NetEvViz: Extending NodeXL for Dynamic Network Visualization

Udayan Khurana, Viet-An Nguyen, Hsueh-Chien Cheng, Xi (Stephen) Chen
Department of Computer Science
University of Maryland, College Park
{udayan, vietan, cheng, xichen}@cs.umd.edu

ABSTRACT
We present NetEvViz, a visualization tool for analysis and exploration of dynamic and static network data. While a few tools for the visual analysis of social networks exist, none provide features for visualization of dynamically changing networks featuring the addition or deletion of nodes or edges. Our tool is a Microsoft Excel 2007 and 2010 plug-in and extends the codebase of NodeXL, a popular network visualization software. The key features of this work are (1) The ability of the user to specify and edit temporal annotations to the network components in an Excel sheet, (2) Visually see the dynamics of the network with multiple graph metrics plotted over the time span of the graph, called Timeline, and (3) Temporal exploration of the network layout using different colors through a dynamic Timeslider. The objective of the new features presented in this paper are to let the data analysts, computer scientists and others to visually observe the dynamics or evolution in a network in an interactive manner.

INTRODUCTION
The visualization of network structures has gained much attention recently. The analysis of large scale complex networks formed by social media has become an interesting topic with great potential in terms of research and business. Through a combination of computers and the great pattern recognition and cognition capability of human visual system, visualization has been considered to be one of the most powerful approaches to the analysis of networks.

Dynamic networks are extensions of conventional networks with temporal attributes attached to the vertices and edges. Since the structure of a dynamic network evolves over time, interesting questions related to temporal attributes can be formed. However the analysis of a dynamic network can be challenging without proper handling of the temporal attribute. When the network is simply showed as it is at a certain timestamp, the relations between the previous and the next timestamp are diminished. Even if the network visualizations are showed side-by-side for comparison, the differences can still be difficult to find.

Of the many tools developed to perform network visualization, NodeXL is a highly recognized tool based on Microsoft Excel which supports flexible manipulation of the visualization. However, there exists only limited ways to analyze dynamic network in the latest version of NodeXL. In order to let users of NodeXL visually analyze the dynamic nature of networks, more advanced and easily usable features need to be included in the tool.

We present NetEvViz, a system prototype to visualize the changes in dynamic networks. The prototype is an extension of NodeXL. Two temporal attributes, start time and end time, represent the time each node or an edge entered or left the network. An example in a social network where each vertex corresponds to a specific user and each edge corresponds to the relation between users, the start and end times of a vertex are the time points when the user joins or leaves the social network on creating or deleting the profile, respectively. The start and end times of an edge can be used to represent the time when the user joins or leaves the social network on creating or deleting the profile, respectively.

Our prototype aids the visualization of the change in a temporal network by highlighting the state of the graph around two chosen timestamps. Given the two timestamps, which are specified by the user, the vertices and edges are categorized into groups with respect to their temporal attributes. Each group is assigned a different color which enables the user to recognize subtle patterns otherwise hard to observe in a homogenous graph. In order to select the points of interest, the user is presented with the summary plots of certain metrics over the timespan of the graph.

RELATED WORK
Dynamic Network Visualization
Visualization is an important and useful approach to analyze network data. Network visualization techniques have been proposed in various domains including social networks [3], biological networks [11], and computer networks [1]. With the recent popularity and availability of temporal or longitudinal network data, there has been an increasing interest in developing visualization techniques for dynamic networks.
In general, these visualization techniques use two popular approaches to represent the network data.

The first approach uses the traditional node-link representation to visualize the network. To add the temporal information, the authors of [2] proposed a 3D layered visualization design in which each newly introduced part of the network is shown in a layer of its own and the composition of consecutive layers represents the corresponding state of the network. [9] studied two types of visualizations of temporal social networks: (1) static flip books (a combination of fixed node layout and dynamic social relationships), where the node positions remain constant but the edges accumulate over time, and (2) dynamic movies where nodes move as a function of changes in relations. The authors of [12] proposed five principles for implementation of temporal visualizations for social networks and presented two network prototypes NodeXL and TempoVis.

The second approach represents a network by an adjacent matrix. TimeMatrix [14] proposed a matrix-based visualization for temporal network data in which each cell in the matrix contains a bar chart glyph to show how a certain attribute of the corresponding nodes or edges change over time.

In addition, a hybrid approach which combines both node-link-based and matrix-based representations have also been used to visualize static networks such as MatrixExplorer [6], MatLink [4] and NodeTrix [5]. However, to the best of our knowledge there is no hybrid approach based tool built for temporal network visualization.

Topological Network Statistics

Besides showing the node-link-based layout of the network, our visualization attempts to incorporate a set of network topology metrics [8] to show quantitatively how the network changes over time. This can be done by plotting a set of network topological measurements at different time points. Network topology has been extensively studied on a wide range of networks including social, biological and computer networks. A variety of topological metrics have been proposed to measure important different types of network properties.

Broadly, the network topology measurements can be divided into three groups from lower to higher abstraction level: local, group (or community) and global metrics. Note that this categorization of metrics is not mutually exclusive. Many metrics at lower levels can be aggregated (e.g., sum, average etc) over multiple nodes to achieve values or distributions at higher levels. Local metrics measure the topological properties of individual nodes in the network. This group of metrics include different centrality measures such as degree, betweenness, closeness, eigenvector and PageRank centralities. In addition, clustering coefficient and assortativity coefficients are a measurement of how nodes locally connect to each other. Group metrics measure the topological properties of a group or community of nodes in the network which include modularity [10] and group evolution metric [11]. Global metrics quantify different properties of the whole network such as number of nodes, number of edges, density etc.

Besides measurements for static networks at different points in time, a number of metrics have been proposed to explicitly quantify the change of network over time. [7] introduced a novel dynamic centrality metric which measures how well connected a node is over time. [11] proposed two metrics to quantify how groups of nodes in the network evolve over time.

NodeXL

NodeXL is a free and open source tool for Network Analysis and Visualization [3] [13]. The tool provides a rich set of functionality for visual analysis of static networks. In the section on design and implementation, we describe in more detail the features of NodeXL and why it is a suitable candidate for extension towards dynamic network visualization.

KEY DESIGNS AND IMPLEMENTATIONS

Introduction to NodeXL

As mentioned before, NodeXL provides many capabilities for Social Network Analysis and Visualization within Excel. Visualizations can be done using different algorithms like Fruchterman-Reingold, Harel-Koren etc. Using select and drag, the user can choose to manually change the layout that was computed algorithmically. Also, the network can be annotated with the names or images of the nodes or edges and the edges can be filtered based upon a numerical attribute associated with that edge. Different nodes and can be colored according to a cluster specification or individually as well. The tool also contains algorithms for calculating network metrics like degree centrality, betweenness centrality, clustering coefficient, eigenvector centrality etc.

Ease of data storage and manipulation with Microsoft Excel, combined with the rich network visualization feature set of NodeXL make the choice of extending NodeXL instead of writing software from scratch a natural one. As explained later, we were also able to utilize Excel’s API for plotting the Timeline. Finally, since NodeXL is an open source software, our future goals of distributing these changes to a large audience of interest are better pursued through this choice.

In order to understand the functionalities provided by NetEvViz, we require the reader to have a working knowledge of NodeXL. For the readers unfamiliar with it, we provide a very brief account of NodeXL in the following sentences and strongly recommend going through the 40-page tutorial that can be found at [13] and optionally refer to [3] as well. In order to use NodeXL in Microsoft Excel 2007 or 2011, the user needs to install the required plug-in and bring up a NodeXL workbook template as shown below. Two worksheets, Vertices and Edges need to be populated in order to specify the input network. The format is quite simple and can be populated entirely manually or can be imported in several formats like Pajek or be imported directly from a Twitter account. Upon data import, the user can calculate various metrics, compute clusters or layout the graph on the pane shown in the right side of the excel plug-in. The NodeXL ribbon
provides various functionalities that can be used to refine the visualization and aid in the network analysis process.

**Implementation with NodeXL**

In order to visualize the temporal aspects of a network, we were required to make a list of changes and add new features to the NodeXL codebase. To start with, the start and end time of a node are essential inputs to the tool for mapping the temporal dimension of data. We introduced two new columns - *Start Time* and *End Time* to the Vertices and Edges worksheets, which specify the time at which a node or an edge joined and the time after which it was no longer a member of the network, respectively. In many networks, a node just the network and stays for ever, where as, in others it can leave the network or break a relationship at a point in time. Hence, we consider it mandatory to enter the Start Time for temporal analysis, but the absence of an end time is understood to be the current time by default. Note that the absence of a timestamp in any of the rows does not affect any existing functionality and hence backwards compatibility is guaranteed. This means that all of NodeXL’s features can still be performed with NetEvViz in the same way. Currently, we have put these two columns as the first two of the user defined columns tentatively, and would propose this to be a part of the main stream worksheet template for NodeXL. For this version of our tool, we have designed two main features - *Timeline*, to view the dynamics of the network using a change in graph metrics, and a *Timeslider for Temporal Comparison*. Accordingly, we created a tab in the NodeXL ribbon. These can be seen in Figure 1. The details of the features follow in the next two subsections.

![Figure 1. The Edges worksheet of NetEvViz for Twitter dataset with Start and End Times](image)

**Network Timeline**

In order to provide a high level overview of the dynamics of a network, we decided to present the user with a Timeline of different graph metrics. The essential idea is to let the user view the change in metrics over time and identify points of interest with respect to those metrics. Graph metrics are different from node metrics in the sense that they present a property of the whole graph instead of components of the graph individually. Examples of such metrics are overall node count, overall edge count, graph density, average betweenness centrality, average clustering coefficient amongst others. We believe that for a user analyzing a new dataset, getting an overview at the highest level is the first important insight. In this project, we implemented the three most basic metrics for temporal graphs, namely, temporal node count, temporal edge count and temporal edge-node density. The user can compute these metrics and plots by clicking the *Timeline* button on the *Temporal* tab of the NodeXL Ribbon. The user can also select the number of sampling points from the drop down box named *Steps*. Three options are provided - 5, 10 or 15, the first one being the default value. According to the number of steps selected, the tool calculates the values of metrics at the different timepoints. The selection of actual timepoints is a separate problem in itself. One way of going about it is selecting the points of maximum change reflected in the graph. However, time being a continuous quantity, it is a rather ill-defined problem to take such an approach. Another option could be to calculate and plot these metrics at all times where there was a change. However, there are obvious performance issues with this approach. In the absence of an obvious method to figure out the timepoints of importance, we adopted the selection of timepoints at uniform intervals of time. Starting from the start time of the first edge and ending in the last time of the edge, the Timeline consists of 5, 10 or 15 equally spaced timepoints. Figure 2 shows the Timeline metric values and plots for a dataset consisting of Twitter replies and mentions described in the section on Evaluation.

**Timeslider**

NodeXL provides a set of dynamic filter sliders corresponding to a column specifying a certain feature of the nodes or edges in the network. These double-headed sliders allow users to interactively filter out nodes and/or edges from the network by selecting the ranges of different features. In NetEvViz, we introduce *Timeslider*, which is implemented based on the built-in dynamic filter sliders in NodeXL. Similar to the above-mentioned sliders, the Timeslider is also a double-headed trackbar which allows users to move its two handlers in order to select a time period. A major difference is that the Timeslider’s values are drawn from the contents of both the *Start Time* and the *End Time* columns. Thus, the range of the Timeslider would be from the earliest time stamp in the *Start Time* column to the latest time stamp in the *End Time* column. We denote these two time points $t_{\text{min}}$ and $t_{\text{max}}$ respectively.

By selecting two time points of interest, the users can visualize the differences of the network at the two specified time points. The two selected time points are denoted by $t_{\text{lower}}$ and $t_{\text{upper}}$ respectively. Figure 3 shows how the *Lower* and *Upper Selected Time* divide the whole time range into three periods. Each edge of the network can be categorized into one of the six disjoint sets based on the relationship between their start and end times with the lower and upper selected times.
Let $t_s$ and $t_e$ denote the start and end time of an edge. The six categories of edges are as follows:

1. $t_{min} \leq t_s < t_{lower} < t_e < t_{upper}$: Set of edges that were added and deleted before the lower selected time $t_{lower}$.

2. $t_{lower} < t_s < t_e < t_{upper}$: Set of edges that were added and deleted between (but not inclusive) the lower selected time $t_{min}$ and the upper selected time $t_{max}$.

3. $t_{upper} < t_s < t_e \leq t_{max}$: Set of edges that were added and deleted after the upper selected time $t_{upper}$.

4. $t_{min} \leq t_s < t_{lower} < t_e < t_{upper}$: Set of edges that were added before the lower selected time $t_{lower}$ and deleted between (but not inclusive) the two selected time points.

5. $t_{lower} < t_s \leq t_{upper} < t_e \leq t_{max}$: Set of edges that were added between the two selected time points and deleted after the upper selected time $t_{upper}$.

6. $t_{min} \leq t_s \leq t_{lower} < t_{upper} \leq t_e \leq t_{max}$: Set of edges that were added before the lower selected time $t_{lower}$ and deleted after the upper selected time $t_{upper}$.

In order to help the user distinguish different sets of edges described above, we add four buttons to let the user choose the colors to represent four different types of edges. The four buttons are shown in Figure 4. In principle, since there are six categories of edges, one can use six colors. However, since edges in categories 1, 2 and 3 do not exist in the network at either of the two selected time points, we decided to use the same color to represent edges in these categories. By default, this group of edges is colored in gray and denoted by “Does not exist in either timestamps” as shown in Figure 4. The other three edge categories are represented by three different colors to be chosen by the user. The default color for category 4 “Only exists in the first timestamp” is green, for category 5 “Only exists in the second timestamp” is blue and for category 6 “Exist in both timestamps” is pink.
To illustrate, we use an example of network consisting of 11 nodes (people) and 21 edges (friendship). Each edge in the network consists of start and end time ranging from 14-Oct-2010 12:00 AM to 25-Oct-2010 12:00 AM. Figure 5 shows the visualization of the network using NodeXL. Figure 6 shows the visualization of the network with the same layout when the Timeslider is used and the two time points selected are 14-Oct-2010 10:03 PM and 21-Oct-2010 12:38 AM respectively. In this new visualization, we can clearly see edges in different categories with respect to the two selected time points. The set of edges which only appear at the first time point is coded in green containing 6 following edges (Wen, Sameer), (Claudio, Wen), (Martin, Jill), (Jill, Steve), (Jill, Anushka) and (Steve, Anushka). There are only two edges which appear in both time points (Craig, Jill) and (Craig, Steve) which are coded in pink. The blue edges are the ones which only appear at the second time point. This set contains 6 edges (Alex, Anwar), (Alex, Sameer), (Wen, Kent), (Craig, Sameer), (Sameer, Kent) and (Jill, Kent). The remaining edges are coded in gray representing the set that does not exist at both time points.

EVALUATION

In order to assess the effectiveness of our new features to NodeXL, we designed and performed an experimental user study. We first decided a list of tasks related to temporal network analysis together with a questionnaire for the user to evaluate and make suggestions after performing the tasks. A pilot study was then conducted on one subject to evaluate our designed tasks. Based on the results and feedbacks from the pilot study, we modified the tasks and questionnaire accordingly and performed the actual usability test on 5 subjects.

Dataset

The dataset used in our evaluation was collected from Twitter using the NodeXL’s Twitter Search importer [3] focusing on the toss-up gubernatorial races in the 2010 United States midterm elections in Georgia between Nathan Deal and Roy Barnes. The importer collected all Twitter users who included a candidate’s full name in their Tweet (e.g., “Nathan Deal”) or alternate variations of their names (e.g., “NathanDeal”). We considered each Twitter user a node and a reply-to or mention relationship as an edge in our network. The choice of including only reply-to’s and mention’s but not the following relationships was due to the fact that only reply-to and mention relationships have the exact time stamps. The data were collected on three days: a) Oct 12, three weeks before the election, b) Nov 2, on the election day and c) Nov 4, post election day. In total, there are 1188 edges and 1015 nodes in the network.

Since our visualization supports networks which have edge additions and deletions, we need the data for the edges’ End Time. Unfortunately, there is no notion of “deletion” of a reply-to or mention relationship. Thus, we assumed that an edge’s End Time is a random date after the edge’s Start Time and the lifespans of all edges are in the range of 3 to 10 days. A node’s timestamp is that of the first edge that the node has in our dataset.

Usability Study

Subjects

In order to make sure that the subjects of the experiments had a good understanding and knowledge about visualizing network data with spreadsheet, we required all the participants to have enough knowledge and experience with both Microsoft Excel and NodeXL. With this requirement we recruited 5 graduate student, denoted by P1, P2, P3, P4 and P5 who are currently taking a graduate-level course in information visualization. The course had included NodeXL in its syllabus which makes all participants familiar with NodeXL. Among these participants, P1 and P3 were also familiar with Twitter data.

Experimental Setup and Procedure
The evaluations were conducted on an Intel Core Duo 2 2.5Ghz laptop with 4GB of RAM. We used the the Twitter dataset described above for the experiments. Subjects were tested independently, using the following procedure:

1. Before the experiment, all participants were explained in general what they were expected to perform in the experiment. They were then instructed to read and sign the Experimental Consent Agreement.

2. Prior to the first session, all participants were asked to complete a questionnaire focusing on their experience on NodeXL and temporal network analysis.

3. All participants were then given a 10-minute tutorial about NodeXL and NetEvViz by the experimenter including

   - Introduction of the new features of NetEvViz including visualizing network metrics over time using the Timeline, comparing network edges at different time points using the Timeslider and editing network data directly from the worksheet.

   - Providing two sample tasks: clicking the Timeline button to generate the graph metrics and sliding the Timeslider randomly to visualize the changes in the graph. These sample tasks were designed to help the participants get familiar with NetEvViz.

4. The participants were asked to perform 3 analytical tasks shown in Table 1. They were asked to think aloud of what they were doing during the experiment and report any problem encountered. At the same time, the experimenter made notes about their actions.

5. Finally, the participants were asked to fill out a post experiment questionnaire described in Table 2. The participants were also asked to leave comments about their experiment with NetEvViz.

Results and Analysis
As mentioned before, we conducted the experiments with 5 participants independently. The main session of the experiment is to perform 3 tasks, each contains from 3 to 7 subtasks as shown in Table 1. It took all the participants from 1 to 3 minutes to complete each subtask. The average time spent on task 1, 2 and 3 are 6, 15 and 10 minutes respectively. The participants spent a large portion of their time on reading the tasks, understanding the meaning of each metric generated by the Timeline and selecting specific edges in the graph visualization.

P1 finished all the tasks successfully. However, P1 had some troubles in understanding the function of the Timeslider. P1 thought that the Timeslider defines a time range, not two specific time points. P1 expressed an opinion: I think you should use two single sliders instead of a range slider.

P2 performed the tasks quickly. P2 thought that the concept of using a Timeslider for temporal network analysis is novel. P2 however encountered some problem with selecting the specific time in the Timeslider.

P3 did not finish task 3 because P3 was not able to select edges and nodes in the graph visualization. The selection function, which does not support selecting edges, is implemented in NodeXL, so we assume this is an existing problem in NodeXL.

P4 did a careful study and provided many useful suggestions, especially in color selection and coding. P4 mentioned that the labels should be attached to the color button. P4 also commented on the color coding of the visualization. P4 said it is very difficult to detect the edges that are activated during specific time period. So it would be much better to filter out gray edges. We agree with her suggestion and expect to make this change in the next release of NetEvViz.

P5 also liked the idea of using color to compare the temporal network at different time points. P5 also thought that it is interesting to use the Timeline to generate and compare network metrics over time. However, P5 thought it would be more convenient to add the selection of date and time in the excel worksheet so that when adding or editing the times of edges, users do not need to manually enter the full text.

After the test, participants were asked to fill out a post experiment usability questionnaire as shown in Table 2. The questions cover the overall performance, the usability of the Timeline and Timeslider and the effectiveness of the color coding. The average score of usability is 6.38 out of 9, with a standard deviation of 2.23. Among the questions, users have highest evaluation for question 6 “I find the feature of changing the color for different types of edges useful” (8.4 points out of 9); question 8 “I am comfortable with changing the color for different types of edges” (8.2 points out of 9) and question 12 “My overall impression of the NetEvViz is positive” (7.4 out of 9). The lowest scored questions are question 10 “I find the Timeslider functions smoothly and correctly as I move the handlers” (4.6 out of 9); question 1 “How to you rate your overall performance on the tasks given” (5.2 out of 9) and question 5 “I am comfortable with using the Timeslider” (5.4 out of 9).

From this questionnaire and the comments of participants, we can conclude that users in general have a positive experience in using NetEvViz. They found it useful to use NetEvViz to visualize and compare temporal networks at two different time points. They were especially in favor of the feature of color selection and comparison. However, they also reported the occlusion of edges made it hard to detect the target edges and inconvenience in selecting the exact time in the Timeslider.

CONCLUSION
In this work, a prototype is built upon NodeXL to explore the potential of using visualization to help the analysis of dynamic network. The prototype contains two new features, the Timeline and the Timeslider. The Timeline gives the user a summary of the network over time, which allows the user to obtain an overview of the changes in the network. Timeslider allows for precise comparison of the differences in the network, which are highlighted with colors, given two times-
Table 1. Analytical tasks used in our user study. The precise time periods and time points used are: \( T_1 \) is from 20-Oct-2010 4:04 AM to 21-Oct-2010 2:20 PM, \( T_2 \) is from 21-Oct-2010 2:20 PM to 23-Oct-2010 0:35 AM, \( t_1 \) is 18-Oct-2010 8:00 PM and \( t_2 \) is 25-Oct-2010 9:00 PM.

<table>
<thead>
<tr>
<th>Task 1: Visualize network statistics over time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. What is the number of edges added to the network during period ( T_1 ) after period ( T_2 )?</td>
</tr>
<tr>
<td>2. What is the number of nodes added to the network during period ( T_1 ) after period ( T_2 )?</td>
</tr>
<tr>
<td>3. How does the density of the network at time ( t_1 ) change after time ( t_2 )?</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Task 2: Visualize the differences of network at different points in time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Move the handlers of the Timeslider in order to compare the network at time ( t_1 ) and ( t_2 )</td>
</tr>
<tr>
<td>2. Change the colors of edges that appear in the network at time ( t_1 ) but not at time ( t_2 ) to Red</td>
</tr>
<tr>
<td>3. Change the colors of edges that appear in the network at time ( t_2 ) but not at time ( t_1 ) to Blue</td>
</tr>
<tr>
<td>4. Change the colors of edges that appear in the network at both ( t_1 ) and ( t_2 ) to Green</td>
</tr>
<tr>
<td>5. How many edges appear in the network at time ( t_1 ) but not at time ( t_2 )?</td>
</tr>
<tr>
<td>6. How many edges appear in the network at time ( t_2 ) but not at time ( t_1 )?</td>
</tr>
<tr>
<td>7. How many edges appear in the network at both ( t_1 ) and ( t_2 )?</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Task 3: Visualize temporal network with interactive spreadsheet</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Add some new edges with time stamps to the Edge Sheet</td>
</tr>
<tr>
<td>2. List all the edges that appear in the network at time ( t_1 ) but not time ( t_2 )</td>
</tr>
<tr>
<td>3. List all the edges that appear in the network at time ( t_2 ) but not time ( t_1 )</td>
</tr>
<tr>
<td>4. List all the edges that appear in the network at both ( t_1 ) and ( t_2 )</td>
</tr>
</tbody>
</table>

Table 2. Post experiment questionnaire used. For each question, we used 9-point scale with 9 being positive.

<table>
<thead>
<tr>
<th>Question</th>
<th>Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. How do you rate your overall performance on the tasks given?</td>
<td>9</td>
</tr>
<tr>
<td>2. I am comfortable using the Timeline.</td>
<td>9</td>
</tr>
<tr>
<td>3. I find the Timeline plots showing the three network statistics (number of edges, number of nodes and density) changing over time useful.</td>
<td>9</td>
</tr>
<tr>
<td>4. I would suggest adding the following network statistics.</td>
<td>9</td>
</tr>
<tr>
<td>5. I am comfortable using the Timeslider.</td>
<td>9</td>
</tr>
<tr>
<td>6. I find the feature of changing the color for different types of edges useful.</td>
<td>9</td>
</tr>
<tr>
<td>7. I am comfortable with detecting different types of edges given two selected time points.</td>
<td>9</td>
</tr>
<tr>
<td>8. I am comfortable with changing the color of different types of edges.</td>
<td>9</td>
</tr>
<tr>
<td>9. I find the ability to edit the spreadsheet while visualizing the network is useful.</td>
<td>9</td>
</tr>
<tr>
<td>10. I find the Timeslider functions smoothly and correctly as I move the handlers.</td>
<td>9</td>
</tr>
<tr>
<td>11. Are you aware of any other tool/software which allows you to compare network at two time points? If yes, please provide the name of the tool/software.</td>
<td>9</td>
</tr>
<tr>
<td>12. My overall impression of the NetEvViz is:</td>
<td>9</td>
</tr>
<tr>
<td>13. I think NetEvViz will be a useful extension for NodeXL.</td>
<td>9</td>
</tr>
<tr>
<td>14. Please leave any comments to the NetEvViz team. Any comment/feedback is much appreciated.</td>
<td>9</td>
</tr>
</tbody>
</table>

tamps which are specified by the user.

A user study was conducted to confirm the usefulness of our prototype. 5 participants with NodeXL experience were recruited. The study follows the procedure of training, pre-experiment questionnaire, performing predefined tasks and post experiment usability questionnaire. Over all, from the feedback of the user study, we can summarize that users have a positive experience and thinks it is useful to use our NetEvViz features to visualize and analysis temporal network data jointly with Excel. Users acknowledged the novelty in the idea of using Timeslider to select the time point and then compare the network instances in the same layout. The users also responded positively to the feature of Timeline and dynamically editing edges and nodes in the spreadsheet. However, they also noted the inconvenience in the user interface and color coding, which resulted in their inefficiency in performing the tasks.

NetEvViz is a first step to understand how NodeXL can be used to analyze dynamic networks. Some possible future work include:

- More sophisticated metrics (e.g. clustering coefficient) can be added to the Timeline beside the three trivial metrics we implemented. The challenge lies in the computation overhead, which may have to be addressed specifically.
- The Timeslider can be extended to color the nodes in a similar way we color edges. Also, it would be interesting to look at visual properties beside color (e.g. shape or opacity).
- Support the analysis of egocentric network by allowing the users to specify the vertex they are interested in.

Proposals to the NodeXL team

Based on our experience with the development of NetEvViz and user feedback, we would propose to the NodeXL development team for incorporating the following:
1. Temporal annotation columns, i.e. Start Time and End Time should be first class citizens of the Network analysis world.

2. Time based filtering of nodes and edges should be present as a separate feature, for instance, using a double slider as shown earlier in this paper.

3. A user should be able to view two versions of a network at the same time, i.e. side by side of each other. The user should be able to switch back and forth between a single and double layout view mode.

4. Network, node and edge metrics should be extended to cover temporal changes.

CONTRIBUTIONS AND ACKNOWLEDGEMENTS
Below are the contributions of the four team members. The tasks performed by more than one member are listed under the credit of each of those involved:

1. Udayan Khurana contributed with ideas and discussions about the subject, exploring NodeXL codebase, designed the Timeline feature, ribbon changes, made the video, and contributed in writing (Abstract, NodeXL, Timeline) and incorporated peer review comments.

2. Viet-An Nguyen contributed with ideas and discussions about the subject, exploring related work, designed the Timeslider feature, processed the data and contributed in designing the user study and writing (Related Work, Timeslider, Usability Study) and incorporated peer review comments.

3. Hsueh-Chien Cheng contributed with ideas and discussions about the subject, designed the Timeslider feature, conducted user study and contributed in writing (Introduction, Timeslider, Conclusion) and incorporated peer review comments.

4. Stephen Xi Chen contributed in designing and conducting the usability study and contributed in writing (Usability Study).

We would like to acknowledge Dr. Jae-wook Ahn (University of Maryland) who mentored the team, providing his guidance and expertise in the subject matters. We also thank Prof. Ben Shneiderman (University of Maryland) for very useful discussions and Mr. Tony Capone (Microsoft) for the guidance on NodeXL related implementation. We are grateful to Dr. Derek Hansen who shared the Twitter dataset for the purpose of evaluation. We thank the six anonymous subjects of the usability testing for their time and feedback. Finally, we thank Nick Gramsky, Rose Kirby, Robert Grove, Sravanthi Bondugula, Varun Nagaraja, Uran Oh and Emre Sefer (University of Maryland) for their constructive critiques which helped a lot in refining this paper.

REFERENCES


