Visualization Techniques for Network Analysis and Link Analysis Algorithms

Ying Zhao[1], Ralucca Gera[1], Quinn Halpin[2] and Jesse Zhou[3]
[1]*Naval Postgraduate School, Monterey, CA, U.S.A.*
[2]*Cornell University, Ithaca, NY, U.S.A.*
[3]*JZ Tech Consulting, San Francisco, CA, U.S.A.*

Keywords: Visualization, Data-Driven Documents (D3), Network Analysis, Lexical Link Analysis (LLA), Smart Data, Automatic Dependent Surveillance-Broadcast, ADS-B.

Abstract: Military applications require big distributed, disparate, multi-sourced and real-time data that have extremely high rates, high volumes, and diverse types. Warfighters need deep models including big data analytics, network analysis, link analysis, deep learning, machine learning, and artificial intelligence to transform big data into smart data. Explainable deep models will play a more essential role for future warfighters to understand, interpret, and therefore appropriately trust, and effectively manage an emerging generation of artificially intelligent machine partners when facing complex threats. In this paper, we show how visualization is used in two typical deep models with two use cases: network analysis, which addresses how to display and present big data both in the exploratory and discovery process, and link analysis, which addresses how to display and present the smart data generated from these processes. By using various visualization tools such as D3, Tableau, and lexical link analysis, we derive useful information from discovering big networks to discovering big data patterns and anomalies. These visualizations become intepretable and explainable deep models that can be readily used by warfighters and decision makers to achieve the sense making and decision making superiority.

1 INTRODUCTION

The US Department of Defense (DoD) faces challenges that demand more deep models to produce intelligent, autonomous, and symbiotic systems to support situation awareness and decision making superiority. Military applications require big distributed, disparate, multi-sourced, and real-time data that have extremely high rates, high volumes and diverse types. Warfighters need deep models including big data analytics, network analysis, link analysis, deep learning, machine learning, and artificial intelligence to transfer big data into smart data. Warfighters then can apply the insight and knowledge generated from big data for decision making and actions. Explainable deep models will play a more essential role for future warfighters to understand, interpret, and therefore appropriately trust, and effectively manage an emerging generation of artificially intelligent machine partners when facing complex threats. Researchers need consider two requirements for understandable, interpretable, and explainable deep models for warfighters and decision makers: The first requirement is to show the process of discovering knowledge, exploring insight from big data and building actionable deep models. The second requirement is to comprehend the resultant smart data and deep models. Visualization provides one of the important components for the two needs. There are two the research questions to address in this paper as follows:

1. How to display and present big data, both in the exploratory and discovery process?
2. How to display and present the smart data generated from these processes?

We show how visualization is used in two typical deep models with two use cases: network analysis, which addresses the first research question and link analysis, which addresses the second research question. They both have the characteristics of discovering and exploring new and high-value information where warfighters lack useful information. The process belongs to the new frontiers of deep analytics with the potentials to handle so-called “unknown unknowns”

Zhao, Y., Gera, R., Halpin, Q. and Zhou, J.
Visualization Techniques for Network Analysis and Link Analysis Algorithms.
DOI: 10.5220/0008377805610568
In *Proceedings of the 11th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2019)*, pages 561-568
ISBN: 978-989-758-382-7

scenarios: We do not know if there are any unknowns in a battlespace.

Many commercially-available data manipulation and visualization tools exist today. Typical spreadsheet tools (e.g., scatter and line plots, bar and pie charts, and bubble and radar charts) are widely used to visualize statistical characteristics to support data-driven reasoning and decision making. Engineers use MATLAB, Octave, and Python libraries to display numeric data and analysis results. Developers use Data-driven Documents (D3) and Javascript to manipulate document object models (DOM) within browsers and generate dynamic and interactive visualizations. Business users use Tableau to generate insights from relational databases and data cubes.

In this paper, we focus on several types of visualization and how they are useful for two deep models such as network analysis and link analysis. Types of visualization considered in this paper include statistical, topical, network, temporal and geospatial. The data types considered include unstructured, structured and network data.

2 NETWORK ANALYSIS

Visualizing data complements the analysis process and enables understanding of the "why" behind the "what" is observed (CED3, 2018).

Inferring the structure of an unknown network is of interest to researchers in the government/military, academia and industry. Generally, the ground truth of a network is not known because it is extremely large or because complete information about it is not available. Thus, researchers make decisions based on the inferred network, whose information is still incomplete, but which acts as the true network. The degree of incompleteness of information is not generally known, since the true network is unidentified and there are no standard techniques to measure their topological difference. This visualization project allows for the exploration of pre-loaded network examples, uploaded networks or the creation of new networks using the left navigation toolbar

Our visualization presents the output of novel algorithms previously introduced to infer nodes and edges in an unknown network (Davis et al., 2016)(Gera et al., 2017)(Wijegunawardana et al., 2017)(Chen et al., 2017). Our algorithms utilize sensors that have the capability to detect neighboring nodes (and their labels) and the edges incident to the sensor. The algorithms infer the networks through a combination of (a) random walks, (b) greedily placing new sensors on a highest degree neighboring node that has been inferred, (c) greedily placing new sensors on a highest degree neighboring node of the current monitor, (d) greedily placing new sensors on highest undiscovered degree node neighboring an inferred node. The algorithms have a probability based restarting feature which varies between restarting at a previously discovered but unmonitored node and a random teleportation to an unexplored node somewhere in the network. The algorithms stop when there is an attempt to place a sensor on a node where a sensor already exists.

We say that a *monitor on node i detects an edge ij* if i and ij are incident, and i detects the label ij of the edge (i.e. the monitor discovers the label of the other end node of ij) (Davis et al., 2016). This then implies that a *monitor on node i detects a node j* if there is an edge ij connecting them. We also allow the monitor on i to discover the deg j (Davis et al., 2016).

The following screenshots overview the interactive site visualizing the progression in discovering a network. Figure 1 displays a network before the discovery process.

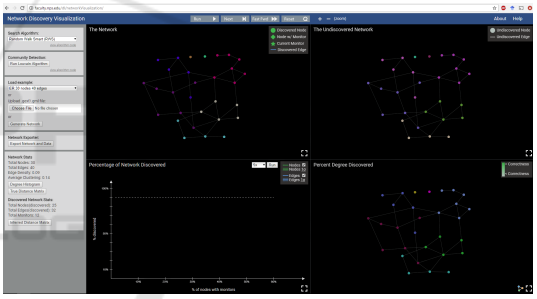

Figure 1: An example of a network to be discovered.

The first column identifies the search algorithm to be used, the network to be analyzed, and the statistical properties of the network as well as the network to be discovered.

The rest of Figure 1 is divided into four quadrants. The top left quadrant, shows the network to be discovered, whose nodes and edges turn green and blue, respectively, as the network is being lit/discovered. The bottom left quadrant shows a temporal progression of the network as it is discovered, both for nodes and edges, color-coded with blue and green to match the first quadrant. Notice that the x-axis doesn't go beyond 60% of monitored nodes, since by that time either the whole network is discovered, or there is no particular strategy needed for the left over part of the network. The upper right quadrant, shows the left-over network that has not yet been discovered; it starts with the original network, and the discovered part is being taken away. The bottom right quadrant displays a network's heat map, a node being colored dark

green if 100% of the neighbors have been discovered, and white if 0% of the neighbors have been discovered, with intermediate percentages represented between white and green.

This visualization was created with the goal of supporting decision makers in planning the discovery of a network, by (1) allowing to visually see what portion of the network has been discovered much like a map would, (2) what percent of the network has been identified, measured both for edges and nodes, and (3) what portion of the network has better coverage through the heat map. The visualization is particularly useful in testing the patterns and performance of different algorithms as discussed below.

The patterns of how the algorithms evolve through networks can be visually seen using the three network quadrants of Figure 1. The same algorithm can be run several times, and while it can visually be observed, the data can be exported using the "Network Exporter" and "Network Statistics" on the very left panel in Figure 1, that capture both the ground truth and the discovered networks one step at the time. The performance of the algorithm can be measured using the bottom left panel in Figure 1, measuring the "Percentage of network discovered" by summarizing several runs of an algorithm displaying the average and confidence intervals (using different fidelity such as $\mu \pm \sigma$, $\mu \pm 2\sigma$, and $\mu \pm 3\sigma$).

While partial network information is sometimes insufficient to make decisions, it is sufficient to influence the process of network discovery. Random walks have been extensively used to explore networks of all sizes. However, alternative, better algorithms to light up a network are useful. We created algorithms that infer networks using a minimal amount of partial information from sampling. We searched for a sampling methodology that minimized the samples needed while maximizing the information each new node reveals about the network, see (Davis et al., 2016)(Alonso et al., 2017)(Chen et al., 2017)(Crawford et al., 2016)(Wijegunawardana et al., 2017).

We measure the captured information by the percent of nodes and edges sampled nodes can observe. Each sampled node (called a monitor) detects its neighbors, and the edges between the monitor and its neighbors. We introduced variants of how the algorithms progress through the network based on discovered nodes' degrees and size of the network (Davis et al., 2016).

We show the progression of a couple o different types of networks. The first network is an Erdós Rényi Random Network (ER) network. In this model the number of nodes and edges are specified, but the distribution of the degrees is random. This network type displays no particular structure being random. The second network is built from four cliques, a modular network identifying how the algorithms work differently if there is structure present.

2.1 First Case Study: A Random Network

We first consider an ER random graph with 30 nodes and 40 edges. Figure 2 displays a temporal snapshot of the first two quadrants on network discovery using a random walk as the inference algorithm to light up the network.

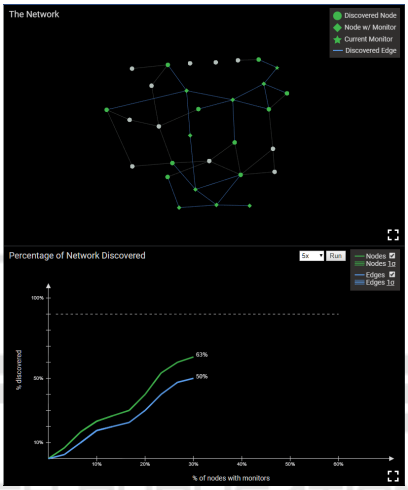

Figure 2: An example of network discovery using a random walk.

Figure 3 displays a temporal snapshot of the first two quadrants of the same random graph with 30 nodes and 40 edges, at the same point in time using the highest degree of the discovered node (Davis et al., 2016) as the inference algorithm to guide the placement of the next monitor to light up the network. One can compare the progression in Figure 3 to Figure 2 in order to see the strength of the algorithms.

2.2 Second Case Study: A Modular Network with Four Communities

We consider a network of 40 nodes, partitioned into 4 cliques of 10 nodes each, with more edges appearing based on a probability of 0.2 between cliques. The communities of the network are identified before lighting it up. Figure 4 displays a temporal snapshot of the first two quadrants of the network using a random walk as the inference algorithm to light up the network. Figure 5 displays a snapshot of the first two quadrants of the network using the highest degree of

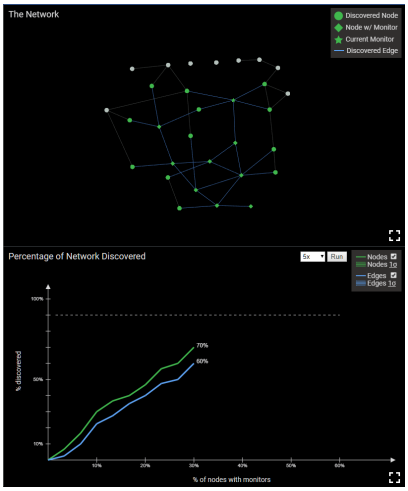

Figure 3: An example of network discovery guided by the highest global degree.

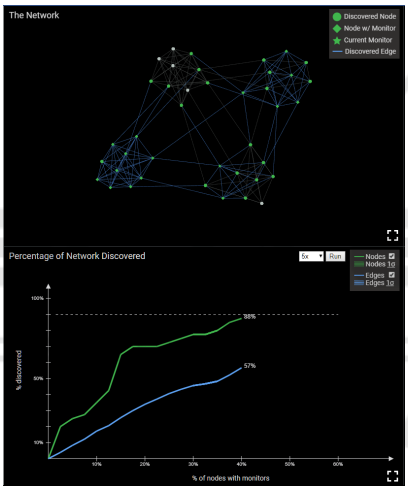

Figure 4: An example of network discovery using a random walk.

the discovered node (Davis et al., 2016) as the inference algorithm to guide the placement of the next monitor to light up the network.

The above use cases illustrate how the network visualization discovery tool gives different methods of network discovery more meaning as we light up the network of concern. That knowledge can immediately lend insight into further decisions based on the discovered network.

3 LEXICAL LINK ANALYSIS (LLA)

We use LLA as an example of deep models (Zhao et al., 2015). In a LLA, we describe the characteristics

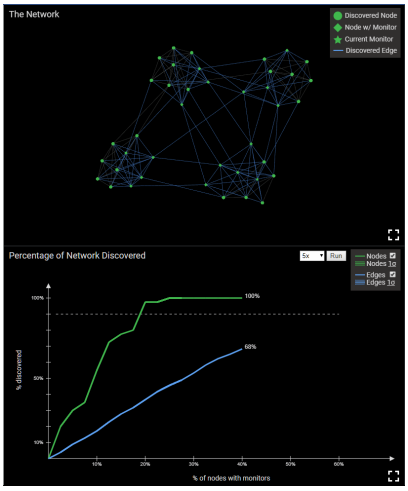

Figure 5: An example of network discovery guided by the highest global degree.

a complex system using a list of attributes or features with specific vocabularies or lexical terms. For example,we can describe a system using word pairs or bi-grams as lexical terms extracted from text data.

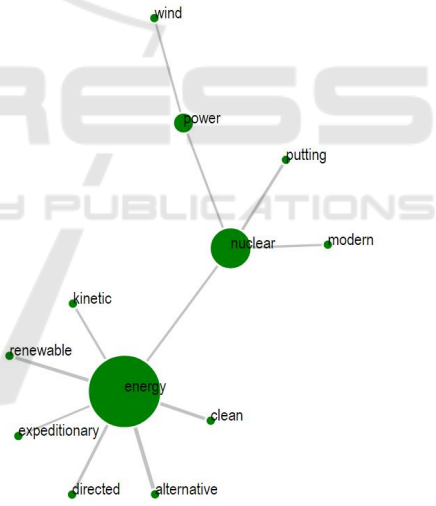

Figure 6: An example of a theme discussed in LLA.

Figure 6 shows an example of a word network discovered from text data using LLA. For a text document, network nodes represent words, and network edges or links represent word pairs or bi-grams between nodes. A list connected words or word pairs forms a network with the center word “energy” as shown in Figure 6. “Clean energy,” “renewable energy” are two bi-gram word pairs examples. The bi-gram method in LLA extends the use of LLA to structured data and combination of structured and unstructured data such as data in the social media

We applied LLA in many use cases to greatly fa-

cilitate the discovery of high-value information in different application domains. LLA outputs smart data such as semantic and social networks , patterns such as rules, associations , themes, and topics . The themes and topics discovered by LLA are further divided into the popular or authoritative, emerging and anomalous information categories. Information users can use authoritative information to discover leadership and archetypes in a social network , use emerging information to discover high-value information from crowdsourcing , and use anomalous associations to identify fraudulent behavior and imposters

The output of LLA includes a file of associations where if word pairs (for unstructured data) or lexical features (for structured data) are linked together. To represent this smart data, we use a variety of visualization methods including D3 detailed below. While, the D3 templates (Bosack, 2017) creates the initial foundation, we designed and implemented custom features and class structure for displaying the smart data output from LLA.

3.1 Use Case: ADS-B Data Analysis

We use a big data set called Automatic Dependent Surveillance-Broadcast (ADS-B) in the context of the Naval Common Tactical Air Picture (CTAP) process and Combat Identification (CID). The CTAP process collects, processes, and analyzes data to provide situational awareness to decision makers. The accurate CID process enables warfighters to locate and identify airborne objects as friendly, hostile or neutral. CID plays an important role in generating the CTAP in the whole kill chain process (Zhao et al., 2016). We downloaded four terabytes historical ADS-B data, sampled every minute, for the whole year (6/2016 to 6/2017) (ADSBexchange.com, 2017). We used LLA to analyze patterns and anomalies in the ADS-B data in an effort to improve CTAP and CID.

3.2 Exploratory and Discovery Process

Exploring and gaining insight from big data starts with visualizing basic statistics, correlations and patterns. In this first step, human analysts use visualization tools to report basic statistical facts within a data set and discover initial patterns and correlations. Human analysts examine the quality of the data, validate and observe initial patterns and compare the knowledge data with their existing knowledge. Here we show examples generated from Tableau and Matlab to display multi-dimensionable (columns, rows, colors and bubble sizes) aggregated data and measures (e.g.,average, total). Figure 7 uses a bubble chart to illustrate the frequencies for each type of aircraft, with the color showing if an aircraft is military or not. The size of bubble shows the number of flights for the aircraft type, e.g., B737 has the biggest bubble. The yellow bubbles represent military aircraft. Commercial aircraft dominate because there are more and bigger bubbles in the plot. Figure 7 is generated using Tableau. Figure 8 shows a MATLAB geospatial visualization of the detailed ADS-B tracks, point-by-point, colored by average absolute heading change in a track. The higher measure of the average absolute heading change may indicate the activities of taking off and landing in an airport area for commercial flights.

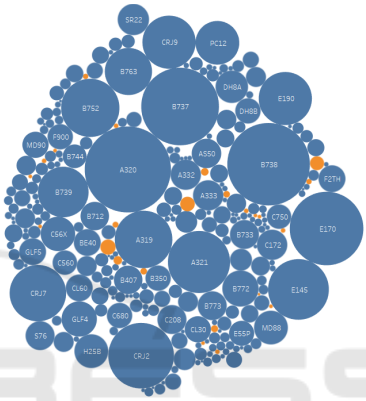

Figure 7: Topical visualization: Bubble charts show military flights (yellow) or commercial Flights (blue) vs. aircraft types. The size of bubble shows the number of flights in the data for the aircraft type.

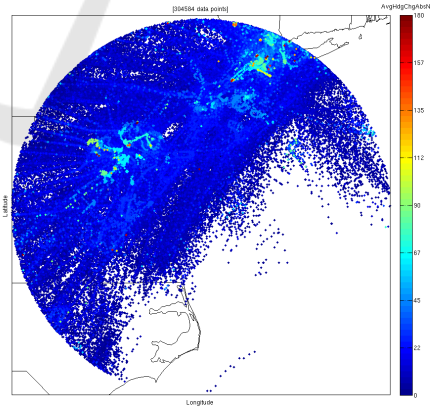

Figure 8: Geospatial visualization: Tracks colored by average absolute heading change. Near the airports: The total heading changes of flights are larger while passing by flights have smaller heading changes.

3.3 Force Directed Graph

To address the research question 2, we consider one of the challenges for LLA is that when anomalous asso-

ciations are considered as a high-value information, it becomes a “needle in a haystack”. In order to find that needle, one has to compute all the associations and then filter accordingly. This can be done in the initial property setting of the LLA tool to some extent. We investigated how to pre-compute all the associations and then filter/select to visualize part of them based on a user’s requirements.

The link or association outputs from LLA are force directed graphs where each node represents a word and each link represents a word association. The nodes are colored according to the node’s anomalous, emerging or popular type. LLA outputs the types. The nodes (or words) are also further clustered into various themes. Each theme contains a group of word pairs that are related to each other based on the data.

The D3 force directed graphs visualization doesn’t show large unfiltered data sets well because the presentation resides in a web browser. The screen begins to lag with too many nodes. The nodes also cannot leave the canvas, so as more nodes are featured they begin to overlap and block other nodes and lead to a cluttered and disoriented view of the data.

Visualizing an unfiltered data set containing a large number of nodes which are very slow to be visualized in the browser. After re-designing the conventional D3 template, we are able to display the same data set with a visible improvement of performance with little latency issues in Figure 9. We improved this visualization by providing tools to filter the data set according to the users needs. For example, one filter the output associations (smart data) from LLA according to the LLA groups (themes), word, or words that begin with ’x’, end with ’x’, or include ’x’, where ’x’ can be any string of alphanumerics. Other filter types include filtering by the strengths of the associations defined using properties initially set by a user. Such properties include node degrees,data sources, association types, and strengths for the output associations. The filter capabilities are important for the research Question 2 because end users often depend on the parameter filters to execute queries and hypotheses.

The forces of the strength of associations on the nodes cluster the nodes automatically, making small clusters naturally lay themselves out in an appealing and easy to view manner. The nodes can also be dragged around and forces can be turned off.

Figure 10 an example of the profiles of aircraft discovered by LLA. Each aircraft track is represented as a time series of kinematic measures such as position (latitude, longitude, and altitude), speed, and heading. Each attribute such as “average altitude” is computed using aggregation statistics such as average or total to the time point for a track. Then such an aggregated track statistics is discretized into three bins: less than (lt) the mean (of the statistics) subtracting one standard deviation, between (bt) the mean subtracting one standard deviation and the mean plus one standard deviation, and more than (mt) the mean plus one standard deviation. For example, the word “avg_alt_mt_31557.5” means average altitude more than 31557.5 feet. These word features are filtered using the visualizer in Figure 9 into five dominant clusters, representing five profiles of the aircraft flying characteristics as follows.

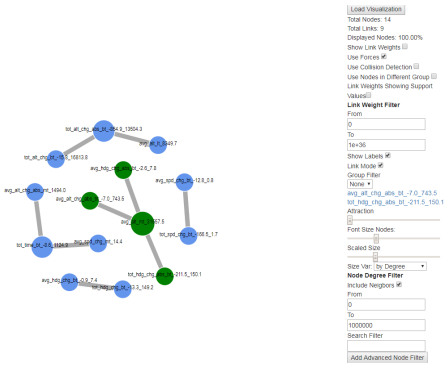

Figure 9: LLA output associations filtered using the filter conditions defined on the right.

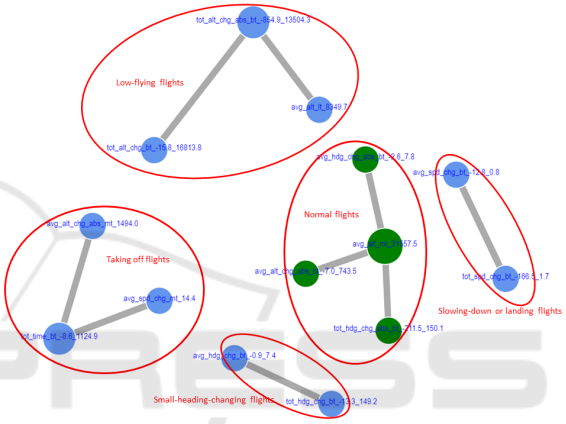

Figure 10: Profiles of aircraft discovered by LLA.

- Normal flights:
  - Average altitude more than 31557.5 feet (avg_alt_mt_31557.5)
  - Average absolute altitude change between 0 and 743.5 (alt_alt_chg_abs_bt_-7.0_743.5)
  - Total absolute heading change between 0 and 150.1 (tot_hdg_chg_abs_bt_-211.5_150.1)
- Low-flying flights
  - Total absolute altitude change between 0 and 13504.3 feet (tot_alt_chg_abs_bt_-

854.9_13504.3)
  - Average altitude less than 8349.7 (avg_alt_lt_8349.7)
  - Total altitude change between -15.8 feet and 16813.8 (tot_alt_chg_bt_-15.8_16813.8, negative change means going down)
- Taking off flights
  - Average absolute altitude change more than 1494.0 feet(avg_alt_chg_abs_mt_1494.0)
  - Average speed change more than 14.4 (avg_spd_chg_mt_14.4)
  - Total time in the area between 0 and 1124.9 seconds (tot_time_bt_-8.6_1124.9)
- Small heading changing flights
  - Average heading change between -0.9 and 7.4 (avg_hdg_chg_bt_-0.9_7.4)
  - Total heading change between -13.3 and 149.2 (tot_hdg_chg_bt_-13.3_149.2)
- Slowing down or landing flights
  - Average speed change between -12.8 and 0.8 (avg_spd_chg_bt_-12.8_0.8)
  - Total speed change between -166.5 and 1.7 (tot_spd_chg_bt_-166.5_1.7)

3.4 Dynamic Time Series

Temporal visualization as shown in Figure 11 used on a LLA output displays sequential patterns. The time variable date is shown on the x-axis. The size of a circle represents the degree of ground truth in a data set. In the ADS-B data set, if a flight is a military or not (mil=1 or 0) is the ground truth of interest. The visualization shows if and how kinematic attributes (e.g., average altitude/speed and change in altitude/speed) and other attribute such as country of the origin (cou), and output types (popular, emerging and anomalous) from LLA are correlated with the ground truth of interest and how the correlation is distributed over time. This visualization can be used in the exploratory and discovery process as well as in the post LLA analysis to address both in research question 1 and 2. For example, in Figure 11, each node is represented using the date (x-axis), y-axis, radius and category. The y-axis can represent any of the numeric variables in the data, *avg_alt_N* as the "average altitude" for a track. The radius can represent any of the numeric variables, e.g., mil=1 or 0 in Figure 11. The category can represent any categorical variable, e.g., cou in Figure 11, using colors. There are three observations that show the insights of the behavior of the aircraft as we can obtain from Figure 11:

- Most flights' average altitudes value between 0 to 40,000 feet
- Flights can fly over 100,000 feet. These can be caused by data errors or anomalies.
- Military flights with larger radius dominate with the country of the origin (cou) of the United States (pink). Some other countries other than the US (other color than pink, e.g., one as highlighted in the attribute list next to one of the gray nodes) also have military flights in the data set, which might be interesting data to investigate more.

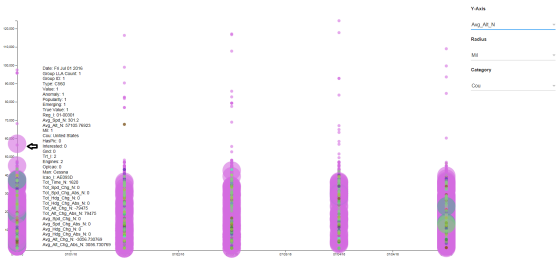

Figure 11: Time series visualization. Radius by mil=1 or 0 (if a track is military or not). Y-axis by average altitude of a track. Category by cou, i.e., the country of origin.

This visualization shows the same data with the flexibility of changing the y-axis and radius dynamically depending on the attributes a user chooses. The user can also hover over a data point (e.g., the point next to the arrow) to see all of the information displayed at once. In addition, because required computations are fairly simplistic, this visualization faces fewer latency issues compared to that of a Force Directed Diagram.

3.5 Virtual Airways of the ADS-B Data

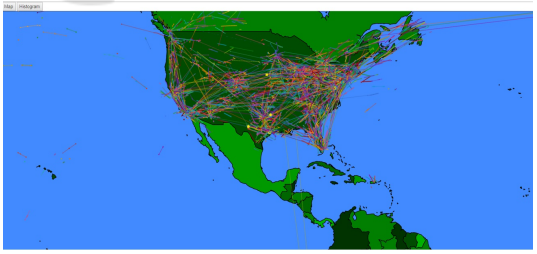

Figure 12: Zoomed-in look at ADS-B visualization. Each line represents the trajectories of all planes that take off at the same specific time and fly in that area over 100 minutes.

The goal of this visualization aims for a user to view the trajectories of airplanes and then filter out based on the requirements of different velocity, flight patterns, destinations, and arrivals. This visualization was created with the D3 projection API . The visualization can display the signal of every plane at any

given moment and animate the planes movement from minute to minute. It can also display all planes that take off at the selected time and show where that plane goes in the scope of all the files loaded into the server. This visualization is limited to the size of the data set. To play 10 minutes of flights, ten 5KB files need to be loaded in and looped through to show all of the data. Therefore, as more time is displayed, latency increases. Current features of this visualization include zooming- in and out and panning to different sections of the world as shown in Figure 12.

4 CONCLUSION

Visualizations help users gain insight from big data. We showed in this paper various visualizations in order to help users understanding big data as well as to extend the users' understanding of smart data through deep models such as link analysis and network analysis. The visualizations implemented for LLA and network analysis vary in complexity and offer some breadth to the viewers. By using D3, Tableau, and MATLAB visualizations, we derived useful information from discovering big networks to discovering big data patterns and anomalies.

What are the challenges of future visualization? Assessing data visualizations includes using heuristic evaluation and user studies. Future work for these visualizations includes designing and developing visualization types associated with the nature of deep models, data types and business problems, and making the visualization easy to use for human analysts both in the pre- and post- analyses of big data. This should be an ongoing effort to improve understandable, intepretable and explainable deep models that can be readily used by warfighters and decision makers to achieve superiority.

ACKNOWLEDGEMENTS

Thanks to the Naval Research Program at the Naval Postgraduate School, the Office of Naval Research (ONR), and the SBIR contract N00014-07-M-0071 for the research of lexical link analysis and collaborative learning agents at Quantum Intelligence, Inc. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied of the U.S. Government.

REFERENCES

ADSBexchange.com, L. (2017). Ads-b exchange.

Alonso, L., Crawford, B., Gera, R., House, J., Knuth, T., Méndez-Bermúdez, J., and Miller, R. (2017). Identifying network structure similarity using spectral graph theory. *Applied Network Science*.

Bosack, M. (2017). D3 data-driven documents.

CED3 (2018). Center for educational design, development, and distribution.

Chen, S., Debnath, J., Gera, R., Greunke, B., Sharpe, N., and Warnke, S. (2017). Discovering community structure using network sampling. In *32nd ISCA International Conference on Computers and Their Applications*. Springer.

Crawford, B., Gera, R., House, J., Knuth, T., and Miller, R. (2016). Graph structure similarity using spectral graph theory. In *International Workshop on Complex Networks and their Applications*, pages 209–221. Springer.

Davis, B., Gera, R., Lazzaro, G., Lim, B. Y., and Rye, E. C. (2016). The marginal benefit of monitor placement on networks. In *Complex Networks VII*, pages 93–104. Springer.

Gera, R., Juliano, N. R., and Schmitt, K. R. (2017). Optimizing network discovery with clever walks. In *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017*, pages 1217–1224. ACM.

Wijegunawardana, P., Ojha, V., Gera, R., and Soundarajan, S. (2017). Seeing red: Locating people of interest in networks. In *Workshop on Complex Networks CompleNet*, pages 141–150. Springer.

Zhao, Y., Kendall, W. A., and Young, B. W. (2016). Big data and deep analytics applied to the common tactical air picture (ctap) and combat identification(cid). In *Proceedings of the 8th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*, volume 1 of *IC3K 2016*, pages 443–449.

Zhao, Y., MacKinnon, D., and Gallup, S. (2015). Big data and deep learning for understanding dod data. In *Journal of Defense Software Engineering, Special Issue: Data Mining and Metrics*.