic EgoCentric Network Exploration Tool

by Hasan Alp Boz

Submitted to the Graduate School of Engineering and Natural Sciences in partial fulfillment of the requirements for the degree of Master of Science

Sabanci University
July 2019

A Visual Multivariate Dynamic EgoCentric Network Exploration Tool

APPROVED BY:

Assoc. Prof. Dr. Selim Balcısoy ....
(Thesis Supervisor)

Prof. Dr. Burçin Bozkaya ....

Keziban
Asst. Prof. Dr. Günce Orman ....

DATE OF APPROVAL: 18/07/2019

A Visual Multivariate Dynamic EgoCentric Network Exploration Tool

Hasan Alp Boz

Computer Science and Engineering, Master's Thesis, 2019
Thesis Supervisor: Selim Balcısoy

Keywords: Exploratory Visual Analysis, Multivariate Dynamic Networks, Egocentric Analysis

Abstract

Visualizing multivariate dynamic networks is a challenging task. The evolution of the dynamic network within the temporal axis must be depicted in conjunction with the associated multivariate attributes. In this thesis, an exploratory visual analytics tool is proposed to display multivariate dynamic networks with spatial attributes. The proposed tool displays the distribution of multivariate temporal domain and network attributes in scattered views. Moreover, in order to expose the evolution of a single or a group of nodes in the dynamic network along the temporal axis, an egocentric approach is applied in which a node is represented with its neighborhood as an ego-network. This approach allows users to observe a node's surrounding environment along the temporal axis. On top of the traditional ego-network visualization methods, such as timelines, the proposed tool encodes ego-networks as feature vectors consisting of the domain and network attributes and projects them onto 2D views. As a result, distances between projected ego-networks represent the dissimilarity across temporal axis in a single view. The proposed tool is demonstrated with a real-world use case scenario on merchant networks obtained from a one-year long credit card transactions.

Çok Değişkenli Dinamik Beniçinci Ağların Görsel Keşif Aracı

Hasan Alp Boz

Bilgisayar Bilimi ve Mühendisliği, Yüksek Lisans Tezi, 2019
Tez Danışmanı: Selim Balcısoy

Anahtar Kelimeler: Görsel Keşif Analizi, Çok Değişkenli Dinamik Ağlar, Beniçinci Analiz

Özet

Çok değişkenli dinamik ağları görselleştirmek zorlu bir iştir. Dinamik ağların zaman ekseni içindeki evrimi, ilişkili çok değişkenli niteliklerle birlikte gösterilmelidir. Bu tezde, uzamsal niteliklere sahip çok değişkenli dinamik ağları görüntülemek için görsel keşif analiz aracı önerilmiştir. Önerilen araç, zamansal çok değişkenli alan ve ağ özelliklerinin dağınık görünümlerde dağılımını gösterir. Ayrıca, geçici ağ boyunca tek bir veya bir grup düğümün dinamik ağdaki evrimini açığa çıkarmak için, bir düğümün mahallesi ile beniçinci ağı olarak temsil edildiği beniçinci merkezli bir yaklaşım uygulanmaktadır. Bu yaklaşım, kullanıcıların bir düğümün içinde bulunduğu ortamı zaman ekseni boyunca gözlemlemelerine izin verir. Zaman çizelgeleri gibi geleneksel beniçinci ağ görselleştirme yöntemlerinin yanı sıra, önerilen araç beniçinci ağları, alan ve ağ özelliklerinden oluşan nitelik vektörleri olarak kodlar ve bunları 2B görünümlere yansıtır. Sonuç olarak, yansıtılan beniçinci ağlar arasındaki mesafeler, farklılıkları tek bir gösterimde temsil eder. Önerilen araç bir yıllık kredi kartı işlemlerinden elde edilen ticari ağlar ile gösterilmiştir.

To my beloved family...

ACKNOWLEDGEMENTS

I would like to express my sincere gratitudes to my thesis supervisor Assoc. Prof. Selim Balcısoy for his constant support, patience and understanding throughout the past two years. Without his encouragement and guidance, It would not be possible to complete this thesis.

In addition, I would like to thank Prof. Burçin Bozkaya, Mohsen Bahrami and Yoshihiko Suhara for their invaluable comments and feedbacks on the developed tool. Lastly, I would like to thank Asst. Prof. Günce Orman for her presence in the thesis jury.

I thank Mert Gürkan, Gökhan Göktürk, Mehmet Çagrı Çalpur, Ece Egemen and Emir Artar for the good times in the lab. And my sincere thanks goes to Yasin Fındık, for his friendship and company.

And finally, I would like to thank my family for their support throughout my entire life, without them nothing would be possible.

TABLE OF CONTENTS

List of Figures x

List of Tables xiii

1 Introduction 1

2 Related Work 4

2.1 Visual Exploration Tools 4

2.2 Network Visualization 5

2.3 Time Domain Analysis 9

2.4 Ego-Network Visualization 12

3 Visual Exploration Tool 15

3.1 Multivariate Dynamic Networks 15

3.2 System Design 16

3.2.1 Exploratory Visual Analytics 16

3.2.2 Interface Design Rationale 16

3.2.3 Visual Components and Interactions 18

3.2.3.1 Network Overview Component 18

3.2.3.2 Spatial Encoding Component 22

3.2.3.3 Attribute Distribution Component 23

3.2.3.4 Pixel Display Component 25

3.2.3.5 Ego-Network Component 26

3.2.3.6 Visual encoding and filtering component 29

3.3 System Architecture 29

3.3.1 Data Model 30

3.3.2 Communication Between Components 31

4 Use Case: Merchant Network 33

4.1 Data Overview 33

4.1.1 Background 33

4.1.2 Transaction Data Set 34

4.1.3 Merchant Network 35
4.2 Use Case Scenarios 38

5 Conclusion & Future Work 42

Bibliography 45

List of Figures

2.1 Different layout options for displaying networks all showing the same data as Beck et al. [17] argues. 6

2.2 Juxtaposed views representing consecutive snapshots of the dynamic network. 7

2.3 Space-Time cube. 9

2.4 Time cutting operation on space-time cube. 10

2.5 Napoleon's march on Russia as a time flattening operation [89]. March is depicted as bi-directional. Yellow area stands for the advance phase with size representing the number of soldiers. Moreover, the black area shows the retreat phase, again size representing the number of existing soldiers. 11

2.6 The evolution of global temperatures between 1884 and 2012. Each dot in the figure represents a year. Clustered dots, or years, share similar temperature values [14]. 11

2.7 A tree-based layout in which ego node is placed at the center. The levels of the encircling rings represent time steps as argued by [34]. . 12

2.8 Interface of *EgoSlider* developed by Wu et al. [96]. Ego-network similarities are displayed on the vertical view on the left. In the middle view, users are able to observe the network attribute evolution with specialized glyph based elements, as a result of a time flattening operation. 13

2.9 Interface of *Segue* developed by Law et al. [60]. Instead of placing ego-networks in a time step manner with temporal feature vectors, they assign event-based features covering the whole temporal axis. On the left, the proximities between ego-networks, dots, represent how similar they are in terms of an event defined by the user. The view in the middle allows users to inspect the evolution of individual ego-networks. 14

3.1 The designed user interface. a) The spatial component depicts the encoding between the network and a geographical map. b) The general network overview. c) The distribution component for domain and network features. d) The pixel display component to visualize temporal changes in the domain and network features. e) The ego-network component to display the ego-network evolution. f) Visual encoding and filtering component. 17

3.2 Each dot represents an ego-network, U_k^t, of node k at time step t. The distance between two dots serves as the similarity in terms of combined domain and network attributes. 17

3.3 The temporal changes in both domain and network attributes for a selected group of nodes in the network. 17

3.4 The initial view of the network overview component with a node-link layout computed with force-directed velocity verlet integration. 19

3.5 The result of click and hover interactions on the network layout. A node's directly connected neighbors are highlighted on hover. In addition, a tooltip is displayed with informative properties. 20

3.6 Multiple nodes in the network are selected with a rectangular brush. As a result, the temporal domain and network attributes of the selected nodes are displayed in designated views. 20

3.7 On the left, communities in the network layout are utilized to color the nodes. In addition, nodes can be colored with respect to one of their domain attributes as displayed in the figure on the right. Furthermore, shape and size elements are employed to encode the rest of the multivariate attributes. 21

3.8 A general view from the spatial encoding component. The spatial attributes of the nodes in the network layout are presented in a map. 22

3.9 Domain attributes can be encoded on the markers. Nodes on the exact location can be separated on click. 23

3.10 Initial view of the attribute distribution component. 24

3.11 Brushing on a particular attribute display would highlight the same data points on other attribute displays as well. 24

3.12 Changes between consecutive time steps are displayed in the pixel display component. 25

3.13 Sorted pixel display based on the activity counts. 26

3.14 Visual encodings of the domain attributes on display. 27

3.15 Hover on a row in the temporal attribute change area activating other components in the tool. 27

3.16 Interactions between components. 29

3.17 Data transformation pipeline. 30
3.18 Data retrieval in the exploratory tool are carried out with HTTP requests. 30
3.19 Propagation of the clicked node's ID between source and target components. 31

4.1 An example transaction schema and merchant network generation. On left, two customers make transactions in different merchants. On right, the resulting merchant network is displayed. 34
4.2 Spatial distribution of the merchants. 34
4.3 Multivariate dynamic network creation from transaction data set. . . 36
4.4 Resulting node attributes in the form of a node-link file. Well-being attributes start with *last* prefix. 37
4.5 Number of nodes in temporal networks along with associated network density values. 37
4.6 Topological and spatial distribution of the top performing merchants in terms of degree centrality. 38
4.7 A group of successful merchants attribute and spatial distributions. . 39
4.8 A significant distance takes place between July and August. Users can observe the differences in detail with the help of the timeline view. 41

List of Tables

3.1 The network features used in ego-network feature vectors. 28

4.1 Sample transaction data. Each row in the table corresponds to an expenditure. 35

4.2 Additional domain attributes used in the merchant ego-network feature vectors. 40

Chapter 1

Introduction

Visualization plays a vital role to understand and analyze the data. Visual encodings of data help viewers to grasp the underlying dynamics, such as outliers, patterns and trends, that remain hidden with tabular representations. In addition, data is internalized by the viewers much quicker with the presence of visualizations. One famous example is Anscombe's Quartet [81] in which four different data sets with distinct trends possess almost identical statistical properties but only differ on a 2-D layout. However, in most cases, the hidden properties do not reveal themselves within a single layer of visualization. Further techniques and tools are required to present the internal dynamics of the data.

Visual analytics is one of the substantial fields in the information visualization community, focusing on analytical sessions by providing a visual environment in which users are able to interact with multiple coordinated views that depicts the data. Existing studies cover a wide range of domains such as sleep behavior analysis [24] to outlier detection [100]. Exploratory visual analytics aim to create visual platforms in which users are able to gain insights about the data, test and form hypotheses and observe certain phenomena such as patterns, trends and outliers in the data. In this thesis, an exploratory visual analysis tool is developed to display multivariate dynamic networks with an egocentric network approach.

Network data is ubiquitous, for instance, social networks, communication networks and epidemiological networks. Networks consist of entities named nodes that are connected to each other over relations named links. In a real-world scenario, nodes in a network may represent many different entities such as people, locations and routers; while links stand for the relationship between them such as friendship, roads and signals. Understanding the topological structure within the network help analysts to acquire useful insights such as clusters within the network. However, nodes or links in the network may have domain related multivariate attributes that have the potential to reveal more insights to the analyst. For instance, Nobre et al. [72] displays topological and domain attributes to enhance the analyst's ability to better comprehend the network.

As time forwards, new nodes and links may emerge while some die, changing the structure of the network. As a result, a collection of consecutive networks are obtained, which, in the end, yields a multivariate dynamic network. Essentially, a multivariate dynamic network consists of chronologically ordered multivariate networks. Depicting the transitions between consecutive time steps in terms of topological and domain attributes is a challenging task. Existing techniques utilize network layouts juxtaposed to attribute display [71] or embed attributes within the network layout [43].

Existing multivariate dynamic network exploration tools focus on the evolution of the overall network. However, narrowing down to a particular or a group of nodes in the network may reveal invaluable insights. An ego-network consists of an ego node with its directly connected neighbors and links connecting them. For instance, in a social network such as Facebook, the ego-network of a user consists of other users who are directly connected. The demographic attributes of the connected users help analysts to better understand the environment in which the select user is located. Analyzing the ego-network of the selected user along a time span, in which new connections occur and existing ones dissolve, would depict the evolving environment around the selected user. For example, Halgin et al. [48] analyze the interactions between customers with the help of ego-networks over the course of a time period to observe collective churn behaviors.

There exist many studies on the evolution of ego-networks, especially in social sciences [26, 28, 33, 52]. For instance, Fares et al. [33] study ego-network evolution to analyze the difference between healthy and diseased people in terms of the perception of care. Furthermore, Lucia et al. [66] utilize ego-networks to classify text messages.

Ego-network visualization literature focuses on displaying the topological evolution across the temporal axis. However, depicting the evolution of the domain attributes reveal additional insights invaluable to the users. In order to visualize multivariate dynamic network from an egocentric approach, an exploratory visual analysis tool is developed in this thesis. The developed tool focuses on displaying both domain and network attributes. In addition, the evolution display of ego-networks aims to answer both mesoscopic and microscopic questions [96].

In this thesis, a visual exploratory tool is presented, which enables users to explore multivariate dynamic networks from an egocentric approach and observe individual nodes' evolution display along the temporal axis. Also, in order to further expose the relationship between nodes in the network, a spatial encoding component is incorporated into the tool. With the help of spatial encoding component, users are able to observe the proximities between nodes in the network.

The contributions of this thesis are in two folds; a visual exploratory analysis tool developed for multivariate dynamic networks with spatial features and displaying individual ego-network evolution on a 2D plane based on both domain and network features.

The remaining of the thesis is structured as follows: in Chapter 2, a comprehensive literature review is presented based on the underlying components. The developed visual exploratory analysis tool is explained in Chapter 3. A real-world use case scenario, on merchant networks, is presented in Chapter 4. Finally, Chapter 5 consists of the conclusion and future work.

Chapter 2

Related Work

In this chapter, the literature will be elaborately examined. First, Section 2.1 will go through the existing studies in exploratory visual analytics tools. The studies on dynamic network visualization field will be examined in Section 2.2. Next, the rationale behind spatial layouts and the existing studies will be analyzed in Section 2.3. Finally, Section 2.4 will elaborate on ego-network visualizations.

2.1 Visual Exploration Tools

Exploratory visual analytics is one of the central research fields in visualization studies. Existing works encompass a wide range of domains; from exploring drug-drug interactions [55] to comparing CVs [35]. Each domain requires its own set of goals and tasks. Battle et al. [16] argues that due to the involved open-ended processes and wide of range of domains, exploratory visual analytics is difficult to be formally characterized and put in a framework. Nevertheless, the primary goal of exploratory visual analytics is to create a visual environment in which users are able to obtain a deeper understanding of the data, develop insights, inspect specific phenomenons such as patterns, trends and outliers in the data, test or form hypothesis by collecting visual evidence [16].

Battle et al. [16] observes three common themes in the current explanatory visual analytics literature; *exploratory goals, structure and performance.* In the exploratory goals theme, studies are evaluated in terms of their assumptions and formations on the desired goals. Next, within the structure theme, functionality paradigms and analysis tasks are considered. Lastly, system performance, such as responsiveness and latency, is analyzed in the performance theme.

From a general perspective, the goal of a visual exploration analysis tool is defined vaguely as helping users gain further insights regarding the data [30, 46, 54, 59]. However, depending on the data and the developed tool, existing studies in the visual exploration analysis have different assumptions and formations regarding the goals of an exploratory analysis because of the vague definition. Battle et al. [16] state two different approaches. In the bottom-up approach, exploration is evaluated as an open-ended process and does not have clear steps [8, 75, 98]. Users gain insights, develop hypotheses and observe different phenomena as they inspect the tool. On the other hand, with the top-down approach, users have specific tasks and goals [42, 56, 73]. They act based on a priori hypotheses as Liu et al. [64] states. For instance, Siddiqui et al. [87] state a scenario in which scientists make use of their tool to observe specific patterns in the data. In the developed tool, the latter approach is embraced. Users have specific goals, such as forming hypotheses and observing certain phenomena in the data.

In the structure theme, the question is what phases, steps and mechanisms are needed to carry out the exploration task. Because of the vague assumptions and definitions as described before, there are different structure approaches followed by different studies. Shneiderman's [85] information-seeking mantra, overview first, zoom and filter, details on demand, is a common approach in many exploratory studies [15, 57]. Users observe the data from a general perspective and apply various transformations on the data such as visual encodings and filtering. And then, view the details of the transformed data on demand. At the same time, Gotz et al. [42] argue that users oscillate between two phases; browse and query data to extract and note down insights. In the developed tool, Shneiderman's approach is embraced by delegating phases to each component in the tool.

Finally, the performance theme evaluates the overall performance of the exploratory analysis system to be feasible in terms of responsiveness and flexibility in order to resonate with the pace of human thought [51, 64]. Liu et al. [64] conclude that high latency values over 500ms may impair the performance of the system. The performance details of the developed tool will be explained in Chapter 4 for the merchant network use case.

2.2 Network Visualization

At the core of the developed tool, multivariate dynamic network visualization takes place. Nobre et al. [71] present a comprehensive literature review on multivariate network visualization, while Beck et al. [17] exhibit an extensive literature review on dynamic network visualization. In this section, existing studies on both fields will be examined, in addition to the areas where they intersect.

Network visualization has been studied in detail since graphs, in general, are one of the central data structures in computer science. In a graph, entities are represented as vertices or nodes, and the relationships between these entities are encapsulated with edges or links. In order to visualize a graph, two major approaches

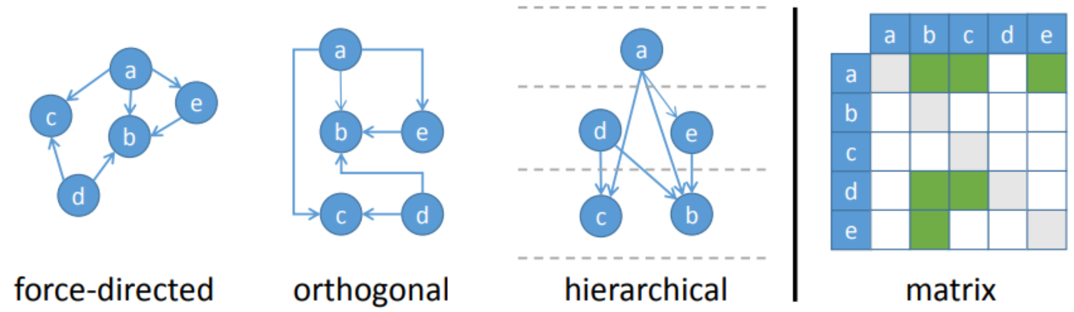


Figure 2.1: Different layout options for displaying networks all showing the same data as Beck et al. [17] argues.

are developed; namely node-link layout and matrix representation as in Figure 2.1. In a node-link layout, each vertex is represented with a visual element, such as dot, and edges with a line connecting associated vertices. Moreover, with a matrix layout, rows and columns stand for vertices while edges correspond to a cell where a row and column intersect.

Within a node-link layout, there exists different layout options as in Figure 2.1. With a force-directed layout, nodes are positioned that simulates physical forces between them. An orthogonal layout places edges between nodes on horizontal and vertical axes. Lastly, in order to display the hierarchical relationships between nodes, the hierarchical layout is utilized.

The layout models mentioned above are designed for static networks. The difference between a static and dynamic network is that nodes and links in a dynamic network may form or dissolve at any given time [17]. A dynamic network can be defined as $D = \langle G_0, G_1, ..., G_t \rangle$, which is a finite sequence of chronologically ordered temporal networks G_t where $G = (V, E)$ is defined by time step t, and V set of vertices and $E \subseteq \{(v_i, v_j) \,|\, v_i, v_j \in V\}$ consisting of vertex pairs. In order to capture the temporal changes in the network, different strategies are developed, namely animation and timeline views.

Within a single view, animated transitions between snapshots of the dynamic network at consecutive time steps help users to observe the changes [13, 22, 38, 68]. For instance, in Figure 2.2, the transitions between three consecutive snapshots, G_{t_1}, G_{t_2}, G_{t_3} are displayed along the horizontal axis.

There exist two major layout options in the animated transitions, namely general and special purpose layouts. In the general-purpose layout, there is no requirement or specification applied to the layout. However, several problems occur within a general-purpose layout. The source of these problems come from the fact that keeping track of the changes between transitions is a heavy cognitive load for the users [96]. Huang et al. [53] propose a force-directed approach in which focus node

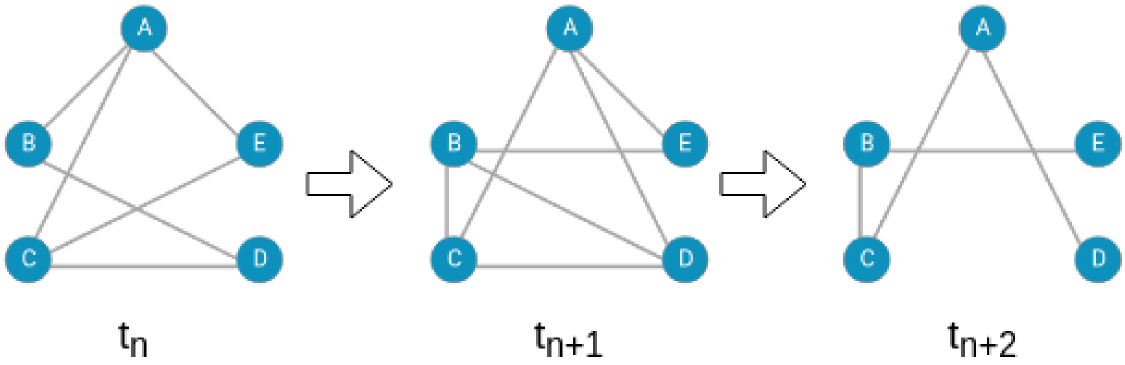


Figure 2.2: Juxtaposed views representing consecutive snapshots of the dynamic network.

is moved to the center, and then surrounding nodes are animated. Moreover, an incremental transition is developed by Friedrich et al. [36], in which dissolving nodes and edges are removed first and the positions of the remaining nodes and edges are optimized by a force-directed layout. Next, emerging nodes and edges are displayed within the existing layout. In order to depict the transition dynamics, Bach et al. [13] develops a staged transition between consecutive snapshots. Visual encodings are utilized to point out changes in each stage of the transition.

Second, during the transition, the positions of the emerging and dissolving nodes must be handled with care so that users can preserve their mental map [10]. Misue et al. [68] utilize for-directed layouts to stabilize the horizontal and vertical orderings in order to prevent overlapping nodes during the transition. Lee et al. [61] propose simulated annealing based approach to compute layouts. However, existing studies that rely solely on animated transitions are not able to alleviate the resulting cognitive load [60].

In a special-purpose layout, depending on the characteristics of the graph, different layouts are created to capture the dynamics between transitions. For instance, Frishman et al. [37] develop a layout specialized for clustered networks with encircling boxes around the clusters, so that the structure of the clusters are preserved. Moreover, Reitz et al. [76] utilize hierarchical node-link layout to encapsulate the hierarchical changes within the transitions.

In a timeline view, the whole snapshots of the dynamic network are visualized as adjacent static displays as in Figure 2.2. In contrast to animation based approaches, users are able to observe the complete evolution within a single view [90]. The rationale behind the timeline view is to place all consecutive snapshots in comparable fashion.

Juxtaposition is one of the layout options in which snapshots are displayed individually along a determined axis. Greilich et al. [44] places snapshots along the vertical axis to trace the changes. Moreover, Burch et al. [23] extend this idea by placing each time axis along the horizontal axis. Scaling is the main drawback in juxtaposed timeline views [17].

With a superimposition approach, nodes and links in trailing snapshots are stacked on top of each other. In order to distinguish the time step, visual encodings such as color, shape and stroke are utilized [31]. Dwyer et al. [29] make use of 3D cylinders to represent nodes and edges connecting nodes on different levels depicts temporal changes.

Animation based approaches contain a heavy cognitive load for users. In parallel, timeline views do not scale for large time periods [17]. Hybrid approaches aim to combine both approaches so that the mentioned problems are alleviated. For instance, Hadlak et al. [47] adapt animated transitions in a timeline. Moreover, Sallaberry et al. [80] apply the same technique to depict cluster evolutions. In the developed tool, a hybrid approach is embraced with a timeline view on the foreground. Details will be given in Chapter 3.

In multivariate networks, in addition to the underlying graph structure, there exists a finite set of attributes, $A = \{a_1, a_2, ..., a_n\}$, assigned to each node and/or edges. However, in this thesis, edges without attributes are assumed. Nobre et al. [71] present a comprehensive literature review on multivariate network visualization. They present their findings under two main topics, namely layouts and view operations.

As mentioned in the dynamic network visualization review, there exist multiple layout options to visualize networks. In order to visualize the attributes stored in a multivariate network, researchers developed further layouts to display these attributes. The topology-driven approach applies one of the layout options mentioned before, such as force-directed, orthogonal, and in order to display the attributes, visual encodings are utilized [39] or visualizations such as bar charts are embedded in the network elements [11]. For instance, Heer et al. [50] utilize photos and labels on the nodes.

In attribute-driven layouts, the positions of nodes are computed based on the values of the nodes. In other words, positions in the layout encode attributes, instead of visual elements. Wattenberg [95] determines the positions of the nodes in a layout based on two attributes in the data. Besides, Shneiderman et al. [86] extends this idea and create separate layouts for different node types and link connected nodes across layouts.

In addition to node-link layouts, matrix-based layouts are also utilized. Kerzner et al. [58] make use of an adjacency matrix in which rows and columns encode attributes of the network. In fact, Rufiange et al. [78] represents edge attributes as color encodings on the adjacency matrix.

Placing attributes within the network visualization as an integrated view enables users to observe both at the same time. However, such visualizations do not scale well under a large network. Naquin et al. [69] propose an integrated view approach in which genome variations are displayed with encircling attribute layer.

Attribute visualization and topology visualization are separated in juxtaposed views. Similar to the previous use cases, in a juxtaposed view, attributes and network topology views are placed next to each other. Guo [45] places a parallel coordinates view that displays attributes juxtaposed with topology view. The developed tool utilizes a juxtaposed view, in order to present network topology and attributes on separate views, so that users have a better comparison within each view.

2.3 Time Domain Analysis

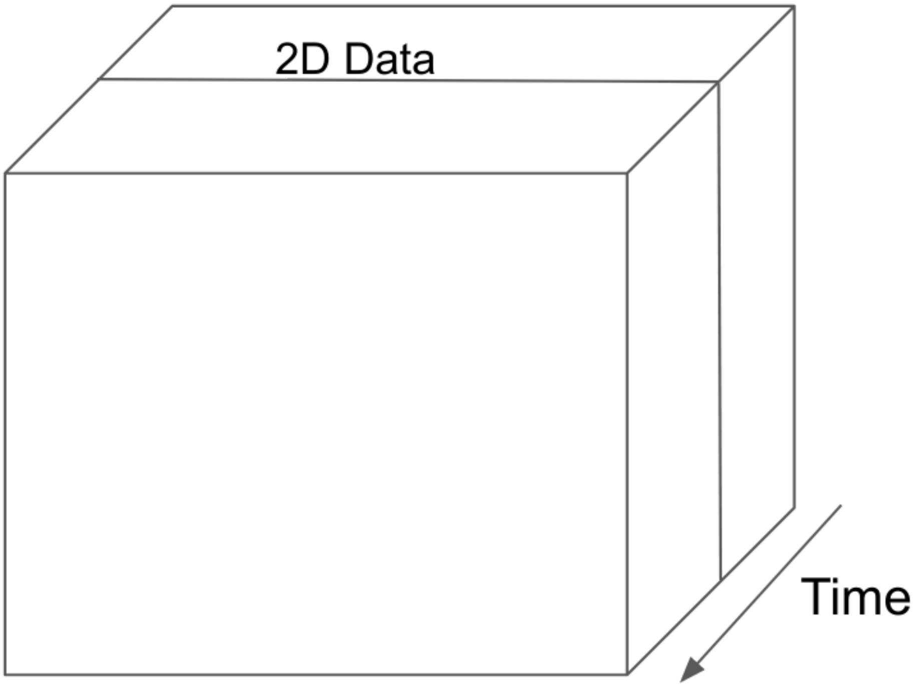


Figure 2.3: Space-Time cube.

As described in the previous section, a dynamic network is a sequence of networks ordered by temporal attributes. Existing layout strategies, such as animation and timeline views, aim to capture the attribute changes within temporal transitions. Bach et al. [12] propose a systematic approach to categorize time analysis operations with a space-time cube. One may think of space-time cube as a storage mechanism for both attributes along the time axis. Entities and associated attributes are stored on x and y-axes, respectively, while time goes along the z-axis.

Bach et al. [12] define time cutting operation as extracting a snapshot, in other words time slice, perpendicular to the time axis from the space-time cube. The extracted snapshot captures the whole attribute layer stored within the corresponding time step. In dynamic network visualization, animation approach is based on a time cutting operation in which snapshots belonging to the whole time steps are transitioned in an iterative manner.

Time juxtaposing is another operation based on time cutting operation. Multiple snapshots are extracted from the space-time cube and placed next to each other on a grid structure. In dynamic network visualization, timeline views are based on time juxtaposing operation. As a consequence, many dynamic network visualization studies utilize time juxtaposing operations [21, 65, 79].

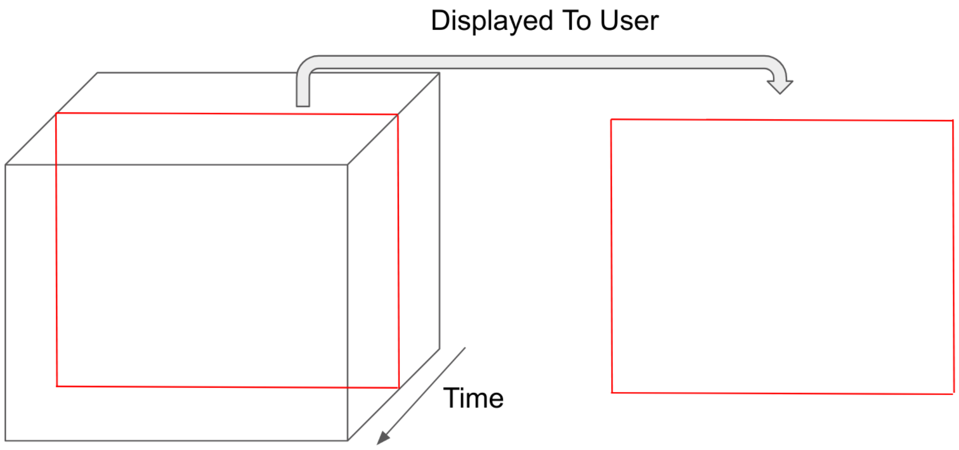


Figure 2.4: Time cutting operation on space-time cube.

In order to visualize the time axis as a whole, time flattening approach is utilized in which space-time cube is merged along its time axis and produce a single 2D view of the data, encapsulating all temporal changes [12]. The most famous example on time flattening approach is Napoleon's march on Russia [89] which utilize visual encodings for attributes along the horizontal axis to represent time and space.

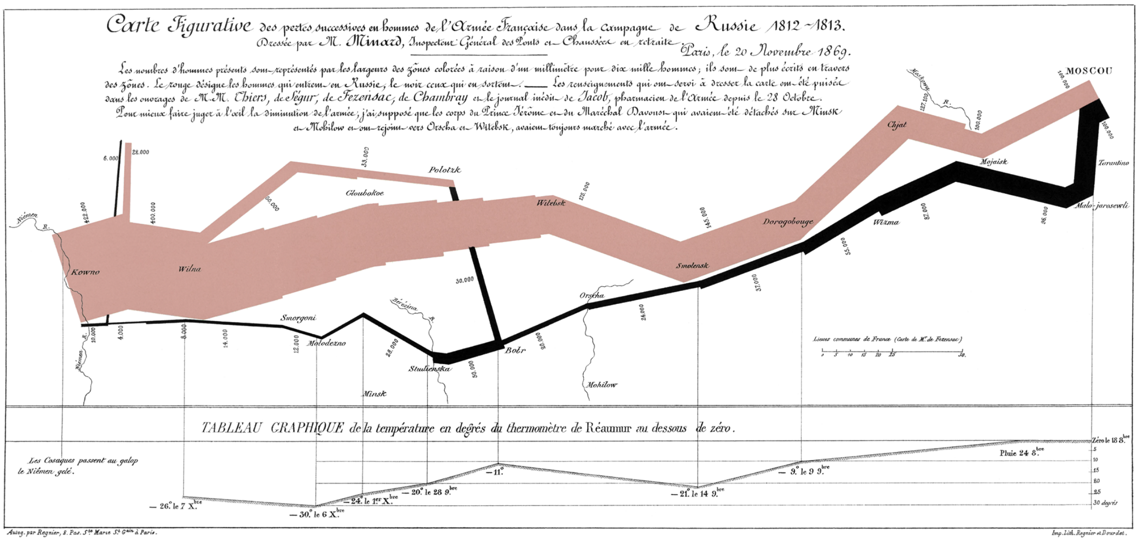


Figure 2.5: Napoleon's march on Russia as a time flattening operation [89]. March is depicted as bi-directional. Yellow area stands for the advance phase with size representing the number of soldiers. Moreover, the black area shows the retreat phase, again size representing the number of existing soldiers.

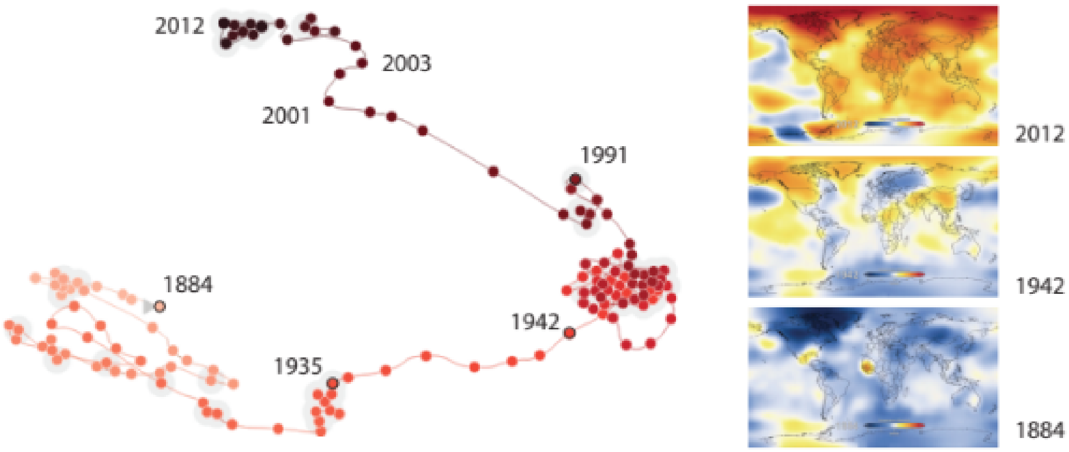


Figure 2.6: The evolution of global temperatures between 1884 and 2012. Each dot in the figure represents a year. Clustered dots, or years, share similar temperature values [14].

Observing the network evolution throughout the temporal axis is one of the primary goals in visual dynamic network analysis. Users should be able to compare consecutive time steps and understand how similar they are. In order to perceive the similarities between entities, Promann et al. [74] argue that proximity plays an active role for the users, defending the "near is similar" metaphor.

Based on the proximity perception and time flattening approach, Bach et al. [14] propose a data type independent time flattening approach to display the evolution along the temporal axis. The idea behind visualizing the evolution is to display the similarities between consecutive time steps. For each time step, a feature vector representing the data point is prepared, for instance, pixels stored in a snapshot extracted from a surveillance camera. Based on a dimensionality reduction technique, for instance, Multidimensional scaling, MDS, or t-SNE [93], applied to the feature vectors, temporal transitions are displayed on a 2D surface. In Figure 2.6, yearly global temperatures are displayed with MDS [19]. MDS is chosen by the authors since it aims to preserve the original distances on the 2D display between entities.

2.4 Ego-Network Visualization

An ego-network, $U_k^t = (V_k^t, E_k^t)$, is defined by an ego node, k, and a time step, t. The nodes in an ego-network consist of an ego node, k, and the directly connected nodes named alters, $V_k^t = \{ v_k^{ti} \,|\, 1 \leq i \leq n_k^t \}$ where n_k^t is the number of alters. Each edge, $e_j \in E_k^t$ has a weight, $w_{e_j} \in \mathbb{R}$. As a result, for a given node, k, the dynamic ego-network of k can be expressed as $\Delta_k = \{U_k^0, U_k^1, ..., U_k^t\}$ as an ordered temporal sequence of individual ego-networks.

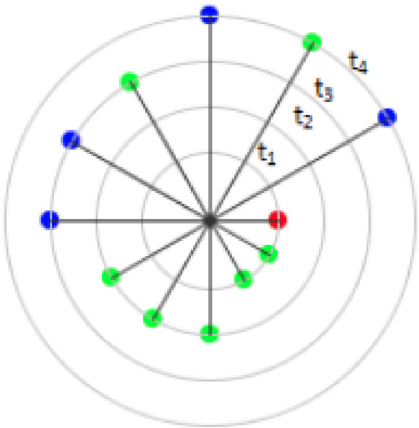
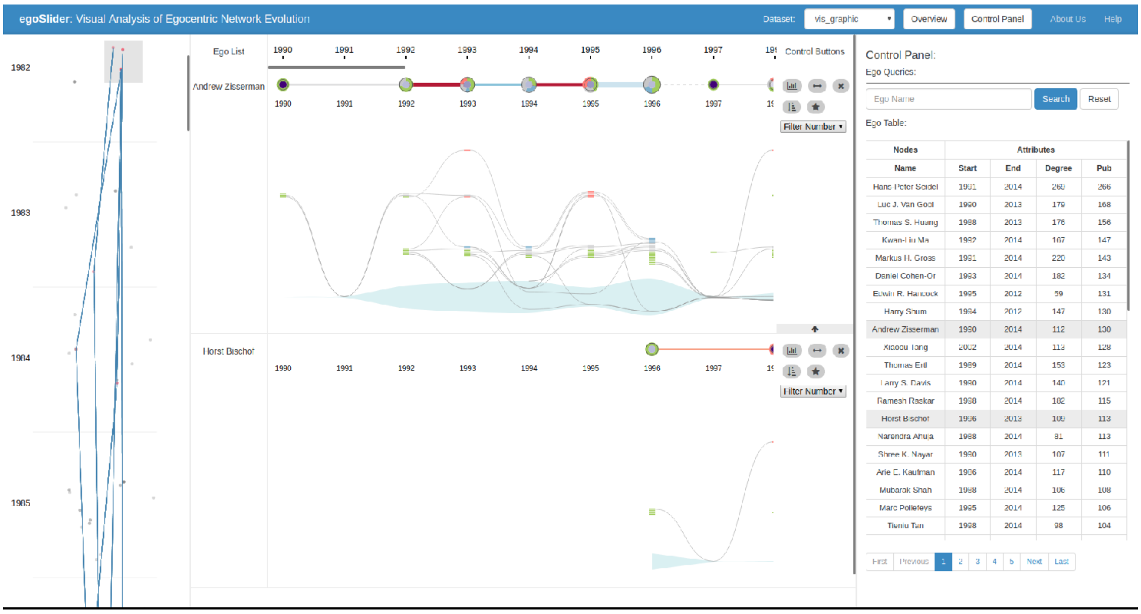


Figure 2.7: A tree-based layout in which ego node is placed at the center. The levels of the encircling rings represent time steps as argued by [34].

As a result of the underlying dynamic network structure, existing visual ego-network studies are based on the previously mentioned visualization techniques [34, 49, 62, 99]. For instance, He et al. [49] utilize a timeline view to display temporal ego-networks. In addition, supplementary views in the tool depict the statistical distributions of the network properties. In parallel, Farrugia et al. [34] make use of a timeline view with a hierarchical layout where ego is placed at the root as in Figure 2.7. Furthermore, Shi et al. [83] apply a time flattening operation and represent time along an axis. On the other hand, Liu et al. [63] display temporal ego-networks as a result of time cutting operation. However, the scope of these studies is focused on displaying a particular ego-network.

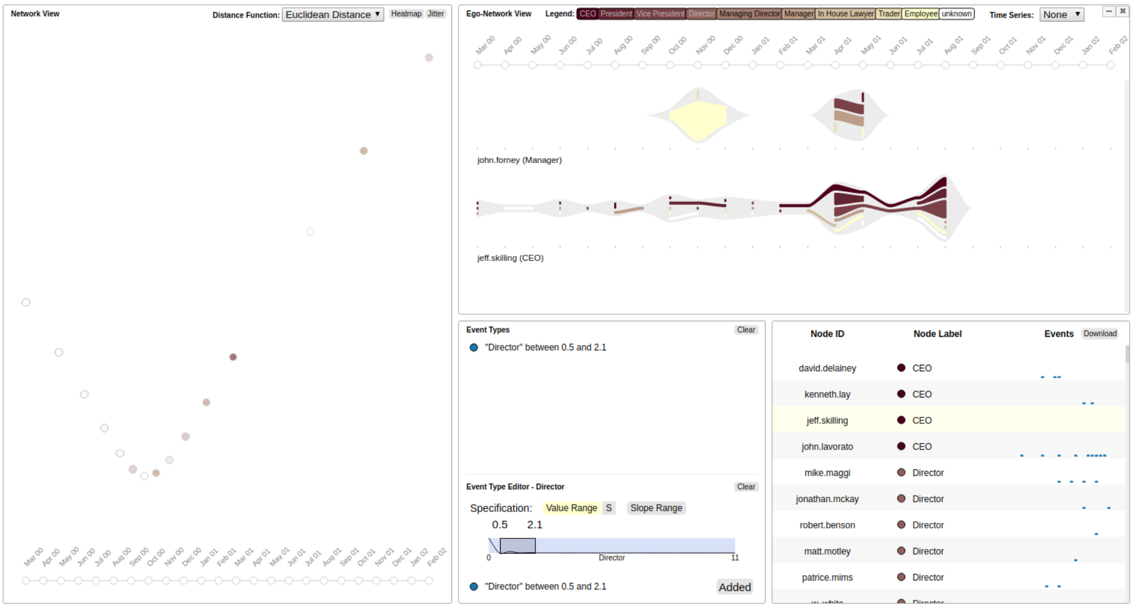
In order to characterize existing ego-network visualization studies into a framework, Wu et al. [96] defines three research question scopes regarding visual ego-network analysis, *macroscopic, mesoscopic and microscopic.* Macroscopic questions aim to find answers regarding overall ego-network evolution patterns hidden in the data. On the other hand, the mesoscopic scope deals with a group of ego-networks to be evaluated, while the microscopic scope is interested in a particular ego-network.



Nodes	Attributes			
Name	Start	End	Degree	Pub
Hans-Peter Seidel	1991	2014	269	266
Luc J. Van Gool	1990	2013	179	168
Thomas S. Huang	1988	2013	176	156
Kwan-Liu Ma	1992	2014	167	147
Markus H. Gross	1991	2014	220	143
Daniel Cohen-Or	1993	2014	182	134
Edwin R. Hancock	1995	2012	59	131
Harry Shum	1994	2012	147	130
Andrew Zisserman	1990	2014	112	130
Xiaoou Tang	2002	2014	113	128
Thomas Ertl	1989	2014	153	123
Larry S. Davis	1990	2014	140	121
Ramesh Raskar	1998	2014	102	115
Horst Bischof	1996	2013	109	113
Narendra Ahuja	1988	2014	81	113
Shree K. Nayar	1990	2013	107	111
Arie E. Kaufman	1986	2014	117	110
Mubarak Shah	1988	2014	106	108
Marc Pollefeys	1995	2014	125	106
Tieniu Tan	1998	2014	98	104

Figure 2.8: Interface of *EgoSlider* developed by Wu et al. [96]. Ego-network similarities are displayed on the vertical view on the left. In the middle view, users are able to observe the network attribute evolution with specialized glyph based elements, as a result of a time flattening operation.

In order to visualize the similarities between ego-networks, Wu et al. [96] embrace the approach developed by Bach et al. [14] which proposes to create feature vectors and display their proximities to each other on a 2D plane. For each ego-network, network specific features such as the number of alters, average tie strength and density are used. However, domain attributes are not considered. Also, similarities are computed for each time step. Overall similarities are neglected. A group of ego-networks can be traced by checking their evolving proximities; however, it is not the case for individual ego-networks since each time step has its own proximity layout.



Distance Function: Euclidean Distance ▾

Time Series: None ▾

Event Types

● "Director" between 0.5 and 2.1

Event Type Editor - Director

Specification: Value Range S Slope Range

0.5 2.1

● "Director" between 0.5 and 2.1 Added

Node ID	Node Label	Events
david.delainey	● CEO	
kenneth.lay	● CEO	
jeff.skilling	● CEO	
john.lavorato	● CEO	
mike.maggi	● Director	
jonathan.mckay	● Director	
robert.benson	● Director	
matt.motley	● Director	
patrice.mims	● Director	

Figure 2.9: Interface of *Segue* developed by Law et al. [60]. Instead of placing ego-networks in a time step manner with temporal feature vectors, they assign event-based features covering the whole temporal axis. On the left, the proximities between ego-networks, dots, represent how similar they are in terms of an event defined by the user. The view in the middle allows users to inspect the evolution of individual ego-networks.

In EgoSlider [96], users must scroll down to observe the ego-network evolution on the left view, which makes it difficult to compare distant time steps. Law et al. [60] defines an ego-network's feature vector based on user-defined events occurred along the temporal axis. The defined events depend on the increase and decrease in the number of alters whose types are entered by the user. However, they discard both raw domain and network specific features in the layout computation.

Chapter 3

Visual Exploration Tool

In this chapter, the proposed visual exploratory tool will be thoroughly covered. First, Section 3.1 will go through some of the definitions and notations on multivariate dynamic networks. Section 3.2 will explain the design rationale behind the tool and will go through each visual component in the tool and explain the interactions between them. Finally, Section 3.3 will illustrate the data model and the overall architecture in the tool.

3.1 Multivariate Dynamic Networks

A weighted multivariate network, N, possesses a weighted graph structure, $G = (V, E)$, that consists of a finite set of vertices, or nodes, V and a set of edges, or links, $E \subseteq \{(v_i, v_j) \,|\, v_i, v_j \in V\}$ consisting of vertex pairs with assigned weights, $w_{ij} \in \mathbb{R}$. In addition to the underlying graph structure, there exists a finite set of attributes, $A = \{a_1, a_2, ..., a_n\}$, assigned to each node and/or edges. However, in this thesis, edges without attributes are assumed.

In a dynamic network, on the other hand, nodes and edges may form and dissolve at any given time. In order to handle these phenomena, temporal multivariate networks are generated by taking snapshots of the dynamic network based on the domain specifications. As a result, a multivariate dynamic network is obtained, $D = \langle G_0, G_1, ..., G_t \rangle$, which is a finite sequence of chronologically ordered temporal multivariate networks.

As stated in previous chapters, rather than solely visualizing the multivariate dynamic networks, the visual display of the ego-network evolution is the main goal of the developed exploratory analysis tool. An ego-network, $U_k^t = (V_k^t, E_k^t)$, is defined by an ego node, k, and a time step, t. The nodes in an ego-network consist of an ego node, k, and the directly connected nodes named alters, $V_k^t = \{\, v_k^{ti} \,|\, 1 \leq i \leq n_k^t\}$ where n_k^t is the number of alters. Each edge, $e_j \in E_k^t$ has a weight, $w_{e_j} \in \mathbb{R}$. As a result, for a given node, k, the dynamic ego-network of k can be expressed as

$\Delta_k = \{U_k^0, U_k^1, ..., U_k^t\}$ as an ordered temporal sequence of individual ego-networks.

3.2 System Design

3.2.1 Exploratory Visual Analytics

Exploratory visual analytics is one of the central research fields in visualization studies. The primary goal of exploratory visual analytics is to create a visual environment in which users are able to obtain a deeper understanding of the data, inspect specific phenomenons such as patterns, trends and outliers in the data, test or form hypothesis in an iterative manner by collecting visual evidence. There are many existing studies [7, 59, 70, 97] that utilize exploratory visual analytics in a variety of different domains such as fraud elicitation [7] or air quality data [97]. As a result of the wide range of domains and involved open-ended processes, each study requires its own definition of exploratory visual analytics environment [16]. In this thesis, the developed exploratory visual analytics tool help users form hypotheses in an iterative manner and observe certain patterns, trends and outliers present in a multivariate dynamic network with an egocentric approach.

3.2.2 Interface Design Rationale

The developed exploratory tool base its design rationale on Shneiderman's [85] "Overview first, zoom and filter, details on demand" paradigm in which users first observe the general overview, select and extract a particular part of the data and then view the details. Each component in the developed tool serves a particular function in the paradigm.

Existing studies [9, 23, 83, 86] utilize sophisticated network layouts to visualize dynamic networks; however, they do not concentrate on the domain features and their evolution. Although there exists such exploratory tools [25, 72] that visualize domain features as well, the scope of such displays is quite narrow.

Most of the studies [84, 96, 99] on the visualization of ego-network evolution do not pay attention to the changes in the domain attributes. Ming et al.[60] utilize the changes in domain attributes to update ego-network similarities. However, they only consider the node label as the domain attribute and do not expand the attribute set.

As stated in the previous section, the goal of the developed exploratory tool is to help users form and test hypotheses and observe certain patterns, trends and outliers present in the multivariate dynamic network with an egocentric approach. In order to achieve this goal, the multivariate dynamic network must be comprehensively displayed in various aspects. The changes in the domain and network attributes must

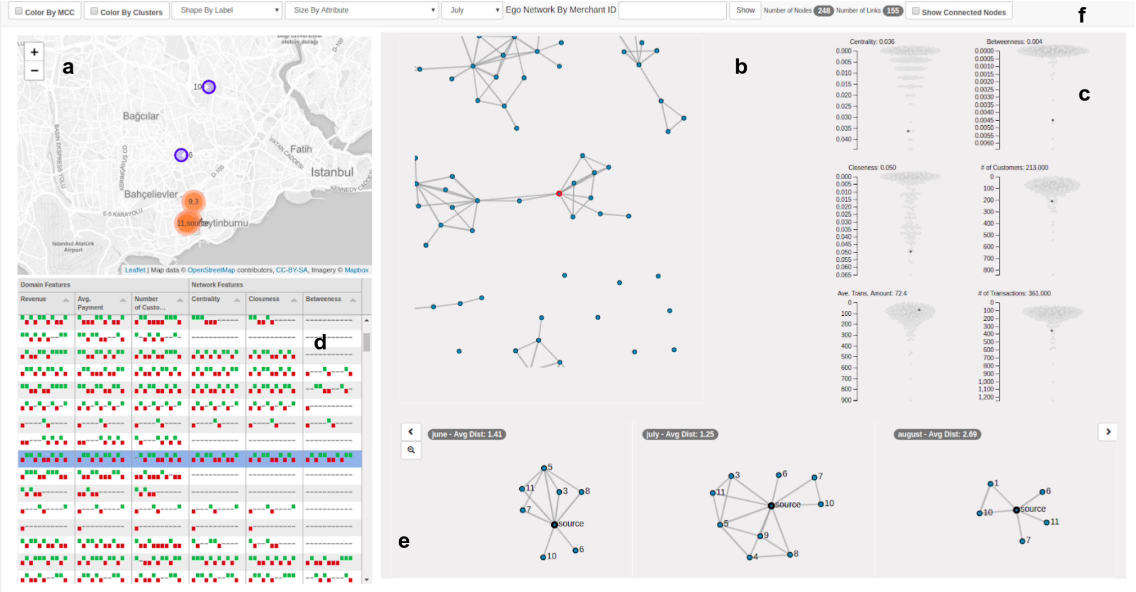


Figure 3.1: The designed user interface. a) The spatial component depicts the encoding between the network and a geographical map. b) The general network overview. c) The distribution component for domain and network features. d) The pixel display component to visualize temporal changes in the domain and network features. e) The ego-network component to display the ego-network evolution. f) Visual encoding and filtering component.

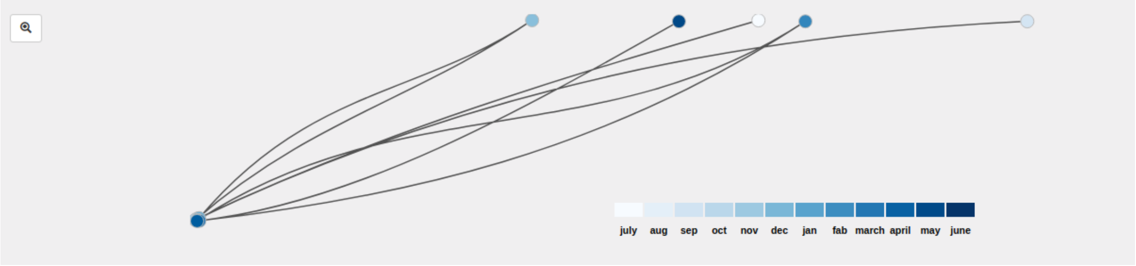


Figure 3.2: Each dot represents an ego-network, U_k^t, of node k at time step t. The distance between two dots serves as the similarity in terms of combined domain and network attributes.

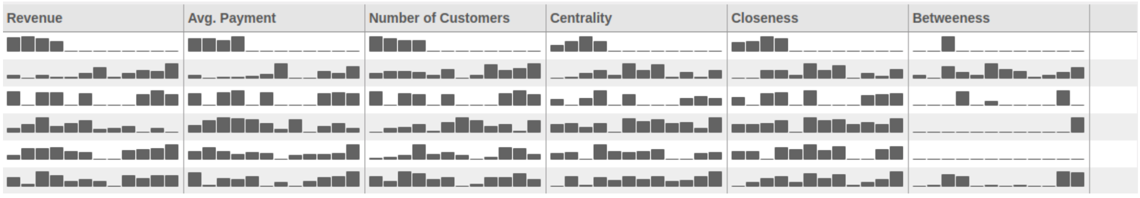


Figure 3.3: The temporal changes in both domain and network attributes for a selected group of nodes in the network.

be available for a particular or a group of node(s) in the network. In addition, the visual display of ego-network evolution should answer mesoscopic and microscopic [96] questions. The mesoscopic questions deal with comparisons among a group of nodes' ego-network evolution similarities. On the other hand, microscopic questions focus on a particular node's ego-network evolution.

In Figure 3.1, the designed interface, consisting of six visual components, can be observed. Each component in the interface serves towards one of the goals as mentioned earlier. The main component of the interface is the temporal network area (part b in the Figure 3.1) in which a snapshot G_t of the multivariate dynamic network is depicted to achieve overview and zoom/filter functions. Although there exists other network layouts, such as matrix notation [43] or 3D networks [41], node-link layout [27, 67] is chosen due to its simplicity and understandability. Next component, part a, displays the spatial attributes of each node on a map. The component on part c displays temporal domain and network attribute distributions with swarm plots [1]. In order to display the temporal increases and decreases in both domain and network attributes, a pixel display [60] is utilized in part d. Lastly, the visual encoding and filtering component, part f, help users assign domain and network attributes to specific visual encodings, for instance, color, shape and size. Besides, it enables users to filter unconnected nodes and select a particular node in the network.

The visual display of ego-network evolution takes place in part e in Figure 3.1. A timeline view [89] in which a node's ego-networks are juxtaposed to each other as small multiples. Moreover, there are two additional components located in the same view. The first one is the spatial layout, in Figure 3.2, a connected scatter plot [91] depicts the similarities between each ego-network U_k^t of node k at time step t. Finally, in order to compare a group of nodes' temporal domain and network attributes, sparklines [40] are utilized as in Figure 3.3 in the temporal attribute component.

3.2.3 Visual Components and Interactions

3.2.3.1 Network Overview Component

Network Overview Component plays a vital role in the developed exploratory tool. Given a snapshot, G_t, of the multivariate dynamic network as a result of the time cutting operation [12], the node-link layout [88] is generated with force-directed velocity Verlet integration [94] which is a frequently utilized method in simulations in order to compute the trajectories and interactions of each particle in the system.

Network Overview Component serves a dual function; overview and zoom/filter [85]. The initial state of the component allows users to assess the general structure of the network. With the help of visual encoding and filtering component and interaction methods to be explained, users can focus on a single or a group of nodes in the network.

Users are able to navigate through the network layout with multiple interactions:

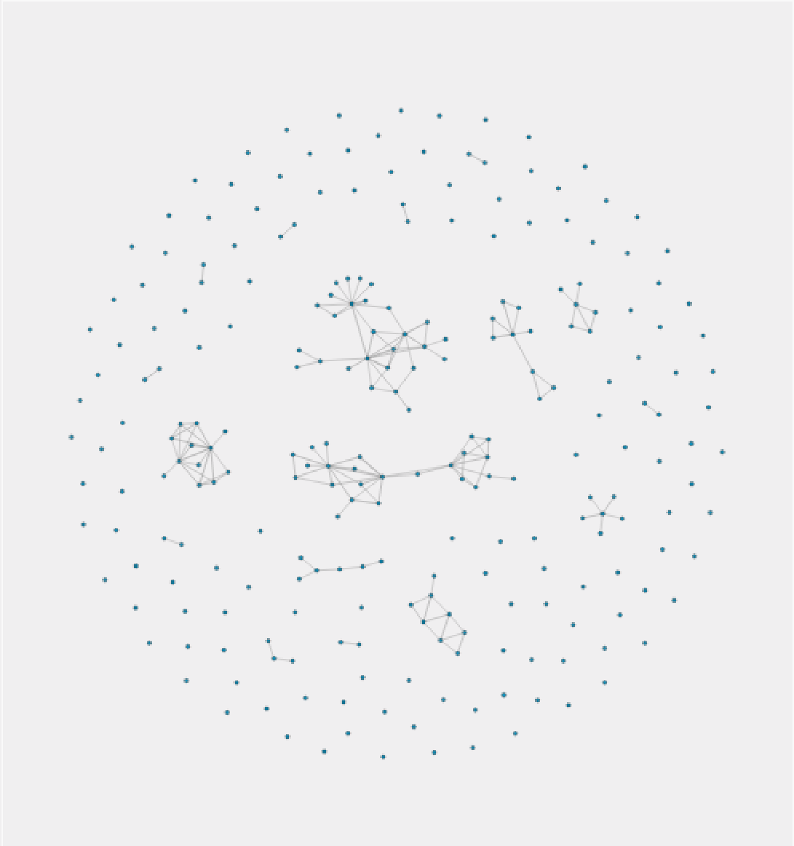


Figure 3.4: The initial view of the network overview component with a node-link layout computed with force-directed velocity verlet integration.

Zoom in/out, pan and drag. Zooming enables users to focus on a particular region of interest in the network by zooming in or have an overall understanding of the entire network by zooming out. Panning allows users to traverse the network within the same zoom level. Lastly, users are able to drag and change the positions of the nodes in the network so that they can re-order the network.

In the network layout, users are able to click on a particular node and activate other components in the network with respect to the clicked node's attribute as in Figure 3.1, the details on activated the components will be examined in the later sections. Besides, the clicked node in the layout is highlighted by an increased stroke width and stroke color, as in Figure 3.5. In order to preserve the consistency, on each click event, the previously clicked node is de-highlighted.

In a complex network layout, the edges, connections between nodes may be difficult to distinguish. In the network overview component, a node's directly connected neighbors are highlighted on hover as in Figure 3.5. In addition, the rest of the nodes and edges are set to a lower opacity value. Moreover, a tooltip is displayed with informative properties of the hovered node.

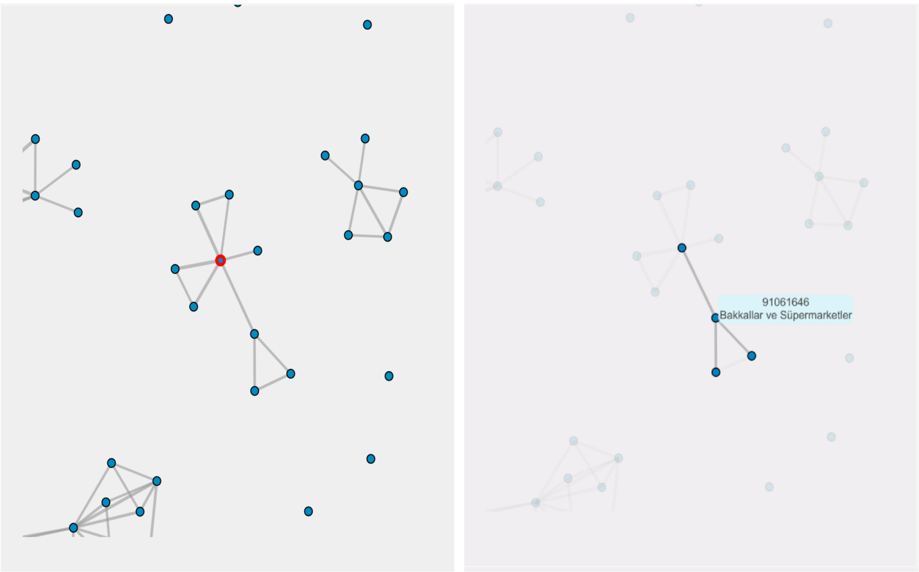


Figure 3.5: The result of click and hover interactions on the network layout. A node's directly connected neighbors are highlighted on hover. In addition, a tooltip is displayed with informative properties.

Setting a lower opacity value for the unrelated nodes and edges enables users to concentrate on the hovered node and its neighbors. In addition, highlighting the directly connected neighbors of the hovered node provides an insight on the prospect ego-network in terms of the number of alters and neighborhood structure..

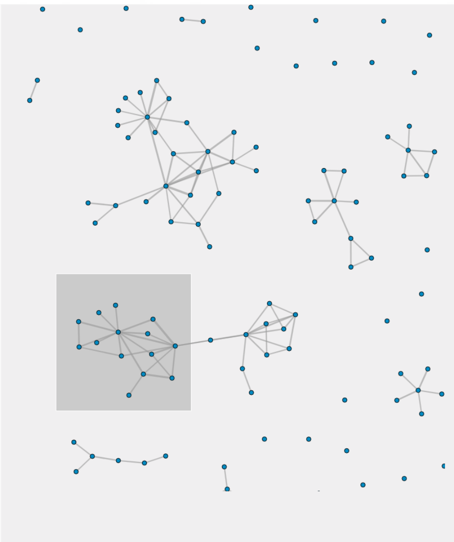


Figure 3.6: Multiple nodes in the network are selected with a rectangular brush. As a result, the temporal domain and network attributes of the selected nodes are displayed in designated views.

To distinguish the nodes stored in the network layout, based on their domain and network attributes, nodes can be encoded with visual variables color, shape and size [25, 82, 92] as in Figure 3.7. Users can change each encoding option from the visual encoding and filtering component, which can be observed in part f in Figure 3.1. As a result, users are able to observe the multivariate attributes of nodes spread across the layout or clustered in a particular region.

Brushing [77, 92] is an effective interaction technique to select and link a group of visual elements across multiple views by creating a selection area of various shapes. In the developed exploratory tool, a rectangular brush is utilized to select a group of nodes in order to display their temporal domain and network attributes. Moreover, users are able to resize and move the drawn rectangular area to another region in the network layout. In order to analyze nodes clustered around a particular region in the network layout, brushing help users observe the patterns and outliers in the clustered nodes; besides, trends in the time domain within the designated view in the exploratory tool which are going to be explained in the following sections.

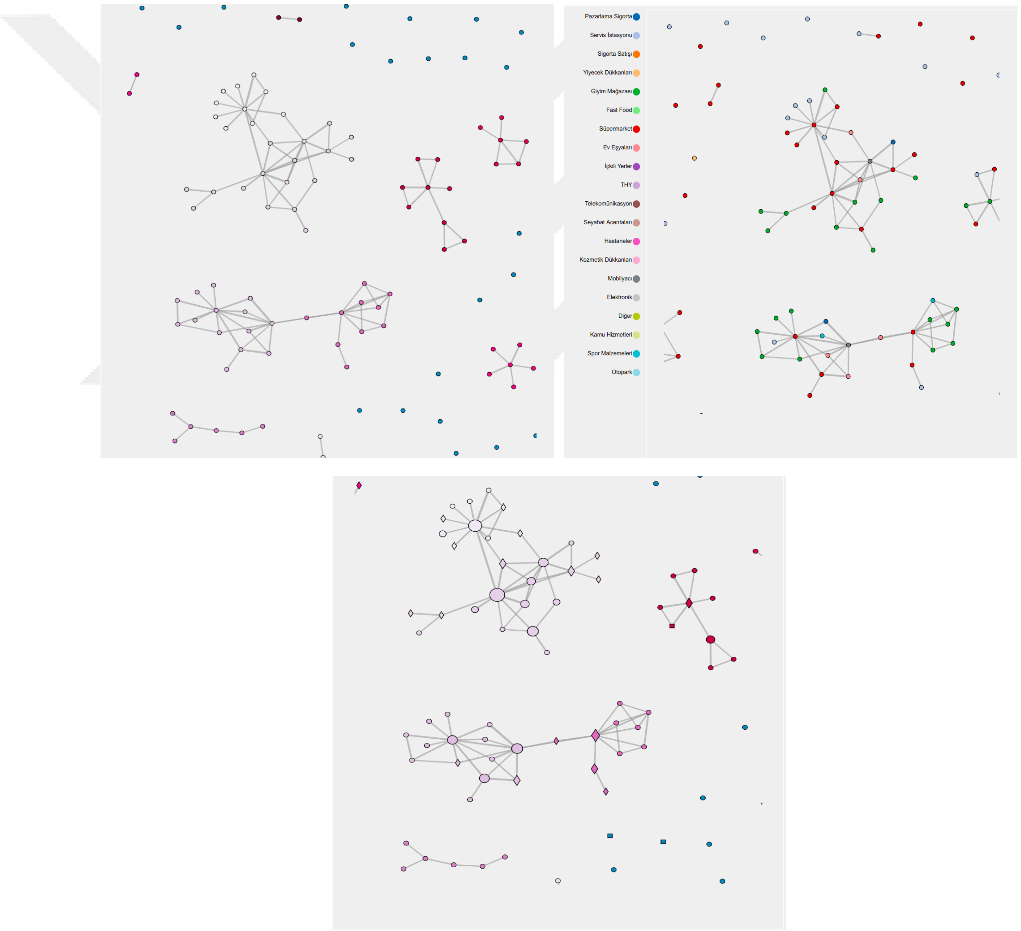


Figure 3.7: On the left, communities in the network layout are utilized to color the nodes. In addition, nodes can be colored with respect to one of their domain attributes as displayed in the figure on the right. Furthermore, shape and size elements are employed to encode the rest of the multivariate attributes.

3.2.3.2 Spatial Encoding Component

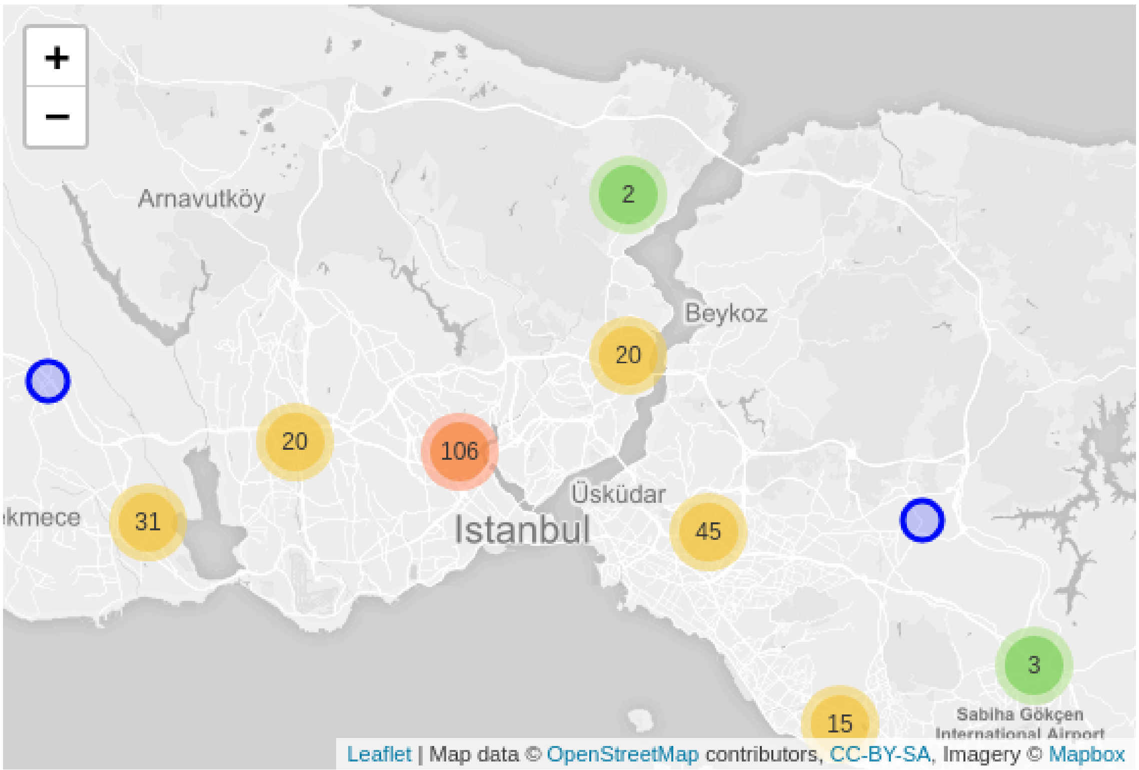

+
−
Arnavutköy
2
Beykoz
20
20
106
Üsküdar
kmece
31
Istanbul
45
3
15
Sabiha Gökçen International Airport
Leaflet | Map data © OpenStreetMap contributors, CC-BY-SA, Imagery © Mapbox

Figure 3.8: A general view from the spatial encoding component. The spatial attributes of the nodes in the network layout are presented in a map.

As stated previously, the developed tool analyzes multivariate dynamic networks with nodes having spatial attributes. Spatial encoding component projects aforementioned spatial attributes on a 2D map. In Figure 3.8, the initial state of the component can be observed. In order to prevent a cluttered view, individual markers are grouped based on their proximities and the number inside group markers state the size of the group with the color encoding as well. With the help of clustering, nodes on the exact location can be separated as well.

With the help of spatial encoding component, users are able to observe the spatial relationship between nodes in the network. In addition to the location information, markers on the map can represent the domain attributes of the nodes with the same encodings in network overview component as in Figure 3.9.

Users are able to interact with the component. On hover over a marker, the associated node will be centered with increased stroke size and color in the network overview component. In addition, domain and network attributes of the hovered node are displayed in the distribution component. The node identification number can be viewed with the help of a tooltip on click. Moreover, users can click on a node in the network overview component; as a result, the ego-network of the clicked node at time step t is displayed on the map with the existing encoding selections.

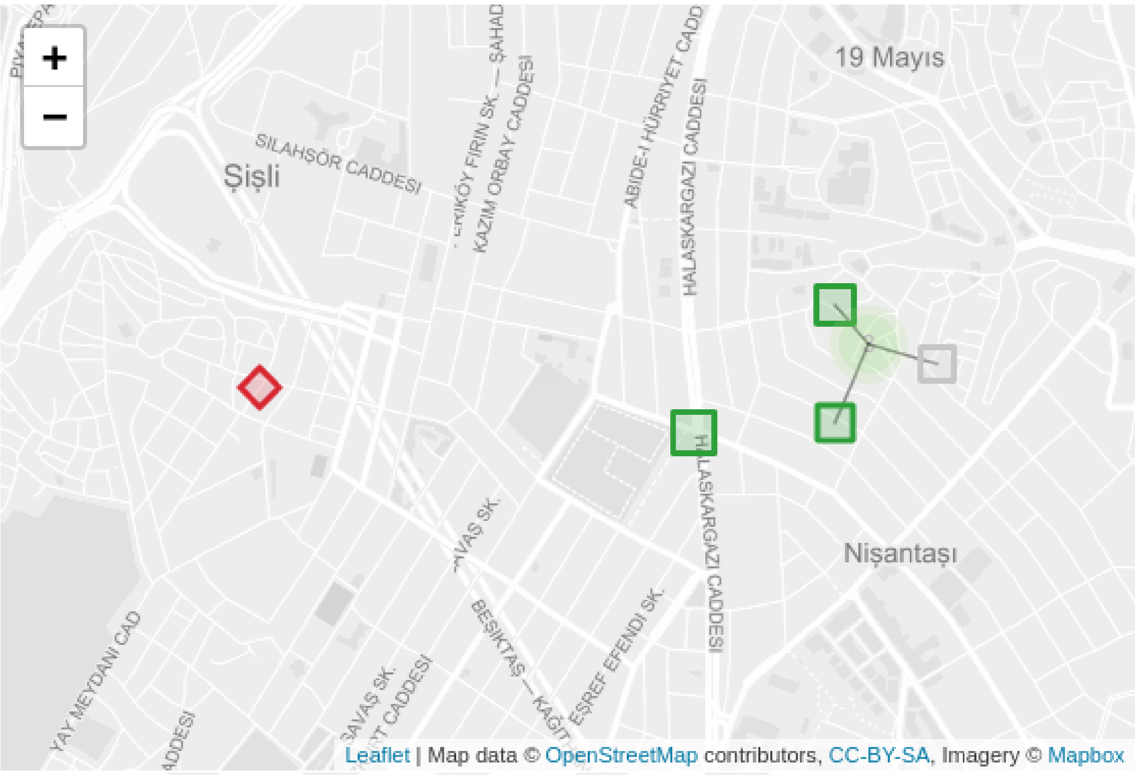


Figure 3.9: Domain attributes can be encoded on the markers. Nodes on the exact location can be separated on click.

Users are able to navigate between consecutive time steps with the help of arrow buttons in the timeline component. In order to compare a group of nodes' locations on the map, users can make use of the brush selection as described in the previous section. The locations of the brushed nodes are displayed with the selected encoding options.

3.2.3.3 Attribute Distribution Component

The distributions of the domain and network attributes are displayed as in Figure 3.10. Instead of traditional distribution displays such as histograms and boxplots, swarm plots are chosen to visualize the distributions. Swarm plots are based on strip plots in which individual data points are plotted along an axis. However, with strip plots, overlapped and cluttered data points may occur and lead to obscured data. Instead of plotting data points merely based on their values, swarm plots utilize a force-directed layout approach to move data points away from each other with minimum distance, causing each data point to be visible on the layout. As a result of visible data points, users have a wide range of interaction methods.

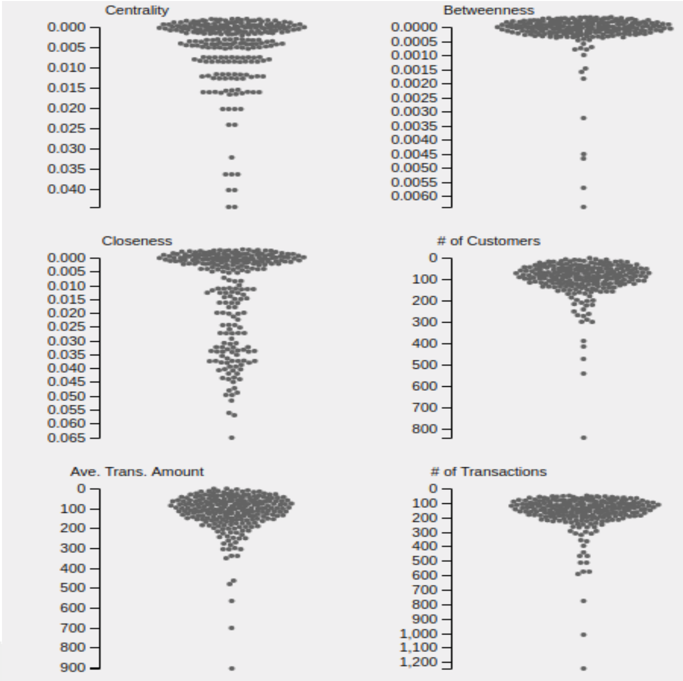


Figure 3.10: Initial view of the attribute distribution component.

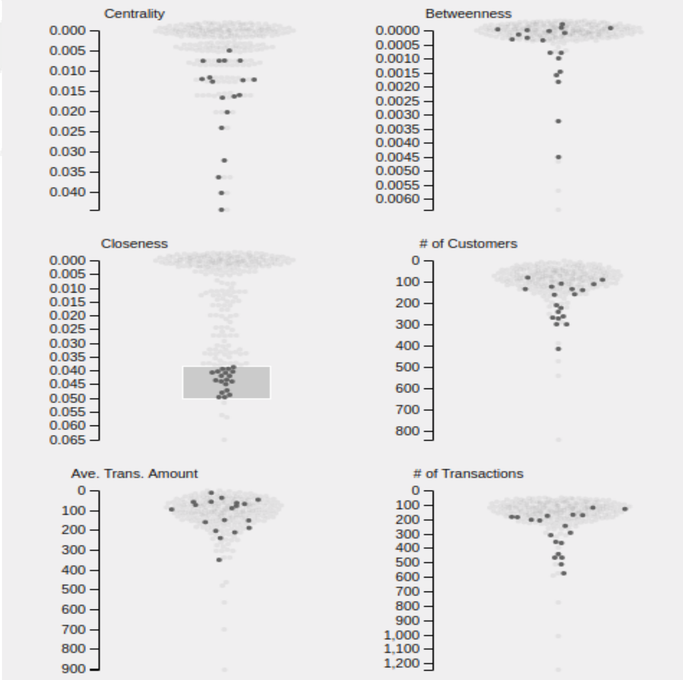


Figure 3.11: Brushing on a particular attribute display would highlight the same data points on other attribute displays as well.

Hover on a data point in one of the attribute displays would activate different components. First, the data point will be highlighted in other attribute displays as well. Next, the associated node in the network is centered and zoomed in the layout. And finally, the associated node is located on the map.

Brushing enables users to observe how a group of data points distribute across domain and network attributes as in Figure 3.11. In addition, associated nodes in the network and spatial layouts are highlighted. As a result, the distributions of the selected data points are observed in topological and spatial dimensions.

3.2.3.4 Pixel Display Component

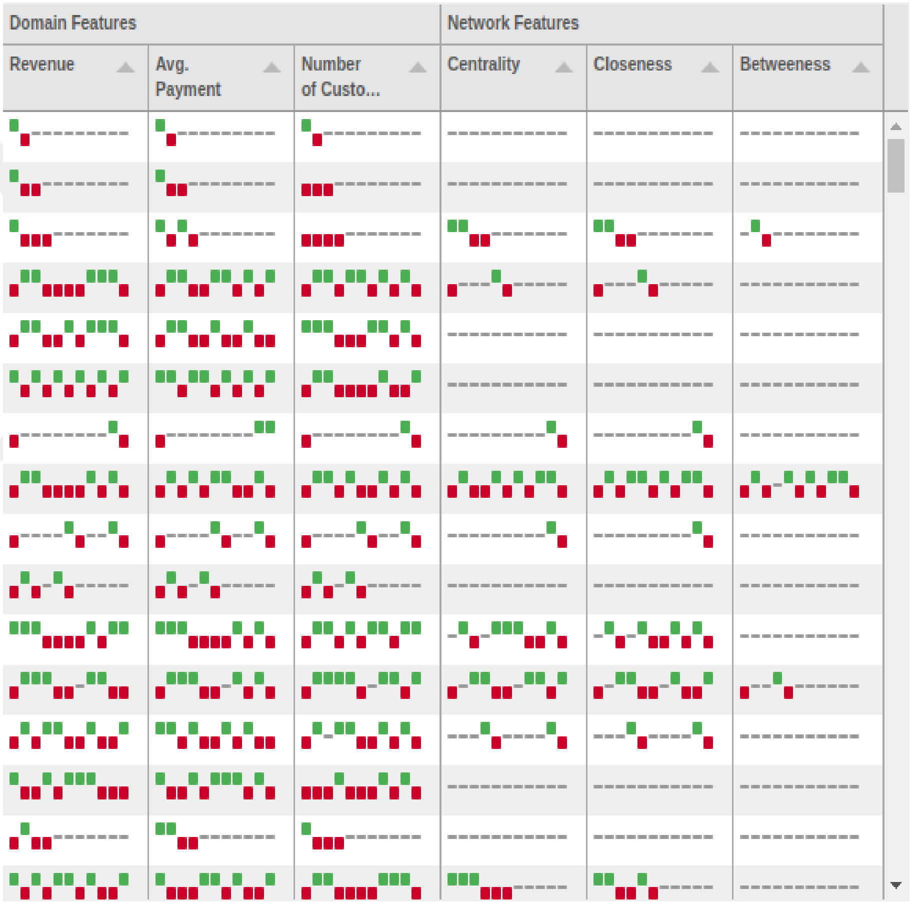


Figure 3.12: Changes between consecutive time steps are displayed in the pixel display component.

Instead of observing raw attribute values for each time step, encoding the changes between consecutive time steps as colored pixel values would reveal an overall pattern as in Figure 3.12. For a time span of size N steps, there are $N-1$ pixels representing the changes between consecutive time step; green-colored pixels state an increase and red-colored pixels state decrease compared to the previous time step. While gray color means no change.

Domain and network attributes of all nodes in the network are formed as a table. If the number of the nodes exceeds the length of the table, users may scroll down to observe further rows representing nodes in the network layout. On click, the associated node is highlighted and centered in the network layout.

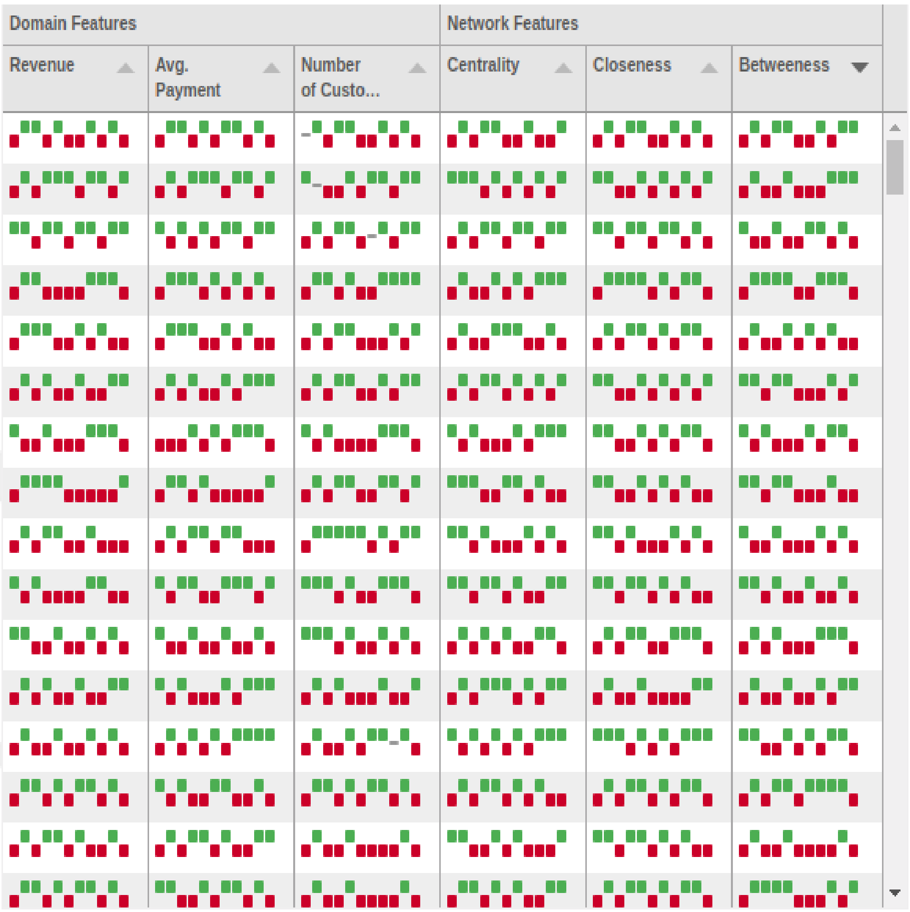


Figure 3.13: Sorted pixel display based on the activity counts.

In order to observe the nodes with the highest or lowest number of changes, users able to sort the nodes based on the number of changed time steps, as in Figure 3.13. Each attribute column contains an arrow symbol to sort rows in ascending or descending order.

3.2.3.5 Ego-Network Component

In order to display the dynamic ego-network evolution of a particular or group of nodes in the network overview layout, a hybrid approach is embraced. Users are able to observe and compare three consecutive time steps juxtaposed within a window in which they can navigate between time steps with the help of buttons placed on the corners. As in Figure 3.14, domain attributes of ego and alter nodes can be encoded on the layout from the visual encoding and filter component. Moreover, in order to identify alters between time steps, each unique alter is assigned to a numeric identification value as in Figure 3.14.

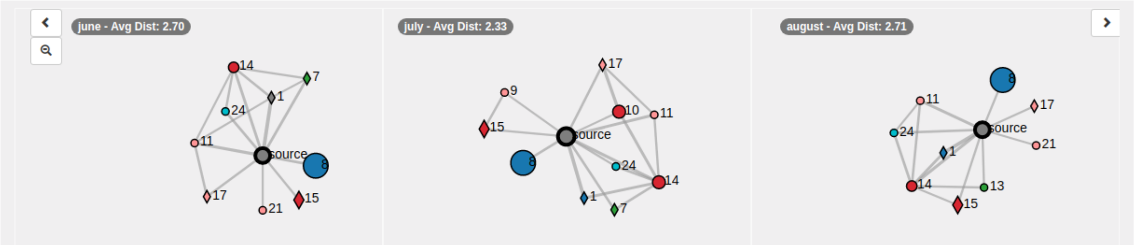


Figure 3.14: Visual encodings of the domain attributes on display.

Furthermore, users are able to compare a group of nodes' domain and network attributes throughout the entire time span as a result of time flattening operation, brushing on network layout would activate the view in Figure 3.3. The evolution of domain and network attributes of the selected nodes are presented as sparklines in a table in which row corresponds to a node. Hover on a row would activate the views in the network overview and attribute distribution components as in Figure 3.15. Hovered node is centered in the network layout and highlighted in the attribute distribution component.

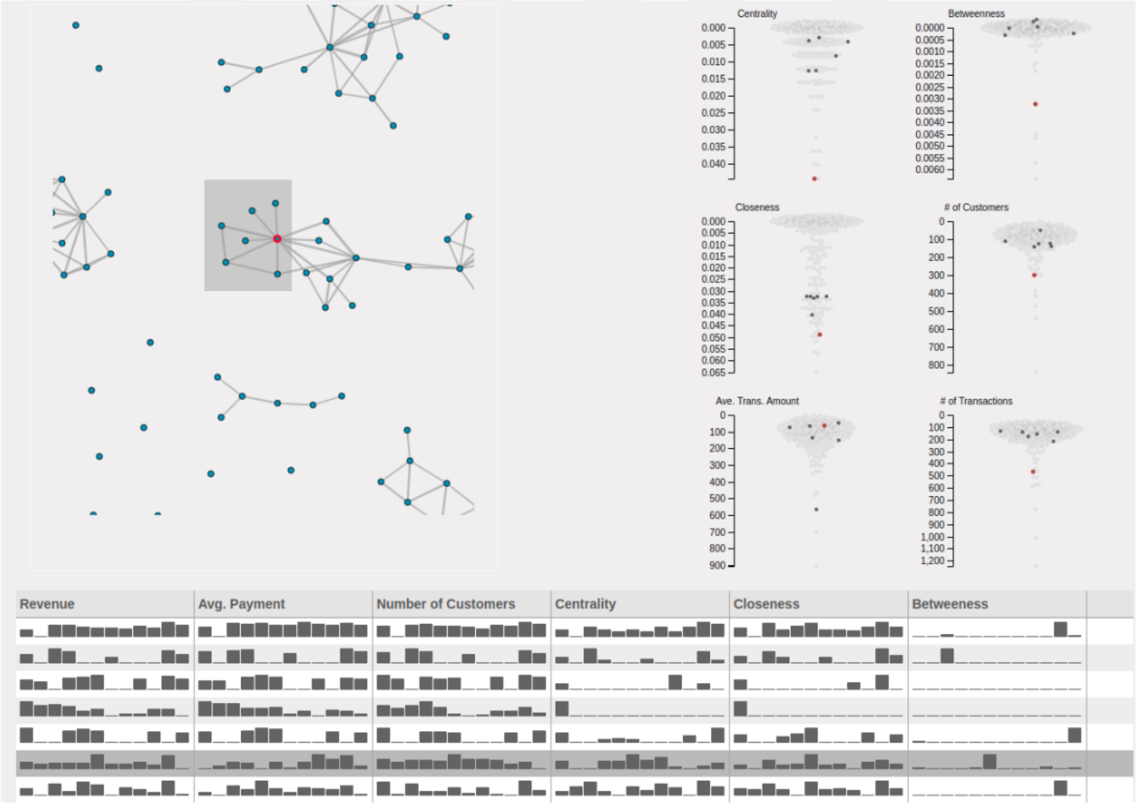


Figure 3.15: Hover on a row in the temporal attribute change area activating other components in the tool.

Within a timeline view, users are able to compare the differences between consecutive time steps. On top of the encoded domain attributes, alters in the ego-networks are assigned with identification numbers. As a result, emerging and dissolving nodes can be detected. However, in order to find distinctions between consecutive time steps, users need to iterate over the timeline, which yields a cognitive load on the user [60]. In order to display the whole evolution in a single view, a time flattening operation is applied and formed a time folding approach [14] as in Figure 3.2. Users can detect the time step where a drift occurs in consecutive time steps, and then with the help of timeline view can further analyze the difference.

Users are able to switch to this view with the help of the zoom-in button on the top left corner. Forming a connected scatter plot, each dot in Figure 3.2 stands for the selected node's ego-network at a particular time step encoded with sequential coloring. Consecutive time steps are connected to each other to depict the transition. The distance between dots, or ego-networks, represents how similar they are. As distance increases between two dots, the dissimilarity between them increases as well. In order to compute the distances, temporal domain and network features are combined into feature vectors for each time step. Moreover, unlike previous studies, locations of all ego-networks belonging to the entire time span are computed in a single projection. Thus, results are comparable within a single view. Network features, which are obtained from Wu et al. [96], can be observed in Table 3.1.

Feature	
Number of Alters	$n_k^t = \lvert V_k^t - \{k\} \rvert$
Number of Edges Between Alters	$L = \lvert E_k^t \rvert - n_k^t$
Average Tie Strength	$avg(w_{e_j})$
Density (Clustering Coefficient)	$\frac{L}{n_k^t(n_k^t-1)/2}$

Table 3.1: The network features used in ego-network feature vectors.

On top of the network features described in Table 3.1, domain attributes are added to the feature vectors as well, in contrast to the previous studies. In order to project the computed temporal feature vectors, dimensionality reduction techniques are utilized. Among the existing studies, Multidimensional Scaling stands out as the frequent choice [14, 60, 96], since it aims to preserve the original distances in the temporal feature space. As a result, metric MDS is utilized in this thesis as well since feature vectors are consisting of quantities measured in terms of coordinates.

MDS expects a dissimilarity matrix as the input to project multidimensional features onto a 2D plane. Distance metric in the dissimilarity matrix plays a crucial role to encapsulate the dissimilarities. The selected distance metric should be sensitive to small changes. Canberra distance, $\sum_{i=1}^{d} \frac{|P_i - Q_i|}{P_i + Q_i}$, is an effective distance metric to discriminate changes near zero [18]. Consequently, MDS is utilized with a distance matrix computed with Canberra distance.

3.2.3.6 Visual encoding and filtering component

With the help of visual encoding and filtering component, users are able to encode domain attributes on the network overview and ego-network components, change the time step in the multivariate dynamic network and enter a node identification number to locate it on the network layout alongside its domain and network attributes on the attribute distribution component.

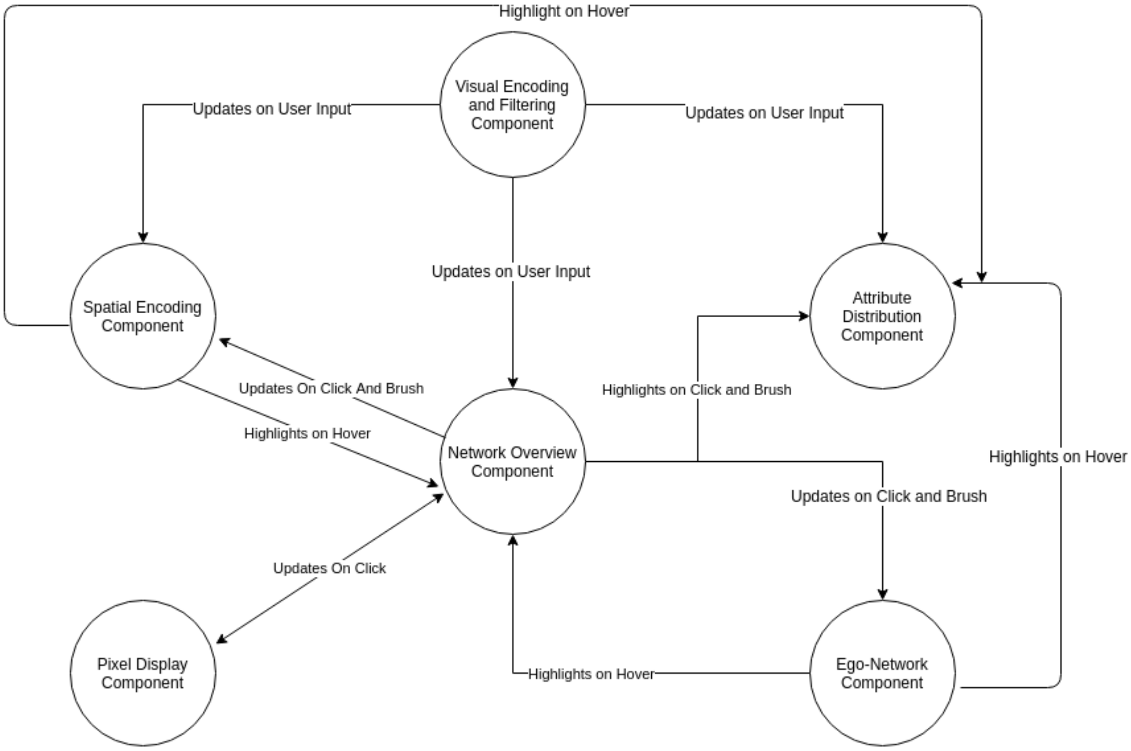

Highlight on Hover
Visual Encoding and Filtering Component
Updates on User Input
Updates on User Input
Updates on User Input
Spatial Encoding Component
Attribute Distribution Component
Updates On Click And Brush
Highlights on Click and Brush
Highlights on Hover
Network Overview Component
Highlights on Hover
Updates on Click and Brush
Updates On Click
Pixel Display Component
Highlights on Hover
Ego-Network Component

Figure 3.16: Interactions between components.

3.3 System Architecture

The developed tool is a client-side web application. Visualizations, interactions and application logic are programmed in Javascript, especially with D3 [20] and JQuery [2] frameworks. Besides, for the interactive map, Leaflet [3] framework is used. Lastly, sparklines that represent temporal attributes in ego-network and pixel display components are implemented with Tabulator [6] framework. For the Louvain community detection algorithm, the JavaScript implementation made by [4] is utilized. In order to reduce the load on the browser, the processing of the raw data is done beforehand. Raw data is processed step with Python, especially with NetworkX [5] library.

3.3.1 Data Model

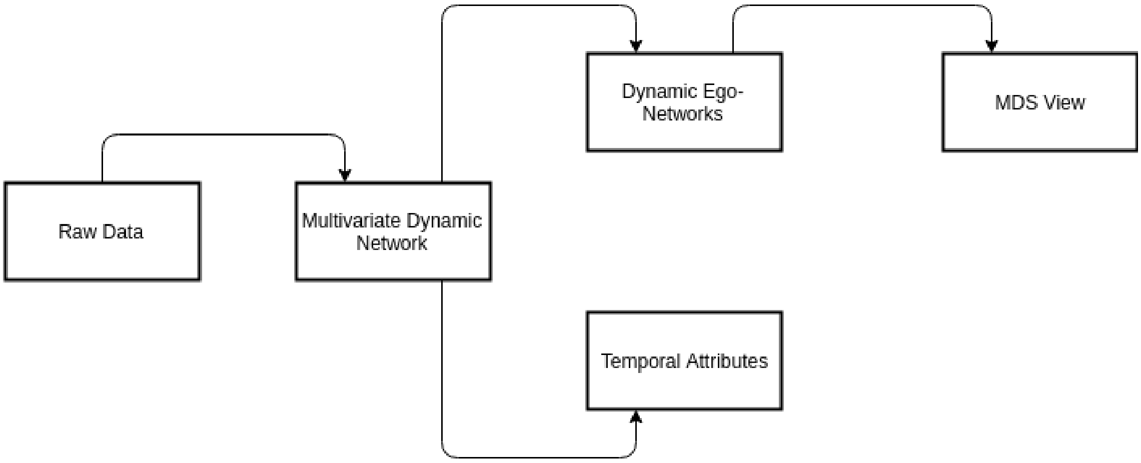


Figure 3.17: Data transformation pipeline.

As in Figure 3.17, Multivariate dynamic network is directly created from the raw data to be displayed in the network overview component. For all nodes in the multivariate dynamic network, their ego-networks are extracted in each time step and stored in a file. Furthermore, temporal domain and network attributes are extracted from the multivariate dynamic network and stored in a file. And lastly, feature vectors are extracted from the ego-networks to compute ego-network coordinates at each time step t with MDS.

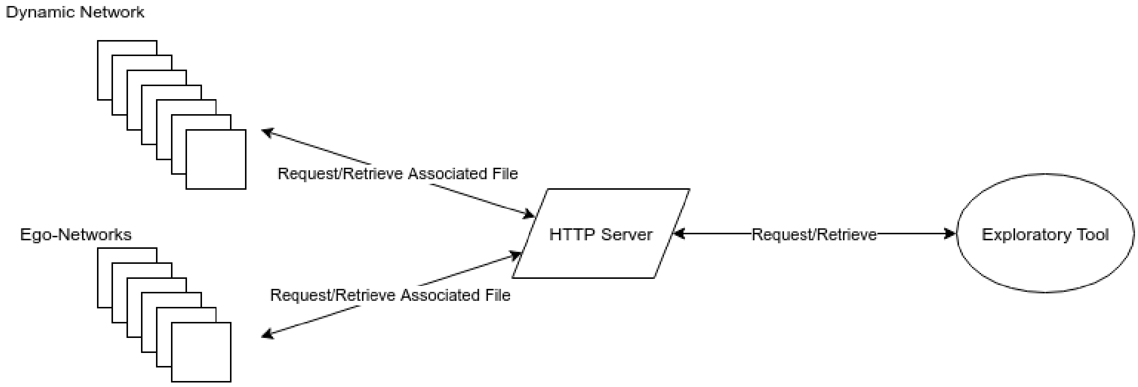
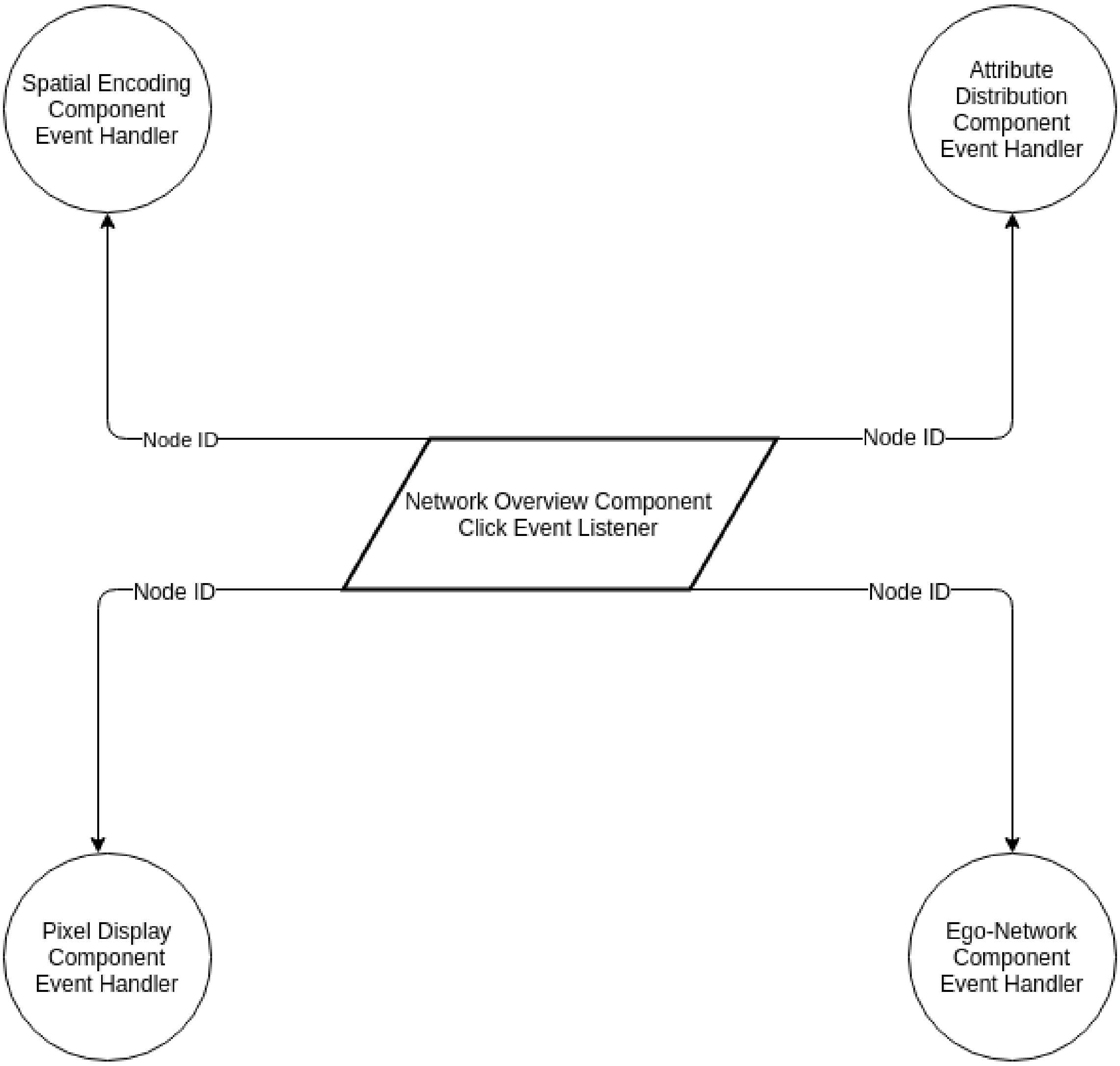


Figure 3.18: Data retrieval in the exploratory tool are carried out with HTTP requests.

Basic HTTP requests retrieve necessary files to update network overview and ego-network components. Whenever the user changes the time step to update network layout or clicks on a node in the layout to view its ego-network, the associated file is retrieved and displayed accordingly as in Figure 3.18.

3.3.2 Communication Between Components



Spatial Encoding Component Event Handler
Attribute Distribution Component Event Handler
Node ID
Node ID
Network Overview Component Click Event Listener
Node ID
Node ID
Pixel Display Component Event Handler
Ego-Network Component Event Handler

Figure 3.19: Propagation of the clicked node's ID between source and target components.

The communication among the components is carried out with an event-based structure. Event listeners in both D3 and JQuery frameworks are utilized to detect click, hover and brush selection in the components. Furthermore, each component in the tool maintains its own event handling schema to take the appropriate action.

In each event, the target node's identification number is obtained from the event attributes and propagated to other components in the tool to be selected or highlighted. For instance, in Figure 3.19, a click on node event is described. The Identification number of the selected node in the network is propagated to associated components. Then, each component updates its view, as stated in the event handlers.

Chapter 4

Use Case: Merchant Network

In this section, a use case scenario of the developed tool will be explained with the help of a real-world data set. In Section 4.1, the data will be analyzed with additional background. And next, in Section 4.2, usage scenarios on the developed tool will be explained.

4.1 Data Overview

4.1.1 Background

In today's world, small and medium-sized enterprises, SMEs, constitute a significant segment in the majority of the economies. For instance, as of 2011, there are about 2 Million SMEs in Turkey accountable for approximately 76 percent of the total employment [1]. As a result, from a bank's perspective, SMEs play a key role in their business plans since most of the SMEs frequently require loans from banks. However, banks must prioritize SMEs based on their financial well-being in order to have a balanced monetary funds allocation. To this end, capable models are required by the banks to predict an SME's future financial well-being.

Existing approaches for predicting an SME's well-being are based on financial criteria, for instance, earnings per asset and equity per asset [32]. However, required evaluation criteria may not be available for all companies, especially for SMEs. In addition, these financial features do not cover the dynamics of an SME within the surrounding environment. In the remaining of the chapter, SME's will be referred to as *merchants*.

A research team in Sabanci University proposed a novel set of attributes solely based on credit card transactions and created a merchant network in which merchants are nodes and shared customers are edges. With the help of created merchant network, a set of network features are extracted and provided to the predictive

[1] `http://www.tuik.gov.tr/HbPrint.do?id=15881`

model. The rationale behind creating such a network is to capture the situation of merchants by incorporating surrounding information.

The same research team utilized the developed tool to explore created merchant networks. Furthermore, with the ego-network component, researchers aimed to observe the changes in selected merchants' ego-networks.

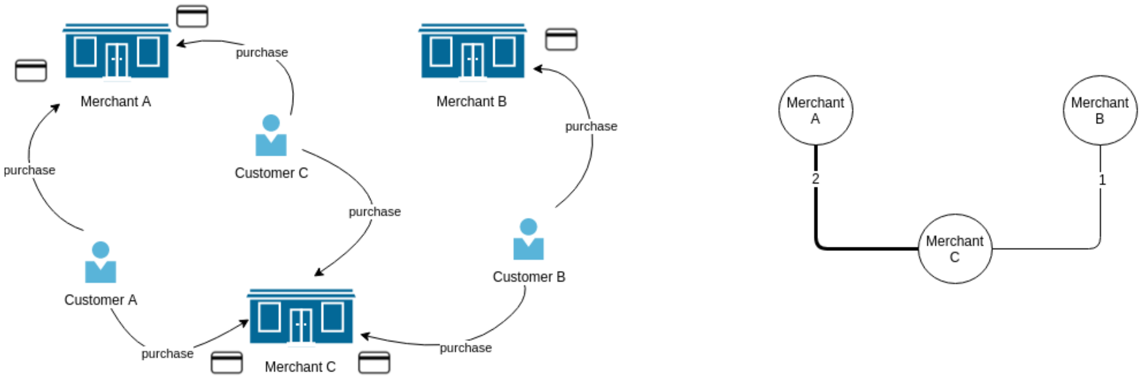


Figure 4.1: An example transaction schema and merchant network generation. On left, two customers make transactions in different merchants. On right, the resulting merchant network is displayed.

4.1.2 Transaction Data Set

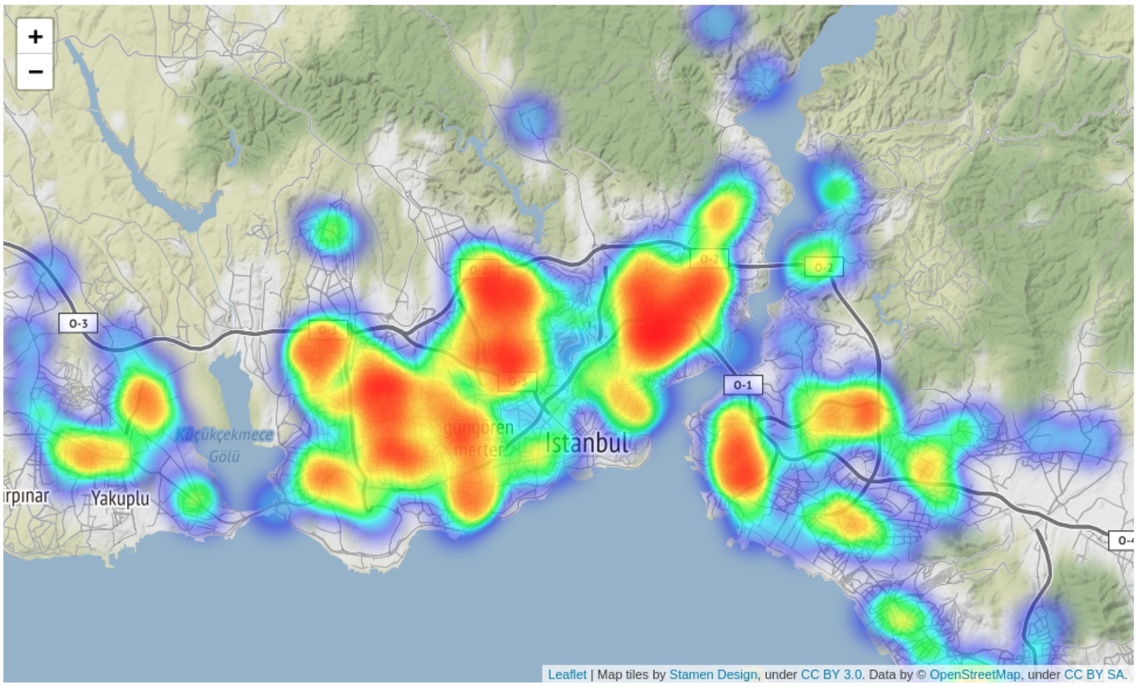

+
−
O-3
O-1
O-2
O-4
Küçükçekmece Gölü
Güngören
Merter
İstanbul
ırpınar
Yakuplu
Leaflet | Map tiles by Stamen Design, under CC BY 3.0. Data by © OpenStreetMap, under CC BY SA.

Figure 4.2: Spatial distribution of the merchants.

The credit card transaction data was provided by a private bank operating in Turkey. In the data set, transactions are made by registered customers in registered merchant venues. The officials of the bank anonymized required fields such as customer and merchant identification values. A snapshot of the obtained transaction data set can be observed in Table 4.1.

MUSTERI_ID_MASK	ISLEM_TARIHI	ISLEM_SAAT	ISLEM_TUTARI	UYEISYERI_ID_MASK	X	Y	mcc_cat
9980121	2014-07-29	02:20:22	293.84	99334438	41.03146	28.98911	Doğrudan Pazarlama- Sigorta Hizmetleri
16768400	2014-07-28	02:15:48	135.00	99334438	41.03146	28.98911	Doğrudan Pazarlama- Sigorta Hizmetleri
4260356	2014-07-28	02:38:33	126.00	99334438	41.03146	28.98911	Doğrudan Pazarlama- Sigorta Hizmetleri
6887140	2014-07-06	02:31:14	135.00	99334438	41.03146	28.98911	Doğrudan Pazarlama- Sigorta Hizmetleri
4956623	2014-07-20	02:12:22	126.00	99334438	41.03146	28.98911	Doğrudan Pazarlama- Sigorta Hizmetleri

Table 4.1: Sample transaction data. Each row in the table corresponds to an expenditure.

Each row in the data set corresponds to a transaction made by a customer. In addition to identification values for both customer and merchant, further information, transaction amount, date and time, merchant category (*mcc* in Table 4.1) and merchant location is stored as well. In total, 2.249.910 transactions are stored within a time span of 12 months from July 2014 to June 2015 with 4.723 merchants in 20 merchant categories spread across Istanbul as in Figure 4.2, although two merchants were located in different places.

In order to enrich the set of merchant attributes, customer demographics data set is provided by the bank as well. The data set contains 20.000 customers' demographics attributes such as age, gender, income and education level. However, the majority of customers do not make frequent transactions on a monthly basis.

4.1.3 Merchant Network

In order to create the multivariate dynamic network, the research team determined to choose month based time steps for the sake of the predictive model. First, transaction data set split into temporal subsets for each time step. For each month, merchants with at least 100 transactions are chosen to be included in the network, which creates a distinct number of nodes for each temporal snapshot of the dynamic network. After the filtering process is done, chosen merchants are represented as nodes in the network with associated financial attributes. Figure 4.5 depicts the number of filtered merchants in each time step, on average 220 merchants with 132 links. With such networks, all interaction results are returned to the users within 500ms.

To assign an edge between two nodes, merchants, in the network, the number of shared customers are computed. As stated earlier, the majority of the customers do not make frequent transactions. In order to prevent cluttered edges and misleading results, edges are set based on a threshold value. As a result, an edge is created between two merchants if the number of shared customers is more than the threshold value. The research team determined to set the threshold value to 15 customers.

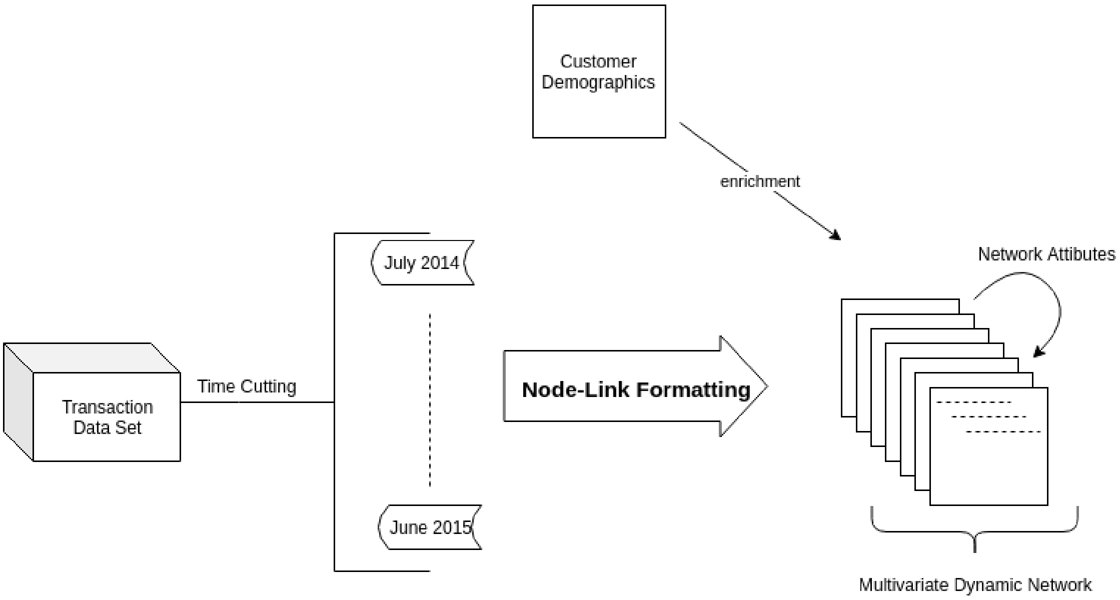
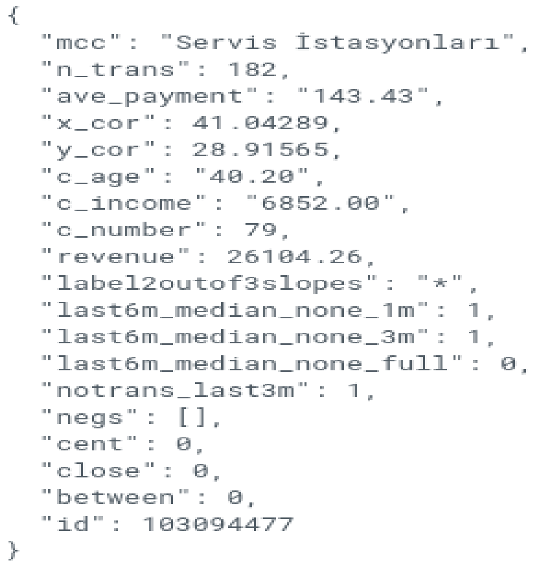


Figure 4.3: Multivariate dynamic network creation from transaction data set.

After the processing step is done, for each node in the temporal network, aggregated customer demographics attributes, which are average age, income and number of customers, are extracted and assigned to associated nodes. In addition, network features are assigned as well, degree centrality, betweenness and closeness measurements for each node. Figure 4.3 depicts the overall process starting from raw transaction data to obtaining the multivariate dynamic network.

In addition, the research team defines their own set of attributes that represent the binary well-being of a merchant in 5 different metrics as in Figure 4.4. The defined attributes play a key role for the research team as a proxy for the well-beings.

Due to filtering operations in the pre-processing phase, many nodes and edges emerge and dissolve between consecutive time steps. Moreover, because of the applied threshold value, resulting temporal networks contain many disconnected nodes in the layout. In Figure 4.5, the sparsity of the resulting temporal networks can be observed in which density is computed as $d = \frac{2m}{n(n-1)}$ where m is the number of edges while n is the number of nodes.



```
{
  "mcc": "Servis İstasyonları",
  "n_trans": 182,
  "ave_payment": "143.43",
  "x_cor": 41.04289,
  "y_cor": 28.91565,
  "c_age": "40.20",
  "c_income": "6852.00",
  "c_number": 79,
  "revenue": 26104.26,
  "label2outof3slopes": "*",
  "last6m_median_none_1m": 1,
  "last6m_median_none_3m": 1,
  "last6m_median_none_full": 0,
  "notrans_last3m": 1,
  "negs": [],
  "cent": 0,
  "close": 0,
  "between": 0,
  "id": 103094477
}
```

Figure 4.4: Resulting node attributes in the form of a node-link file. Well-being attributes start with *last* prefix.

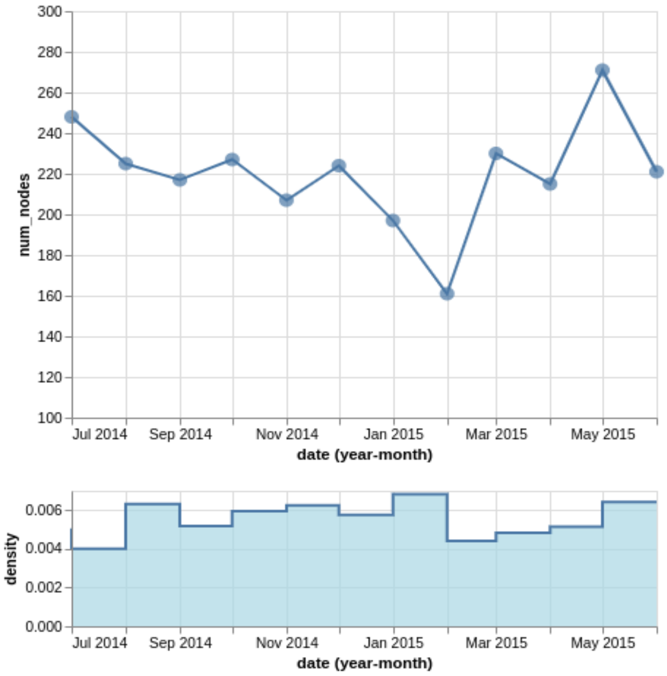


Figure 4.5: Number of nodes in temporal networks along with associated network density values.

4.2 Use Case Scenarios

On network overview and ego-network components, merchant category is encoded by color. Moreover, well-being status is encoded by shape, diamonds representing successful merchant; while circles represent failing ones. In addition, squares stand for unknown states. Size of the nodes encodes several domain attributes such as average transaction amount, number of customers and revenue. And lastly, the width of the links encodes the number of shared customers.

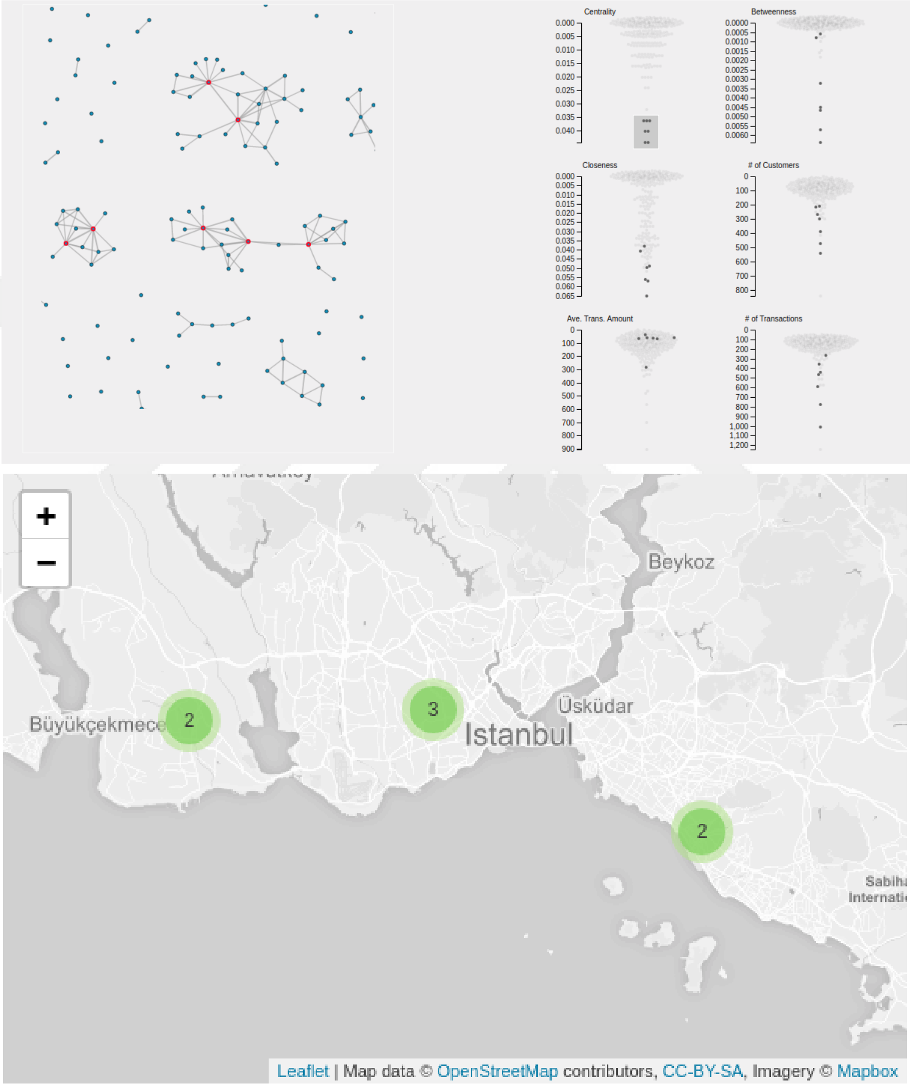


Figure 4.6: Topological and spatial distribution of the top performing merchants in terms of degree centrality.

With the help of the attribute distribution component, users are able to observe how does a group of merchants with similar attribute values are distributed in topological and spatial order. For example, in Figure 4.6, the top-performing merchants in terms of degree centrality are brushed in the attribute distribution component. As a result, selected merchants are displayed in network overview and spatial encoding components. In terms of spatial encoding, there is no relation between them. However, as expected, they are all central merchants in the network layout. In addition, they are all groceries except two of them and share similar attribute values.

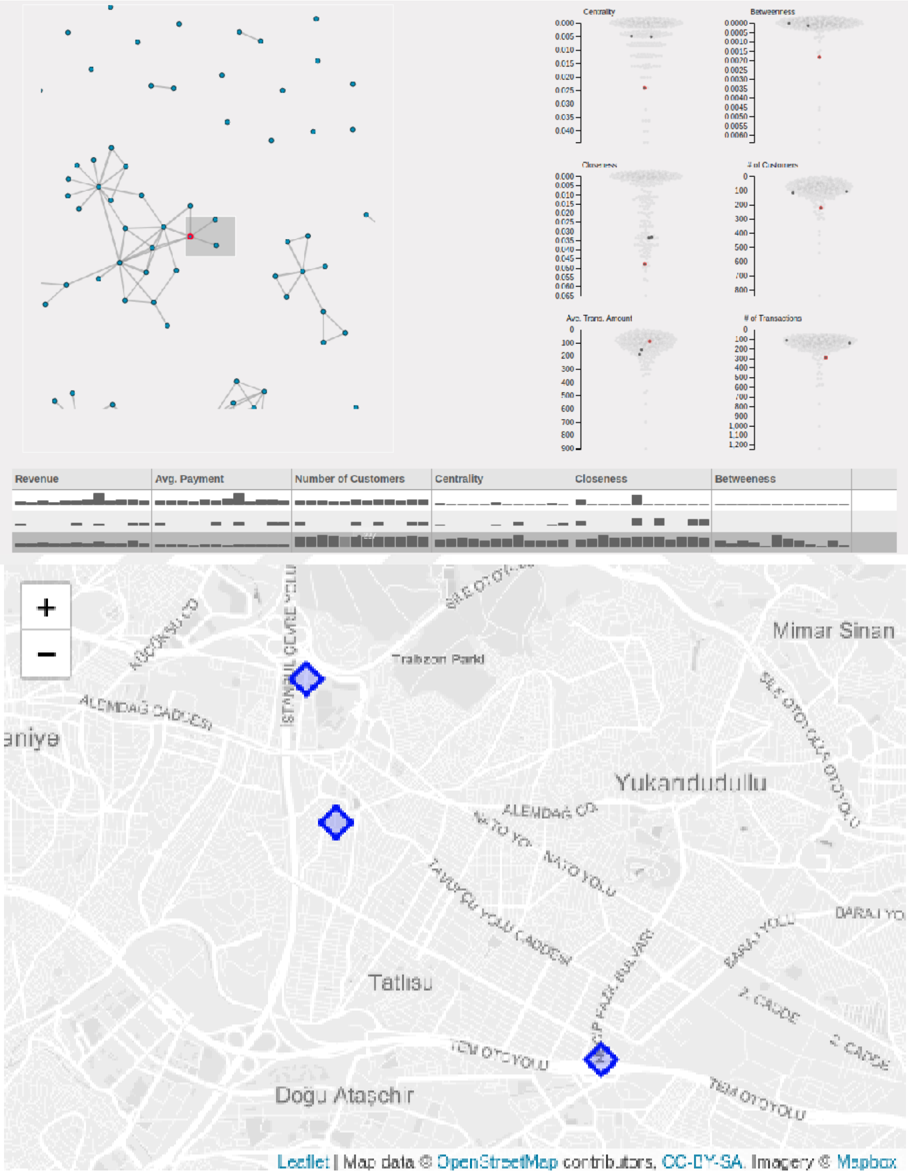


Figure 4.7: A group of successful merchants attribute and spatial distributions.

As merchant well-being represented with shape encoding, merchant groups with the same marker shape are crucial. For instance, in Figure 4.7, three successful merchants are selected.The merchant with red-stroke is connected to remaining merchants. They are located around the same district. Moreover, they share similar values in the domain and network attributes, except degree centrality. This example provides visual evidence towards the hypothesis proposing that well-being is related to the domain and network attributes; in addition, the location may play an import role.

With the help of pixel display component, users may find merchants with increasing domain and network attributes and analyze how they are located in the network overview and attribute distribution components. Locating such merchants on the network layout would reveal more insights as other domain attributes encoded on the layout. For instance, a user may relate increases and decreases in attributes with merchant well-beings.

Domain Feature
Revenue
Average Transaction Amount
Number of Transactions
Number of Customers

Table 4.2: Additional domain attributes used in the merchant ego-network feature vectors.

In order to prepare the feature vectors for ego-networks, the domain attributes in Table 4.2 are combined with the features listed in Table 3.1. As stated in the system explanation, in order to track changes between consecutive time steps, projected feature vectors can be utilized. The distances between dots belonging to consecutive time steps, stand for the dissimilarity in terms of the computed feature vectors.

For instance, in Figure 4.8, a significant distance between July and August can be observed with the help of projected feature vectors. Once the distance is noticed, users can go back to the timeline view and analyze how the difference between feature vectors affects the topology between consecutive time steps. In addition, other attribute encodings can be utilized to observe the changes. In Figure 4.8, a significant change occurs between July and August, as many alters leave the ego-network in the latter time step.

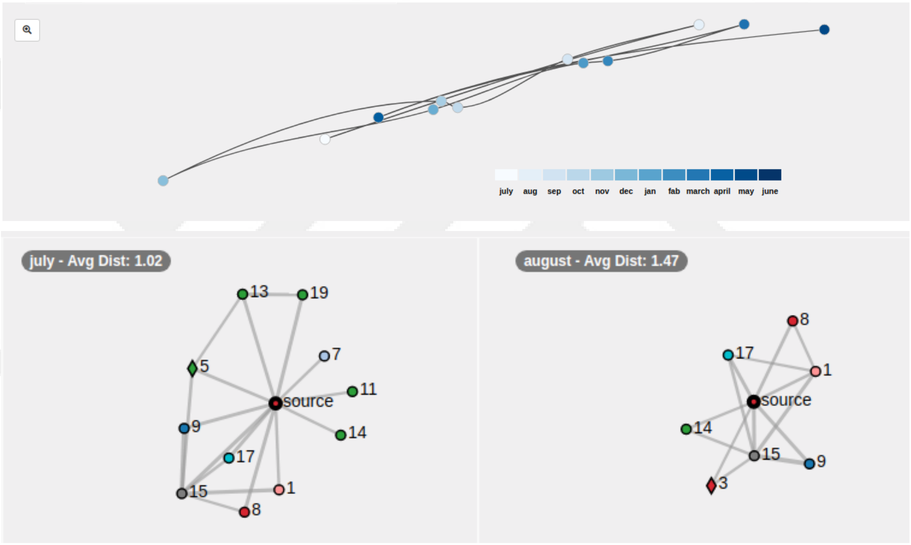


Figure 4.8: A significant distance takes place between July and August. Users can observe the differences in detail with the help of the timeline view.

Chapter 5

Conclusion & Future Work

In this thesis, a visual analytics tool is demonstrated for the exploration of multivariate dynamic networks involving spatial attributes from an egocentric approach. In addition to visualizing multivariate temporal networks from a topological point of view, the developed tool equally depicts the evolution of domain and network attributes in multiple views. With the help of interaction methods as mentioned earlier on integrated components, users are able to observe how the domain and network attributes of a single or a group of nodes in the network changes throughout the entire time span.

On top of depicting raw domain and network attributes, with the help of pixel display component, the changes between consecutive time steps are presented to the user, with an option to sort the nodes in the network based on the activity counts on changed attributes. In consequence, users can observe the changes in attributes within a single view.

In contrast to the existing multivariate network exploration tools, the developed tool enables users to observe the spatial relationship between nodes in the network on the spatial encoding component in which users are able to utilize the same set of visual encodings available to network layout on the 2D map. As a result, the spatial distribution of domain and network attributes are presented to the user.

With the help of the ego-network component, users can trace the changes between domain and network attributes in consecutive time steps, by means of visual encodings and labeled alters in the timeline view. On top of that, a group of nodes' domain and network attributes are displayed to the users within a table of sparklines. As a result, selected nodes can be compared to each other across time steps.

Along with a timeline view, the developed tool presents projections of temporal feature vectors on a 2D view to depict the dissimilarities between ego-network in consecutive time steps belonging to a particular node in the network. Previous studies on displaying ego-network evolution utilize feature vectors as well; however, each time step has its own projection for a group of nodes [96]. As a result, a particular node's ego-networks in different time steps cannot be compared due to separate projections. In this thesis, for a selected node, temporal feature vectors are projected within a single frame, which enables users to compare consecutive time steps.

As a use case scenario, a real-world data set is utilized. A merchant network from credit card transactions is created and provided to the developed tool. Users are able to observe monthly created networks with visual encodings depicting financial and network attributes. Moreover, users can examine how financial attributes are distributed in spatial and topological layers with the help of network overview and spatial encoding components. Lastly, the ego-network component allows users to analyze a particular or a group of merchants' evolution along the temporal axis.

In this thesis, there exists a single use case on the merchant network. However, the developed tool must be evaluated with more use cases involving larger networks. Moreover, user studies with detailed questionnaires must be carried out on a group of participants, and received responses must be analyzed to further evaluate the performance of the tool.

The developed tool utilizes a time cutting operation on the dynamic networks in order to depict each temporal network. However, users need to switch between time steps to observe the changes between them. More sophisticated methods based on time flattening operations can be developed to depict temporal networks.

The pre-processing step plays a crucial role in the developed tool. After the required temporal and ego networks are prepared, they are stored in files. However, the existence of external files causes extensive I/O operations that create an extra load on the developed tool. A graph database such as Neo4j [1] can be utilized to alleviate the load on the tool and further improve the filtering via graph queries.

The ego-network component presents a node's evolution with a time folding approach using its combined domain and network attributes as a feature vector. Users are able to observe the dissimilarities between ego-networks in different time steps based on the distance. However, the current implementation does not provide interpretable views. Users can observe how distant they are but can not grasp the reason behind it. In order to increase the interpretability, the tool may allow users to create their own feature vectors on the fly and observe the changes between them.

With the help of derived feature vectors, users utilize distances between consecutive time steps and further analyze the difference in the timeline view. However, in order to compare far-off time steps, users need to go back and forth on the timeline, which yields a significant cognitive load. In order to prevent this issue, the time folding view can be enriched with additional visual elements that depict the detailed view of associated ego-networks.

[1] `https://neo4j.com/`

The projected feature vectors can be utilized to compare a group of nodes' ego-network evolution. As a result of the projected feature vectors, patterns, trends and outliers among the selected nodes can be identified. However, specialized layouts must be designed to prevent a potential cluttered view.

The depicted distances between temporal feature vectors play a vital role in a node's ego-network evolution. However, the applied distance metric significantly affects a user's perception of distance. In this thesis, Canberra distance is utilized because of its ability to handle changes near zero. However, further distance metrics must be evaluated.

Bibliography

[1] Beeswarm boxplot (and plotting it with r). `https://www.r-statistics.com/2011/03/beeswarm-boxplot-and-plotting-it-with-r/`. Accessed: 2019-06-20.

[2] Jquery. `https://jquery.com/`. Accessed: 2019-06-20.

[3] Leaflet - a javascript library for interactive maps. `https://leafletjs.com/`. Accessed: 2019-06-20.

[4] Louvain community detection algorithm. `https://github.com/upphiminn/jLouvain`. Accessed: 2019-06-20.

[5] Networkx: Software for complex networks. `https://networkx.github.io/`. Accessed: 2019-06-20.

[6] Tabulator. `http://tabulator.info/`. Accessed: 2019-06-20.

[7] Almendra, V., and Roman, B. Using exploratory data analysis for fraud elicitation through supervised learning. In *2011 13th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing* (Sept. 2011), IEEE.

[8] Alspaugh, S., Zokaei, N., Liu, A., Jin, C., and Hearst, M. A. Futzing and moseying: Interviews with professional data analysts on exploration practices. *IEEE Transactions on Visualization and Computer Graphics 25*, 1 (Jan. 2019), 22–31.

[9] Antemijczuk, P., Magiera, M., Lehmann, S., Cuttone, A., and Eg Larsen, J. Visualizing multi-channel networks. *22nd International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, WSCG 2014, Poster Papers Proceedings - in co-operation with EUROGRAPHICS Association* (2014), 5–8.

[10] Archambault, D., Purchase, H., and Pinaud, B. Animation, small multiples, and the effect of mental map preservation in dynamic graphs. *IEEE*

Transactions on Visualization and Computer Graphics 17, 4 (April 2011), 539–552.

[11] Auber, D., Chiricota, Y., Jourdan, F., and Melancon, G. Multiscale visualization of small world networks. In *IEEE Symposium on Information Visualization 2003 (IEEE Cat. No.03TH8714)* (2003), IEEE.

[12] Bach, B., Dragicevic, P., Archambault, D., Hurter, C., and Carpendale, S. A review of temporal data visualizations based on space-time cube operations, 2014.

[13] Bach, B., Pietriga, E., and Fekete, J.-D. GraphDiaries: Animated transitions andTemporal navigation for dynamic networks. *IEEE Transactions on Visualization and Computer Graphics 20*, 5 (May 2014), 740–754.

[14] Bach, B., Shi, C., Heulot, N., Madhyastha, T., Grabowski, T., and Dragicevic, P. Time curves: Folding time to visualize patterns of temporal evolution in data. *IEEE Transactions on Visualization and Computer Graphics 22*, 1 (Jan. 2016), 559–568.

[15] Battle, L., Chang, R., and Stonebraker, M. Dynamic prefetching of data tiles for interactive visualization. In *Proceedings of the 2016 International Conference on Management of Data - SIGMOD '16* (2016), ACM Press.

[16] Battle, L., and Heer, J. Characterizing Exploratory Visual Analysis: A Literature Review and Evaluation of Analytic Provenance in Tableau. *Computer Graphics Forum 38*, 3 (2019), 145–159.

[17] Beck, F., Burch, M., Diehl, S., and Weiskopf, D. The state of the art in visualizing dynamic graphs, 2014.

[18] Berlingerio, M., Koutra, D., Eliassi-Rad, T., and Faloutsos, C. Netsimile: A scalable approach to size-independent network similarity. *CoRR abs/1209.2684* (2012).

[19] Borg, I., Groenen, P. J., and Mair, P. *Applied Multidimensional Scaling.* Springer Publishing Company, Incorporated, 2012.

[20] Bostock, M., Ogievetsky, V., and Heer, J. D3: Data-driven documents. *IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis)* (2011).

[21] Boyandin, I., Bertini, E., and Lalanne, D. A qualitative study on the exploration of temporal changes in flow maps with animation and small-multiples. *Comput. Graph. Forum 31*, 3pt2 (June 2012), 1005–1014.

[22] Brandes, U., and Wagner, D. A bayesian paradigm for dynamic graph layout. In *Graph Drawing*. Springer Berlin Heidelberg, 1997, pp. 236–247.

[23] Burch, M., Vehlow, C., Beck, F., Diehl, S., and Weiskopf, D. Parallel edge splatting for scalable dynamic graph visualization. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (Dec. 2011), 2344–2353.

[24] Caballero, H. S. G., Westenberg, M. A., Gebre, B., and van Wijk, J. J. V-Awake: A Visual Analytics Approach for Correcting Sleep Predictions from Deep Learning Models. *Computer Graphics Forum 38*, 3 (2019), 001–012.

[25] Cava, R., Freitas, C. M. D. S., and Winckler, M. ClusterVis: Visualizing nodes attributes in multivariate graphs. In *Proceedings of the Symposium on Applied Computing - SAC '17* (2017), ACM Press.

[26] Costa, A., Nalon, R., Jr., W. M., and Veloso, A. Ego-centric analysis of supportive networks. In *Proceedings of the 10th ACM Conference on Web Science - WebSci '18* (2018), ACM Press.

[27] Dudas, P. M. Cooperative, dynamic twitter parsing and visualization for dark network analysis. In *2013 IEEE 2nd Network Science Workshop (NSW)* (Apr. 2013), IEEE.

[28] Durcinoska, I., Chung, K. S. K., Young, J., and Solomon, M. J. Social networks and healthcare coordination. In *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017 - ASONAM '17* (2017), ACM Press.

[29] Dwyer, T., and Eades, P. Visualising a fund manager flow graph with columns and worms. In *Proceedings Sixth International Conference on Information Visualisation* (July 2002), pp. 147–152.

[30] ElTayeby, O., and Dou, W. A survey on interaction log analysis for evaluating exploratory visualizations. In *Proceedings of the Beyond Time and Errors on Novel Evaluation Methods for Visualization - BELIV '16* (2016), ACM Press.

[31] Erten, C., Kobourov, S. G., Le, V., and Navabi, A. Simultaneous graph drawing: Layout algorithms and visualization schemes. In *Graph Drawing* (Berlin, Heidelberg, 2004), G. Liotta, Ed., Springer Berlin Heidelberg, pp. 437–449.

[32] Fantazzini, D., and Figini, S. Random survival forests models for sme credit risk measurement. *Methodology and computing in applied probability 11*, 1 (2009), 29–45.

[33] Fares, J., and Chung, K. S. K. Personal networks and perception of care. In *Proceedings of the 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining* (Piscataway, NJ, USA, 2016), ASONAM '16, IEEE Press, pp. 1181–1188.

[34] Farrugia, M., Hurley, N., and Quigley, A. A.: Exploring temporal ego networks using small multiples and tree-ring layouts. In *In Proceedings of the 4th International Conference on Advances in Computer-Human Interactions, ACHI* (2011).

[35] Filipov, V. A., Arleo, A., Federico, P., and Miksch, S. CV3: Visual Exploration, Assessment, and Comparison of CVs. *Computer Graphics Forum 38*, 3 (2019), 107–118.

[36] Friedrich, C., and Eades, P. Graph drawing in motion. *Journal of Graph Algorithms and Applications 6*, 3 (2002), 353–370.

[37] Frishman, Y., and Tal, A. Dynamic drawing of clustered graphs. In *IEEE Symposium on Information Visualization*, IEEE.

[38] Frishman, Y., and Tal, A. Online dynamic graph drawing. *IEEE Transactions on Visualization and Computer Graphics 14*, 4 (July 2008), 727–740.

[39] Ghani, S., and Elmqvist, N. Improving revisitation in graphs through static spatial features. In *Proceedings of Graphics Interface 2011* (School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 2011), GI '11, Canadian Human-Computer Communications Society, pp. 175–182.

[40] Goffin, P., Boy, J., Willett, W., and Isenberg, P. An exploratory study of word-scale graphics in data-rich text documents. *IEEE Transactions on Visualization and Computer Graphics 23*, 10 (Oct. 2017), 2275–2287.

[41] Göhnert, T., Ziebarth, S., Detjen, H., Hecking, T., and Hoppe, H. U. 3d DynNetVis. In *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015 - ASONAM '15* (2015), ACM Press.

[42] Gotz, D., and Zhou, M. X. Characterizing users' visual analytic activity for insight provenance. In *2008 IEEE Symposium on Visual Analytics Science and Technology* (2008), IEEE.

[43] Gove, R., Gramsky, N., Kirby, R., Sefer, E., Sopan, A., Dunne, C., Shneiderman, B., and Taieb-Maimon, M. Netvisia: Heat map & matrix visualization of dynamic social network statistics & content. In *2011 IEEE Third Int'l Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third Int'l Conference on Social Computing* (Oct. 2011), IEEE.

[44] Greilich, M., Burch, M., and Diehl, S. Visualizing the evolution of compound digraphs with timearctrees. *Computer Graphics Forum 28*, 3 (2009), 975–982.

[45] Guo, D. Flow mapping and multivariate visualization of large spatial interaction data. *IEEE Transactions on Visualization and Computer Graphics 15*, 6 (Nov 2009), 1041–1048.

[46] Guo, H., Gomez, S. R., Ziemkiewicz, C., and Laidlaw, D. H. A case study using visualization interaction logs and insight metrics to understand how analysts arrive at insights. *IEEE Transactions on Visualization and Computer Graphics 22*, 1 (Jan. 2016), 51–60.

[47] Hadlak, S., Schulz, H., and Schumann, H. In situ exploration of large dynamic networks. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (Dec 2011), 2334–2343.

[48] Halgin, D., and Borgatti, S. An introduction to personal network analysis and tie churn statistics using e-net. *Connections 32* (01 2012), 36–48.

[49] He, Q., Zhu, M., Lu, B., Liu, H., and Shen, Q. MENA: Visual analysis of multivariate egocentric network evolution. In *2016 International Conference on Virtual Reality and Visualization (ICVRV)* (Sept. 2016), IEEE.

[50] Heer, J., and Boyd, D. Vizster: visualizing online social networks. In *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.* (Oct 2005), pp. 32–39.

[51] Heer, J., and Shneiderman, B. Interactive dynamics for visual analysis. *Communications of the ACM 55*, 4 (Apr. 2012), 45.

[52] Hemmati, A., and Chung, K. S. K. Social networks and quality of life: The national health interview survey. In *Proceedings of the 2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining* (Piscataway, NJ, USA, 2014), ASONAM '14, IEEE Press, pp. 572–579.

[53] HUANG, M. L., EADES, P., and WANG, J. On-line animated visualization of huge graphs using a modified spring algorithm. *Journal of Visual Languages & Computing 9*, 6 (dec 1998), 623–645.

[54] Jankun-Kelly, T., liu Ma, K., and Gertz, M. A model and framework for visualization exploration. *IEEE Transactions on Visualization and Computer Graphics 13*, 2 (Mar. 2007), 357–369.

[55] Kakar, T., Qin, X., Rundensteiner, E. A., Harrison, L., Sahoo, S. K., and De, S. DIVA: Exploration and Validation of Hypothesized Drug-Drug Interactions. *Computer Graphics Forum 38*, 3 (2019), 095–106.

[56] Kalinin, A., Cetintemel, U., and Zdonik, S. Interactive data exploration using semantic windows. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data - SIGMOD '14* (2014), ACM Press.

[57] Keim, D. A. Visual exploration of large data sets. *Communications of the ACM 44*, 8 (Aug. 2001), 38–44.

[58] Kerzner, E., Lex, A., Sigulinsky, C., Urness, T., Jones, B., Marc, R., and Meyer, M. Graffinity: Visualizing connectivity in large graphs. *Computer Graphics Forum 36*, 3 (jun 2017), 251–260.

[59] Lam, H., Russell, D., Tang, D., and Munzner, T. Session viewer: Visual exploratory analysis of web session logs. In *2007 IEEE Symposium on Visual Analytics Science and Technology* (Oct. 2007), IEEE.

[60] Law, P. M., Wu, Y., and Basole, R. C. Segue: Overviewing evolution patterns of egocentric networks by interactive construction of spatial layouts. *CoRR abs/1811.04895* (2018).

[61] Lee, Y.-Y., Lin, C.-C., and Yen, H.-C. Mental map preserving graph drawing using simulated annealing. In *Proceedings of the 2006 Asia-Pacific Symposium on Information Visualisation - Volume 60* (Darlinghurst, Australia, Australia, 2006), APVis '06, Australian Computer Society, Inc., pp. 179–188.

[62] Li, Q., Shen, Q., Ming, Y., Xu, P., Wang, Y., Ma, X., and Qu, H. A visual analytics approach for understanding egocentric intimacy network evolution and impact propagation in MMORPGs. In *2017 IEEE Pacific Visualization Symposium (PacificVis)* (Apr. 2017), IEEE.

[63] Liu, Q., Hu, Y., Shi, L., Mu, X., Zhang, Y., and Tang, J. EgoNetCloud: Event-based egocentric dynamic network visualization. In *2015 IEEE Conference on Visual Analytics Science and Technology (VAST)* (Oct. 2015), IEEE.

[64] Liu, Z., and Heer, J. The effects of interactive latency on exploratory visual analysis. *IEEE Transactions on Visualization and Computer Graphics 20*, 12 (Dec. 2014), 2122–2131.

[65] Liu, Z., Navathe, S. B., and Stasko, J. T. Network-based visual analysis of tabular data. In *2011 IEEE Conference on Visual Analytics Science and Technology (VAST)* (Oct 2011), pp. 41–50.

[66] Lucia, W., and Ferrari, E. EgoCentric. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management - CIKM '14* (2014), ACM Press.

[67] Meidiana, A., and Hong, S.-H. MultiStory: Visual analytics of dynamic multi-relational networks. In *2015 IEEE Pacific Visualization Symposium (PacificVis)* (Apr. 2015), IEEE.

[68] Misue, K., Eades, P., Lai, W., and Sugiyama, K. Layout adjustment and the mental map. *Journal of Visual Languages & Computing 6*, 2 (June 1995), 183–210.

[69] Naquin, D., d'Aubenton Carafa, Y., Thermes, C., and Silvain, M. Circus: a package for circos display of structural genome variations from paired-end and mate-pair sequencing data. *BMC Bioinformatics 15*, 1 (Jun 2014), 198.

[70] Neagae, I., Faur, D., Vaduva, C., and Datcu, M. Exploratory visual analysis of multispectral EO images based on DNN. In *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium* (July 2018), IEEE.

[71] Nobre, C., Meyer, M., Streit, M., and Lex, A. The State of the Art in Visualizing Multivariate Networks. *Computer Graphics Forum 38*, 3 (2019), 807–832.

[72] Nobre, C., Streit, M., and Lex, A. Juniper: A tree+table approach to multivariate graph visualization. *IEEE Transactions on Visualization and Computer Graphics 25* (2018), 544–554.

[73] Perer, A., and Shneiderman, B. Systematic yet flexible discovery. In *Proceedings of the 13th international conference on Intelligent user interfaces - IUI '08* (2008), ACM Press.

[74] Promann, M., and Brunswicker, S. The effect of proximity in social data charts on perceived unity. In *2018 IEEE Conference on Visual Analytics Science and Technology (VAST)* (Oct 2018).

[75] Reda, K., Johnson, A. E., Papka, M. E., and Leigh, J. Modeling and evaluating user behavior in exploratory visual analysis. *Information Visualization 15*, 4 (July 2016), 325–339.

[76] Reitz, F., Pohl, M., and Diehl, S. Focused animation of dynamic compound graphs. In *2009 13th International Conference Information Visualisation* (jul 2009), IEEE.

[77] Roberts, J., and Wright, M. Towards ubiquitous brushing for information visualization. In *Tenth International Conference on Information Visualisation (IV'06)*, IEEE.

[78] Rufiange, S., J. McGuffin, M., and P. Fuhrman, C. TreeMatrix: A hybrid visualization of compound graphs. *Computer Graphics Forum 31*, 1 (feb 2012), 89–101.

[79] Rufiange, S., and McGuffin, M. J. Diffani: Visualizing dynamic graphs with a hybrid of difference maps and animation. *IEEE Transactions on Visualization and Computer Graphics 19*, 12 (Dec. 2013), 2556–2565.

[80] Sallaberry, A., Muelder, C., and Ma, K.-L. Clustering, visualizing, and navigating for large dynamic graphs. In *Graph Drawing* (Berlin, Heidelberg, 2013), W. Didimo and M. Patrignani, Eds., Springer Berlin Heidelberg, pp. 487–498.

[81] Schneider, E., and Dineen, C. Adding a dimension to anscombe's quartet: Open source, 3-d data visualization. In *Proceedings of the 76th ASIS&T Annual Meeting: Beyond the Cloud: Rethinking Information Boundaries* (Silver Springs, MD, USA, 2013), ASIST '13, American Society for Information Science, pp. 96:1–96:3.

[82] Shamir, A., and Stolpnik, A. Applications of geometry processing: Interactive visual queries for multivariate graphs exploration. *Comput. Graph. 36*, 4 (June 2012), 257–264.

[83] Shi, L., Wang, C., and Wen, Z. Dynamic network visualization in 1.5d. In *2011 IEEE Pacific Visualization Symposium* (Mar. 2011), IEEE.

[84] Shi, L., Wang, C., Wen, Z., Qu, H., Lin, C., and Liao, Q. 1.5d egocentric dynamic network visualization. *IEEE Transactions on Visualization and Computer Graphics 21*, 5 (May 2015), 624–637.

[85] Shneiderman, B. The eyes have it: A task by data type taxonomy for information visualizations. In *Proceedings of the 1996 IEEE Symposium on Visual Languages* (Washington, DC, USA, 1996), VL '96, IEEE Computer Society, pp. 336–.

[86] Shneiderman, B., and Aris, A. Network visualization by semantic substrates. *IEEE Transactions on Visualization and Computer Graphics 12*, 5 (Sept. 2006), 733–740.

[87] Siddiqui, T., Kim, A., Lee, J., Karahalios, K., and Parameswaran, A. Effortless data exploration with zenvisage. *Proceedings of the VLDB Endowment 10*, 4 (Nov. 2016), 457–468.

[88] Tominski, C., Abello, J., and Schumann, H. Technical section: Cgv-an interactive graph visualization system. *Comput. Graph. 33*, 6 (Dec. 2009), 660–678.

[89] Tufte, E. *Envisioning Information.* Graphics Press, Cheshire, CT, USA, 1990.

[90] TVERSKY, B., MORRISON, J. B., and BETRANCOURT, M. Animation: can it facilitate? *International Journal of Human-Computer Studies 57*, 4 (2002), 247 – 262.

[91] van den Elzen, S., Holten, D., Blaas, J., and van Wijk, J. J. Reducing snapshots to points: A visual analytics approach to dynamic network exploration. *IEEE Transactions on Visualization and Computer Graphics 22*, 1 (Jan. 2016), 1–10.

[92] van den Elzen, S., and van Wijk, J. J. Multivariate network exploration and presentation: From detail to overview via selections and aggregations. *IEEE Transactions on Visualization and Computer Graphics 20*, 12 (Dec. 2014), 2310–2319.

[93] van der Maaten, L., and Hinton, G. Visualizing data using t-SNE. *Journal of Machine Learning Research 9* (2008), 2579–2605.

[94] Verlet, L. Computer "experiments" on classical fluids. i. thermodynamical properties of lennard-jones molecules. *Phys. Rev. 159* (Jul 1967), 98–103.

[95] Wattenberg, M. Visual exploration of multivariate graphs. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2006), CHI '06, ACM, pp. 811–819.

[96] Wu, Y., Pitipornvivat, N., Zhao, J., Yang, S., Huang, G., and Qu, H. egoSlider: Visual analysis of egocentric network evolution. *IEEE Transactions on Visualization and Computer Graphics 22*, 1 (Jan. 2016), 260–269.

[97] Yu, C. Research of time series air quality data based on exploratory data analysis and representation. In *2016 Fifth International Conference on Agro-Geoinformatics (Agro-Geoinformatics)* (July 2016), IEEE.

[98] Zgraggen, E., Zhao, Z., Zeleznik, R., and Kraska, T. Investigating the effect of the multiple comparisons problem in visual analysis. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18* (2018), ACM Press.

[99] Zhao, J., Glueck, M., Chevalier, F., Wu, Y., and Khan, A. Egocentric analysis of dynamic networks with EgoLines. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems - CHI '16* (2016), ACM Press.

[100] Zhao, X., Cui, W., Wu, Y., Zhang, H., Qu, H., and Zhang, D. Oui! Outlier Interpretation on Multi-dimensional Data via Visual Analytics. *Computer Graphics Forum 38*, 3 (2019), 213–224.