INTERACTIVE DASHBOARD FOR USER ACTIVITY USING NETWORK FLOW DATA

by

LALIT KUMAR NAIDU

Presented to the Faculty of the Graduate School of The University of Texas at Arlington in Partial Fulfillment of the Requirements for the Degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE AND ENGINEERING

THE UNIVERSITY OF TEXAS AT ARLINGTON

December 2015
To my father Tulsi Das and my mother Rajani
ACKNOWLEDGEMENTS

I would like to acknowledge my supervisor Mr. David Levine who has been my teacher and friend. Thank you for the support and guidance, and much freedom to express and execute my ideas that has been crucial towards my professional and personal growth. I would like to acknowledge and sincerely thank my committee members, Dr. Farhad Kamangar and Dr. Ramez Elmasri for their constructive feedbacks and support. I would also like to extend my sincere thanks to Dr. Giacomo Ghidini and Dr. Steve Emmons for their valuable suggestions and guidance which helped me greatly though the course of this thesis.

I’d like to thank the administrative staff, specifically Ms. Pam McBride and Ms. Sherri Gotcher for their valuable support and services. Special thanks to Ms. Camille Costabile in helping me through the final requirements of my thesis, and the department of computer science, UTA.

My heartfelt thanks to my family and to Jananee Narayanan for continuous support and inspiration. Last but not the least thanks to all my friends who have kept me motivated through this journey and made it memorable.

December 3, 2015
ABSTRACT

INTERACTIVE DASHBOARD FOR USER ACTIVITY USING NETWORK FLOW DATA

LALIT KUMAR NAIDU, M.S.
The University of Texas at Arlington, 2015

Supervising Professor: David Levine

Data visualization is critical in analytical systems containing multi-dimensional dataset and problems associated with increasing data size. It facilitates the data explanation process of reasoning data and discovering trends with visual perception that are otherwise not evident within the data in its raw form. The challenge involved in visualization is presenting data in such a way that helps end users in the process of information discovery with simple visuals.

Interactive visualizations have increasingly become popular in recent years with prominent research in the field of information visualization. These techniques are heavily used in web-based applications to present myriad forms of data from various domains that encourage viewers to comprehend data faster, while they are looking for important answers.

This thesis presents a theme for visualizing discrete temporal dataset (pertaining network flow) to represent Internet activity of device (interface) owners with the aid of interactive visualization. The data presentation is in the form of a web-based interactive dashboard with multiple visual layouts designed to focus on end user
queries such as who, when and what. We present an "event map" as a component of this dashboard that represents user activity as collections of individual flow from the dataset. In addition, we look into design issues, data transformation and aggregation techniques involved in the narration of data presentation.

In addition to studying visualization methods, an outcome of this thesis is a functional proof-of-concept, which allows demonstration of a network flow dashboard that can be served as a front-end interface for analytical systems that use such data (network flow).
TABLE OF CONTENTS

ACKNOWLEDGEMENTS ...................................................... iv
ABSTRACT ......................................................................... v
LIST OF ILLUSTRATIONS .................................................. viii
LIST OF TABLES ............................................................... ix

ChapterPage
1. INTRODUCTION ............................................................ 1
   1.1 Introduction and background ...................................... 1
   1.2 Motivation behind the thesis ....................................... 4
   1.3 Goals of the thesis .................................................. 5
   1.4 Organization of the thesis ......................................... 5
2. RELATED WORK ............................................................ 7
   2.1 Understanding Data .................................................. 7
   2.2 Visualization ......................................................... 8
      2.2.1 Color ............................................................. 9
      2.2.2 Charts and forms of Data Visualization ................. 10
      2.2.3 Brush and Linking, Focus and Context ................. 13
   2.3 Visualization on web ................................................. 14
      2.3.1 D3(data driven documents) .................................. 15
3. CONTEXT AND CHARACTERISTICS OF DATASET .............. 16
   3.1 Context of Dataset .................................................. 16
   3.2 Characteristics of Dataset ........................................ 18
4. CONCEPT AND DESIGN .................................................. 20
## LIST OF ILLUSTRATIONS

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>An example of simple bar chart where the length of bar is the value associated with it.</td>
<td>3</td>
</tr>
<tr>
<td>2.1</td>
<td>Illustrates different color schemes a) sequential, b) diverging and c) quantitative.</td>
<td>10</td>
</tr>
<tr>
<td>2.2</td>
<td>Bar chart represents a quantitative color scheme.</td>
<td>10</td>
</tr>
<tr>
<td>2.3</td>
<td>Heatmap in the form chart representing daily count.</td>
<td>12</td>
</tr>
<tr>
<td>2.4</td>
<td>Illustrates equity stocks in the form of heatmap, stocks colored red are down whereas stocks colored green are doing well.</td>
<td>12</td>
</tr>
<tr>
<td>2.5</td>
<td>An interactive time series chart illustrating the use of brush, context and focus.</td>
<td>14</td>
</tr>
<tr>
<td>3.1</td>
<td>Simple architecture shows the flow of data.</td>
<td>18</td>
</tr>
<tr>
<td>4.1</td>
<td>Two different categorical information vertically with five different overlapping time elements horizontally.</td>
<td>21</td>
</tr>
<tr>
<td>4.2</td>
<td>Four similar events occurring concurrently.</td>
<td>21</td>
</tr>
<tr>
<td>4.3</td>
<td>Time-series line chart, vertical axis is price (dollars)[1].</td>
<td>24</td>
</tr>
<tr>
<td>4.4</td>
<td>Multiple line chart, vertical axis is temperature (Fahrenheit)[2].</td>
<td>24</td>
</tr>
<tr>
<td>4.5</td>
<td>Transformation of intervals into triangular models.</td>
<td>25</td>
</tr>
<tr>
<td>4.6</td>
<td>Illustration of an event-band with various attribute grouped as a theme, the extent of the time-line is the fixed year in the theme.</td>
<td>26</td>
</tr>
<tr>
<td>4.7</td>
<td>Illustration of multiple event bands for three category values(Social media, Email and Torrent).</td>
<td>27</td>
</tr>
</tbody>
</table>
4.8 Illustration of multiple event bands for three category values (Social media, Email and Torrent) ........................................... 28
4.9 Illustrates various stages of data in the process of visualization ...... 29
4.10 Internal communication architecture of the dashboard .................. 30
4.11 Depicts a simple wireframe of the dashboard ............................. 32
4.12 Different layouts of chart for visualization focused towards user-centric questions ................................................................. 32
5.1 Illustrates various cases how two intervals may overlap, 1) b overlaps a from tail, 2) b overlaps a from head, 3) a and b super impose each other, 4) and 5) intervals do not overlap ................................. 36
5.2 Illustrates the order of intervals in the same category (colored blue) and resultant interval after removing the overlap (colored green) .... 37
5.3 Illustrates the sorted list of intervals from Figure 5.2 ...................... 38
5.4 Illustration above is for a selected time range, intervals which are out of bounds are clipped or excluded during aggregation ...................... 40
5.5 Netflow summary dashboard representing event chart with overlapping events, time ranges where category colors are intense are the busiest regions ................................................................. 42
5.6 Illustrates the anatomy of the dashboard .................................... 43
5.7 Illustrates the components in the control panel of the dashboard .... 43
5.8 Dashboard representing event chart when overlapping events where removed, color intensity is low for events and individual events are clearly visible, on hovering over each event details can be viewed about it .... 44
5.9 Dashboard representing events chart when a time range is selected from the brush above the chart .................................................. 44
5.10 Dashboard representing, categorical usage in hours, vertical axis represents hours

5.11 Dashboard representing, categorical usage in hours, vertical axis represents hours, sort button is provided to dynamically change the chart with sorted bar chart

5.12 Dashboard representing categorical data usage of the selected device

5.13 Dashboard representing categorical data usage of the selected device in selected time range from the brush above the chart

5.14 Dashboard representing total usage of the selected device, bar represents the usage in hours for a day and donut represents percentage of upload and download data
## LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Sample record of decorated flow, which represents different attributes of the dataset (for brevity below table shows only those attributes which are used for analysis).</td>
<td>19</td>
</tr>
</tbody>
</table>
CHAPTER 1

INTRODUCTION

1.1 Introduction and background

We live in the information era where data is being collected, stored and analyzed from ever increasing new streams. This has been possible due to the rapid progress made in computer power, storage and networking technology coupled with the growth of the Internet. In the later half of the past decade global IP traffic has increased more than fivefold, and it is expected to triple in the next few years by 2019 [3]. The rapid advancement in technology has resulted in vast amounts of data that comes in different forms and complexity. Such data in their raw form have multiple dimensions and contain intricate details of the system from which they have been generated. Exploration of these datasets (which represents a system) represent an important process and can bring about hidden meanings and facets, which are not obvious to the naked eye, at first glance. The task of exploration and analysis becomes tougher when the volume of raw data becomes overwhelming, which makes it necessary to scale down and transform into a understandable format. This, in addition to effective methods of analysis and representation, make it possible to derive deeper hidden insights.

Card, McKinley, and Shneiderman[4] describe visualization as the use of computer-supported, interactive visual representations of data to amplify cognition. Data visualization can be described as the representation of data in a pictorial or graphical format, which is visually appealing and simple to analyze. For many years, people have used visual representations such as maps and some kinds of charts to better
understand data from different streams more easily and quickly. In other words, data visualization is one way of deriving information from the data. With the size of data ever-increasing, decision makers at all levels are beginning to realize the power of visualizations finding relevance among the millions of variables, communicate concepts and hypotheses to others, and even predict the future.

Because of the way the human brain processes information, it is faster for people to grasp the meaning of many data points when they are displayed in charts and graphs rather than poring over piles of spreadsheets or reading pages and pages of reports. The success of this kind of an approach critically depends upon the design, concept, readability and scalability of the visual.

Another vital part of visualization is the process of mapping information or data to visual elements i.e. devising ways to interpret data and express values as visual properties. An example of this could be a simple bar chart generated with sets of rule such as larger values mapped to taller bars and a second dimension as color for objects of each value (Fig 1.1 demonstrates a simple bar chart). This approach is more data driven and can be further extended to control datasets with convoluted or complex dimensions. Examining deeper we see that the process of mapping data with visual elements has to be automated such that any changes in data would also mean change in visual elements by means of an interaction available throughout the system. Such interactions enable the users to view data in a meaningful way and help in the process of data exploration and subsequent consumption to make meaningful decisions.
Figure 1.1. An example of simple bar chart where the length of bar is the value associated with it.

Significant contributions have been made in the field of information visualization in recent years, notable among them is the seminal work by Ben Shneiderman in his paper at the 1996 IEEE Visual Languages Conference [5, 6]. In it he explains the visual information seeking mantra, as a concept of presenting data to end-users, where the mantra or the secret sauce (process) for effective data presentation is overview first, zoom and filter, then details-on demand. This technique of design pattern is ubiquitous in visual platforms and helps in user driven data exploration.

Using web technologies to visualize data is a faster and wide spread method of publishing visual artifacts, since larger audience or user-base use browsers (becomes easier regardless of the type of device being used), it is important to utilize web-based toolkits and libraries built for data visualization. D3 is a powerful web-based library built using JavaScript, and can be used to implement custom web-based visualization. The design of D3 is very similar to JQuery and supports method chaining much like the later. The core concepts of data driven visualization are supported in the library through selections and data joins which are methods to filter and bind input data to DOM elements (mapping data to visual objects) [7].
In this thesis we apply the existing interactive data visualization techniques combined with the core design principles of interactive data visualization on a specific domain of discrete time series dataset in the form of network flow which will help end-users in answering questions in which they might be interested. The output of this thesis is an interactive dashboard which is a proof-of-concept for a part of machine-to-machine (M2M) system that facilitates an interface for end-users that provides trends in Internet activity. This is achieved by using web technologies like HTML, CSS, JavaScript and D3.js. The design of dashboard, charts and interactive elements are such that it leverages end-users in the process of data-exploration and could also aid in identifying any anomaly patterns.

1.2 Motivation behind the thesis

Effective visual data exploration involves design patterns that are user centric and does not mainly focus on the nature of data. The key aspects are overview first, filtering and zooming and providing data on-demand. Overviewing involves understanding the shape and overall size of data, it helps discern important relationships and patterns that might not be otherwise obvious. In addition, it assists in targeting specific data ranges in which the user might be interested. Zooming and filtering refers to magnifying subsets of information by filtering extraneous data, it assists user in detailed investigation of minutiae pockets of data that may further help in rediscovering new information or trend. Filtering reduces the complexity of data without modifying the data representation or user’s view. Typically, a good vantage point in a visualization is when overview is not cluttered (this could happen due to the limitations in screen size and the visual complexity of data) and representation of information is clear, it is important to present information as point-by-point on users demand with the help of usable components like navigation bar and selection menus.
Considering the design patterns and applying them to visualize data will result in the discovery of new trends that might not be possible in other cases such as a static chart. We use the existing examples of interactive charts to focus on user’s need of searching answers for important questions and this process also help us keep our target audience engaged with the system.

1.3 Goals of the thesis

The broader goal of this thesis is to provide a simple analytical tool with the help of interactive visualization in the form of a dashboard that focuses on finding useful trends in a user’s daily online activity. Network traffic data collected in the form of IP flow presented as charts are implemented in the dashboard. Charts and dashboard being interactive can capture useful trends and focuses on key questions commonly asked, hence the dashboard built in the process is a proof of concept for a bigger system provides such services to its end users in the form of interactive visual interface.

1.4 Organization of the thesis

This thesis is divided into 6 chapters, first chapter is dedicated to introduction, background and motivation and goal. Chapter 2 is about related work and terminologies that have been used throughout this thesis. Chapter 3 explains the background and environment of the dataset that is being used for visualization. Chapter 4 explains the concept and design of the framework of dashboard, here we explain different interactive components, color schemes, type of charts and how they help in data exploration process. Chapter 5 is about the implementation and explanation of
algorithm being used. Finally, we conclude this thesis in Chapter 6 by outlining the future work.
CHAPTER 2
RELATED WORK

In recent years there has been swift progress in the field of information visualization and it is on the verge of a paradigm shift. This can also be attributed to information overload which happens when the amount of input (data) to a system exceeds its processing capacity. The quality of decision making is affected by size of data and degrades faster with large amounts of data; relation between quality in decision and data is inversely proportional such that when amount of data increases quality in decision decreases [8], it therefore becomes important to perceive data in different forms that will alleviate this situation. Extensive research has been conducted in the field of information visualization, various tools, frameworks and models have been designed in recent years spanning almost all domains related to science, sociology, geology, business and finance, such works have hugely impacted the way data is comprehended. The topics covered in this thesis are drawn from related research work in the field of information visualization (infoviz) and this chapter explains those terminologies and concepts briefly. Vocabulary from this chapter is used throughout this thesis document.

2.1 Understanding Data

Data can be collected, stored and measured. A dataset is collection of data items where it can also be perceived or cognitively consumed in tabular form with columns representing dimensions and rows as individual records. Such multidimensional dataset identifies each item uniquely that represents both row and dimension
(it can be understood in the form of matrix, where rows and columns being uniquely represented with their position). It can also be seen as a collection of variables and the study associated with such multivariate dataset is known to be multi-dimension data analysis or otherwise multivariate data analysis. These variables further can be grouped as set of dependent and independent variables where dependent variables are typically described as a function of independent variables [9]. Lex [10] describes in-homogeneity or homogeneity as a fundamental property of such multivariate datasets related to data diversity. Inhomogeneity is referred as differences in data items within the dataset and can be classified based on characteristics (refers to type), semantics (refers meaning of data item) and statistics (behavior or distribution of data items). Where homogeneity refers to high diversity of evenly distributed data items within the dataset. Data comes in various sizes and depends on the type of system and domain it represents. In the scale of bytes, it ranges from a single byte to petabytes of data. The size of data can also be ranked as small, medium or large(big). 'Big Data' are words coined for large and complex dataset typically ranging in few gigabytes to petabytes, where as medium sized data and small sized data range from few kilo bytes to mega bytes and few bytes to kilobytes respectively. Storage systems and analytical platforms differ based on size and scale of data. Further visualization schemes and framework also vary based on same parameter.

2.2 Visualization

The core of good visualization includes several concepts and nuanced techniques studied well over time. Large amounts of data can be understood using the visual perception with a well designed data visualization scheme. The following subsections explains each of those concepts briefly.
2.2.1 Color

Color is a fundamental and essential part of data visualization; hues add layer or dimensions to visualization which highlights hidden relations in data that are otherwise difficult to observe. Color used well can enhance the clarity of presentation and can effectively communicate the information that needs to be delivered to its audience conversely if poorly used, can obscure and muddle the meaning of data. Another interesting aspect of color or wavelengths of light is, it is not perceived in an absolute manner, but rather is affected by the background that surrounds it. Figure 2.1 shows a good use of color, where it strongly correlates with the meaning of data being presented. Brewer[11] has explained color schemes as brewer palates classified as three broad types Figure 2.2 illustrates these color scheme, i.e. sequential, qualitative and diverging.

Types of Brewer palettes:

1. qualitative: unlike sequential color schemes, qualitative color schemes don’t imply order but rely on the differences in hue to create a color scheme. Class of data applying such color schemes do not have have conceptual ranking.

2. diverging: this scheme is used when the class of data being represented, significantly differ from each other at a point, this allows to emphasize important break point. The color scheme has to be adjusted based on the number of break points that is required in the chart.

3. sequential: sequential color schemes can be used to represent data range in a particular order, either on an ordinal scale or on a numerical scale. In general, light and dark colors are associated with low and high data values, respectively. Single-hued and sophisticated multi-hued color schemes may be used for data visualization.
2.2.2 Charts and forms of Data Visualization

William Playfair[12] has been attributed to be the first person who devised charts to present statistical data and most notably among all his contributions are
time-series line graph, circle-chart and the bar chart. Since then, charts have evolved and in modern day visualization they represent information interactively. Color overlays can be added or removed dynamically, key points can be highlighted with the help of visual highlighter and tool tips. Figure 2.3 represents an example chart of the heat map of daily counts [13], the use of abstract forms of chart to represent information is not very old and provides an effective means of reasoning quantitative information. According to Tufte [14] well-designed data graphics are customarily the simplest and most powerful. Data presentation is not limited to charts and extends to other forms of visual presentation such as heat maps, data clusters in the form of scatterplots, network graphs, etc. Heatmap is a way of graphically presenting the data where data values or items contained in a matrix layout are represented with different colors [15]. This kind of visualization is also referred as thematic representation of data, where color of data attributes to a theme, like the value of an equity stock (up, low or no loss), this type of presentation layout provides easy way of classifying and summarizing data and its environment. Figure 2.4 illustrates the advantage of using heatmap for presenting data [16].
Figure 2.3. Heatmap in the form chart representing daily count.

Figure 2.4. Illustrates equity stocks in the form of heatmap, stocks colored red are down where as stocks colored green are doing well.
2.2.3 Brush and Linking, Focus and Context

Displaying data with static displays can be attributed to poor data explanation, however if the same data is presented interactively it serves the purpose. Brush and linking is an important theme of an interactive chart that helps in the process of data explanation. The idea is to combine various presentation techniques to overcome the shortcomings of a single method [17]. Brushing is an act of selecting data from the visual space (chart or map) which is similar to the operation of selecting words in a document where the selected words are highlighted. Here instead of a document, the context is a visual space and the items selected are visually encoded data in the space. Linking is referred as a cause of brushing operation and commonly takes place in the background whose effects are simultaneously (with a delay in fractions of a second normally in milliseconds) visible in the visual space along with the brushing operation, in a scenario of multiple charts in a visual space a brushing operation could be selecting elements such as dots in a scatterplot or bars in a bar chart which results in propagating this event and modifying other charts in the same visual space, such as highlighting the connected element selected by the brush. Zooming visual elements particularly becomes important when the dataset is large, this is partly due to the limited screen space that makes the visual items (encoded data) densely packed resulting in an overcrowded presentation of overall information or data. Thus zooming helps by focusing visual elements that otherwise would be difficult to view. Visual context (overview) of the whole dataset is lost by just focusing (detailed view) on subjected visual element which could be vital in the data explanation process, for example such layout may help in determining the relative size of the subjected visual element when compared to the whole dataset facilitating the detection of anomaly pattern in the dataset, for this reason it is vital to have detailed view of the visual element without losing its context. Separating context from its focus as shown in
Figure 2.6 is an effective layout that aids viewing comfort in data explanation process [18], where the smaller chart is context and above it is focus. Due to the physical separation of the two views (context and focus), users interact with the views separately such that the changes made in the context reflects immediately in the focus [19, 20]. This helps the viewer to comprehend the subjected visual elements relative to the whole dataset; inclusion of brush and linking in context enables zooming that provides granular view of selected elements in the focus.

Figure 2.5. An interactive time series chart illustrating the use of brush, context and focus.

2.3 Visualization on web

Shared representation of a page made possible by the document object model (DOM) seamlessly integrates various web technologies (i.e. HTML, JavaScript, CSS, SVG) thereby providing a hierarchical structure. This is an object oriented approach of representing a web page that facilitates dynamic modification of web contents, and tools selected for implementing data visualization should use it.
2.3.1 D3(data driven documents)

D3.js is an open source JavaScript library for creating interactive data visualization. Bostock at el. [7] designed it such that it leverages imperative programming style and easily integrates with other JavaScript libraries (JQuery). It is freely available on GitHub, released under a BSD license. Before d3.js, ‘prefuse’ [21] a compile language for data visualization written in Java was introduced as the first web-based data visualization toolkit available to less-experienced programmers [22]. Development of other toolkits followed by ‘prefuse’ were Flare written in ActionScript and viewed on web using Adobe’s Flash Player that required no plugin and later Protovis [23] which was an open source library written in JavaScript which is replaced by d3.js. D3 directly operates on DOM by binding data to nodes (visual elements in the page) and later modifying them to create interactive visualization. It facilitates manipulation of web documents through loading, binding(data), transforming and translating (visual elements). D3 recently has become popular in the data visualization community due to the availability of rich and well documented API’s that help in creating interactive web-based charts and makes debugging easier [24].
CHAPTER 3
CONTEXT AND CHARACTERISTICS OF DATASET

3.1 Context of Dataset

The dataset used in this thesis is a network flow or traffic flow that uses NetFlow protocol. NetFlow is a type of IP flow introduced by Cisco, it captures the network information by collecting IP flows during packet ingress and egress phase at the router or switch. The flow records include attributes such as the IP address of source and destination, port number, timestamp of flow ingress, input and output interface, packet and byte count, direction of flow, type of service, etc. (Table 3.1 displays a summary of dataset). IP flow information is of specific interest to network administrators, and the majority of research efforts in network traffic analytics is in this context, many tools have been developed which provide insight with the help of dashboards and visual aids to detect an anomaly and unusual patterns. The analysis of IP flow dataset from a network traffic administrator point of view can be extended further to answer other important questions which relate to an end-user of a device or interface who are usually interested in trends of day to day Internet activity.

The overall architecture of the system which uses this data is depicted in Figure 3.1. Network information or IP packets are collected in the form of IP flows, a wireless switch (Netflow exporter) is installed (the traffic is diverted through this device) that acts as an intermediate between the interface (devices such as mobile phones, tablets or computer) and the router which connects to the Internet. The job of the switch is to aggregate individual IP packets into a single flow (flows are always considered unidirectional) based on a certain set of pre-defined rules (a flow may be derived by
considering source and destination IP address, port number or combination of both, Quality of Service). These flows are then transmitted by the switch to be collected in a repository (such as Amazon S3).

Before the data is prepared for exploration or visualization it passes through filtration stages, where the flow is analyzed for foreground and background traffic followed by classification into categories (since these stages are not the part of this thesis they are briefly explained). Background traffic involves data that is generated due to periodic polls of applications to the server. Such information does not necessarily indicated user driven activity or communication, therefore is not important and is eliminated from the system. The output of this stage constitutes the majority of foreground traffic and relates to user driven communication which is then passed to the next level; in the classification stage unsupervised machine learning techniques are applied to label the foreground traffic with appropriate categories which are closely related to end-user activity (examples of such meaningful categories could be social networking, email, search etc.). Once categorized, the data is ready to be visualized and further explored by the end-user.
3.2 Characteristics of Dataset

1. Flow: each data point in Netflow dataset represents a flow. Flows are collection of data packets captured in a session based on set of rules (explained in the data context above in section 3.1).

2. Event: each flow corresponds to an interval or event in time. Such events are triggered by either the end-user or some software application.

3. Duration: length of an event in the dataset; this information is already given, so end-time of an interval can be computed using the start-time and duration of the flow.

4. Activity: this could be a single event or collection of events in a selected range in time.

5. User Activity: user driven or application driven categorical Internet activities.
Table 3.1. Sample record of decorated flow, which represents different attributes of the dataset (for brevity below table shows only those attributes which are used for analysis).

<table>
<thead>
<tr>
<th>SL</th>
<th>Attributes</th>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>dateTimeFirstSeen</td>
<td>time stamp of when ip flow started</td>
<td>2015-09-10 T14:24:35.006Z</td>
</tr>
<tr>
<td>2</td>
<td>duration</td>
<td>duration of ip flow</td>
<td>64.293</td>
</tr>
<tr>
<td>3</td>
<td>protocol</td>
<td>is an id for the protocol above ip</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>endPointA.ipAddress</td>
<td>endpoint A ip address</td>
<td>54.225.241.76</td>
</tr>
<tr>
<td>5</td>
<td>endPointA.portNo</td>
<td>port no of endpoint A</td>
<td>443</td>
</tr>
<tr>
<td>6</td>
<td>endPointB.ipAddress</td>
<td>Endpoint B ip address</td>
<td>192.168.22.153</td>
</tr>
<tr>
<td>7</td>
<td>endPointB.portNo</td>
<td>port no of endpoint B</td>
<td>59700</td>
</tr>
<tr>
<td>8</td>
<td>outboundByteCount</td>
<td>byte count for outbound data packets</td>
<td>1320</td>
</tr>
<tr>
<td>9</td>
<td>inboundByteCount</td>
<td>byte count for inbound data packets</td>
<td>666</td>
</tr>
<tr>
<td>10</td>
<td>category</td>
<td>category for this flow summary</td>
<td>AWS Amazon</td>
</tr>
<tr>
<td>11</td>
<td>dhcpLease_expiry</td>
<td>Expiry date of DHCP lease</td>
<td>2015-09-10 T21:56:31.000Z</td>
</tr>
</tbody>
</table>
CHAPTER 4
CONCEPT AND DESIGN

This section is dedicated to the core concepts and design of visualization that is adopted in the creation of the dashboard. First few sections describe color schemes and representation of visual elements. Next section explains the type of chart followed by interactive components and their flow.

4.1 Color Scheme

Using the perceptual dimensions of color to organize the representation of data is vital in gaining useful insights through visualization [25]. Color can be picked based on three primary dimensions or properties i.e. hue, lightness and saturation. Hue is a perceptual dimension normally associated with primary colors Red, Green, Blue and Yellow, though there is another set of hues (Magenta, Cyan and Yellow) mainly used in printing technology; all the other colors can be derived using these set of primary colors or hues. Lightness is a perceptual dimension associated with brightness or is the amount of light reflected by an object compared to white. On the other hand, saturation is the measure of contrast or vividness in the color. Netflow data visualization involves categorical values that references horizontal axis for time. To increase the uniqueness across categories and similarity along time, qualitative and binary color schemes combined with varying levels of color saturation have been employed. Here brightness and saturation is same across categories (equal importance has to be given to each category) but may differ across time (this happens automatically when two categorical time elements overlap each other). Uniqueness is maximized by picking
different hues for different categories Figure 4.1 explains this analogy between color combination and saturation for categorical information along time.

Figure 4.1. Two different categorical information vertically with five different overlapping time elements horizontally.

Figure 4.2. Four similar events occurring concurrently .

Above figure 4.2 shows when similar events occur concurrently, saturation of hues will be different at sections where data points overlap. Upon careful observation we find the depth of hue increases when the intervals overlap each other, hence busy areas can be defined by the selected regions with deeper intensity or depth in color signifying concurrently occurring events. Color schemes in D3 is achieved using selective arrays of hues based on the schemes mentioned by Cynthia Brewer. D3 Category20 color scheme has been employed to visualize data points on this note.
4.2 Exploring types of chart

In this section we explore types of chart that are most suitable for data presentation. Graphical perception of discrete time-series data depends on few critical factors which are listed below:

1. Managing visual space: this factor emphasizes the efficient usage of screen space to visualize discrete time-series dataset by 'sharing' or 'splitting'. As explained by Javad et al. [26] shared space is typically more amenable to comparison between series, while data in split space may be easier to perceive. In shared space single baseline (normally referred as horizontal axis) is shared between the visual elements, while in split space every visual element requires a separate baseline.

2. Configurability: the choice of visualization model depends on the way device data is retrieved or the functions that help them retrieve. Change in device data should result in visualization with minimal effort.

3. Size of time intervals: data-points can be viewed as discrete time-intervals and duration of such intervals could be in fractions of seconds. In contrast there could also be intervals which span is in hours, in such situations larger intervals would overshadow intervals that are relatively much smaller.

4. Con-current intervals: data-points could represent concurrent time intervals which could result in overlapping points thus visualization would require techniques to capture this behavior.

5. Number of ordinal attributes using the vertical visual space: due to limited screen space ordinal points in the vertical space has to be efficiently managed.

6. Viewer’s comfort: main focus of presentation of data is the audience viewing the visualization. It should guide the end-user or viewer through-out the process of information discovery.
7. Chart junk: the word was first explained by Tufte [14, 27]. It is the additional ornamentation of chart which does not necessarily yield any benefit or help in comprehending the message to be delivered. Attempt of visualizing the data in such orchestrated forms of chart should be discouraged.

Factors described above are important in realization of charts that are key components in the data exploration process. These factors will also narrow down the search for the most relevant chart for the given dataset. Though considering all the factors in a single type of visualization is difficult, for this reason chart has to be interactive and should give ability to viewers to scale and zoom, these parts will be explained in later sections of this chapter. Further to make our search of charts more concrete we pose few questions that are: what are the existing charts to present time-series data? How do they fit our need based on factors of effective visualization? And what is the outcome or exact interpretation of data through the medium of chart?

Charts that explore time series data can be represented in many ways. Among all, the most common way is using a line chart. Line chart was first introduced by William Playfair and was used for representing time-series data [26, 12]. Path of this line is decided by the horizontal axis (which is referenced for time, normally in increasing order towards right end of the axis) and vertical axis (which is reference for some numerical value of a quantity). Multiple such lines constitute a multi-line time series chart; such charts are used in wide range of fields like science, finance, business, statistics etc. Figure 4.3 shows an example of line chart [1] and Figure 4.4 shows an example of multiple line chart [2].
Multi-line time chart shown above uses a third ordinal dimension as cities, represented in different colors. Such types of charts are good for representing data which continuously fluctuate with time and cannot be used to present discrete time intervals. Another drawback of this chart is if the number of ordinal value increases the chart becomes clumsy and cluttered.

As mentioned in section 2.2 (chapter 2) Netflow dataset represents discrete time events, the need of visualization is to present these discrete time events for each
category (meaning of these attributes is elaborately explained in chapter 2) and from the end-user perspective these categorized events can be understood as discrete user or application driven events in time. Clearly in the above mentioned example of multi-line time series chart Netflow data cannot be visualized, since line can only highlight the length of an event but cannot present events which are instantaneous and further it also cannot highlight busy event driven regions (when two or more events in the same category occur at the same time) within each category.

Qiang et al. [28] explains intervals with a triangular model, which is a two dimensional transformation of linear time intervals. In TM (triangular model) each interval is derived from an intersection point of two lines; which extends from the start and end of an interval in linear space. Intervals are transformed in two dimensional space such that the duration (difference of end and start of an interval) in triangular model is proportional to the duration in linear space. The visualization emphasizes on various selections (like intervals between a range, overlapping intervals, intervals outside a selected range etc.) that can be applied on intervals and good for homogeneous collection of temporal data that works well even when the dataset is large.

Figure 4.5. Transformation of intervals into triangular models.
However, presenting categorical time events as TM will not result in effective (considering the factor of viewer’s comfort) visual, part of the reason is the complexity involved which requires additional effort to comprehend the chart and configurability of the model to accommodate categorical information.

Bread et al [29]. Describes a concept of event band, an event band consists of several small events in time pertaining a theme, such themes could be temperature, pressure, low or high tide or any categorical values (that explains the context or meaning of data) that are kept constant with respect to time. By keeping such categorical values constant it becomes easier to comprehend events in the same context or theme. The model also accommodates multiple themes that can be analyzed with the help of event band. Fig 4.6 shows an example, this model primarily aims at ocean observing systems and data being generated from sensing systems like GoMOOS (Gulf of Maine Ocean Observing System). The event band in the example below illustrates a theme as a group of dimensions i.e. low pressure, location and year, which are fixed with events occurring between a time range, here each bar is an event (note few gray bars that represent no-data or data missed by the sensor). This model manages the visual space by splitting it such that each event band has its own baseline or reference axis thus requires more screen space to present the information.

Figure 4.6. Illustration of an event-band with various attribute grouped as a theme, the extent of the time-line is the fixed year in the theme.
Model as explained above can be applied to variety of other domains and can be extended to Netflow dataset as well. Advantage of such visual model is expressing temporal data with different dimensions as a theme. In our process of visualizing Netflow data we can extend theme to be as category or activity (one of the dimension in the dataset). Events with same categorical value contribute to event-band that represents activity.

Figure 4.7. Illustration of multiple event bands for three category values (Social media, Email and Torrent).

Figure 4.7 shows a chart (concept of event bands extended to Netflow) with a theme i.e. category (dimension or attribute from the schema of Netflow dataset). This chart shares the baseline or axis among event bands; each event band represents a categorical value and contains several events in time. The length of each event in the band is the duration and is represented with same color with in the band (color differs across the bands since each of them represent a categorical value). Range of bands in the vertical axis of the chart is limited by the extent of categorical values or in other words one band for every categorical value in vertical axis. Most of the visual factors mentioned in the beginning of this section are met by this kind of visual
layout; minimizing visual space (by sharing the baseline among all the bands) and a simple layout where vertical axis is for categorical values and horizontal axis for time (simple configuration of chart). Though this layout is good for representing large number of events we still need zooming and scaling of each band such that it gives the viewer a closer look and make events (which are fractions of seconds in length) clearly visible. Overlapping events in the layout is reflected by the intensity of colored regions as explained in section 4.1. Figure 4.8 illustrates an example of overlapped events.

![Diagram of multiple event bands for three category values](image)

Figure 4.8. Illustration of multiple event bands for three category values (Social media, Email and Torrent).

Other forms of chart used in the dashboard are the simple bar chart and pie chart to show trends in user activity and daily device summary respectively for a day.

4.3 Interactive components

1. Zoom and Scale: allowing chart to be zoomed or scaled enables the viewer of the chart to have a deeper insight of the data, in the case of visualizing Netflow data this component is a critical part of the chart. The main advantage in providing
such functionality is that user can select the ranges of time and analyze how events are dispersed. Additional functionalities can be added such as rolling up of data (important questions in which the user might be interested is mostly solved by aggregating data in different time ranges more on this will be explained in chapter 5).

2. Controls for navigation: controls such as selection box and buttons are added to the dashboard such that user could switch to different chart to view different trends, can select devices to view daily trends and hover over visual elements in the chart to view information.

3. Tool Tip: this component displays the information of visual element on mouse event (such components are added using the on mouse over and mouse out events provided by D3)

4.4 Architecture of dashboard

This section explains various processes involved in visualizing data through dashboard.

![Diagram](image)

Figure 4.9. Illustrates various stages of data in the process of visualization.
Above are the four levels or stages in the dashboard before the data is visualized, refer figure 4.9:

1. Load: this is the first level when the page is loaded along with the data store (here we use tsv file as input).

2. Pre-process data: This is the second stage where we parse the dataset line-by-line, all the preprocessing is done in this stage, such as converting string types to date objects and grouping data points based on device.

3. Transform: in this level data is transformed based on the type of layout to be displayed. Such transformations are event driven and will be triggered by end-users with the help of controls in the dashboard. In this level data is aggregated based on what chart is being currently viewed. D3 provides dispatcher which listens for user driven events and handles the events by informing the registered handler for respective events.

4. Visualize: this is the final level where data is presented in the form of chart elements. Charts in this level are interactive and also allow the users to zoom and pane the data (when such interactions happen internally the data in the selected range is transformed and the chart is updated or refreshed).

Figure 4.10. Internal communication architecture of the dashboard.
Above figure 4.10 explains the internal architecture of the dashboard. Control collectively represents all the elements in the dashboard that help in navigating through the view, users can select different types of chart and select different available devices to view their trends; these objects are responsible for making the chart interactive. Each control in the dashboard is registered with the dispatcher (listener) such that if an event takes place appropriate handler registered for the event handles the request. In turn charts view specific abstractions of data which helps user in finding trends, for this reason dispatcher is also responsible for transforming the data for specific view.

Figure 4.11 and 4.12 illustrates the layout of the dashboard and the types of charts used for presenting data. Control panel contains all the control objects; brush (zoom and pane control) is a special control which helps in zooming the chart and allows the user to select a range. Canvas is the part of dashboard where charts are displayed, these charts can be changed with the help of controls provided in the layout of the dashboard. The dashboard primarily uses three different layouts of chart i.e. event chart for visualizing categorical events of the selected device, bar charts to compare the categorical trends for each device and combination of bar chart and pie chart to display daily device usage for online activities.
Figure 4.11. Depicts a simple wireframe of the dashboard.

Figure 4.12. Different layouts of chart for visualization focused towards user-centric questions.
4.4.0.1 User stories

The design of dashboard explained in this section primary focuses on user-centric questions such as how long was the device used in a particular category? What was the time when some event happened? What is the trend of data in a specified time range? Or how long the device was active online? The simple design of this interactive dashboard makes it possible to answer such questions. In the next chapter implementation of this design is explained.
CHAPTER 5
DATA AGGREGATION AND IMPLEMENTATION OF DASHBOARD

This chapter explains the implementation details of the dashboard. Toolkits and libraries that were used in this process were HTML5, CSS, bootstrap, JavaScript, D3.js and occasionally JQuery [30, 31, 32, 33, 34]. Bootstrap and HTML5 were mainly used for dashboard layout. Separate styles were maintained as CSS classes for the dashboard and charts. Most of the transformation achieved in the data was by creating reusable functions in JavaScript. D3.js is a widely popular open source library available for visualization tasks and easily integrates with JQuery and JavaScript; all the data driven visualization is achieved using this library.

This dashboard is a single page application [35]; it loads all the required resources on first load. First out of the four stages in the life cycle of data in the dashboard explained in chapter 4 in Figure 4.9 is when the page is loaded along with the data store (as explained before the data store in this case is a tsv file). Although there are effective libraries to create single page applications in JavaScript, such as ember.js, backbone.js, react.js etc., we only use JavaScript, Bootstrap and D3.js for the scale and simplicity in implementation of the application.

5.1 Development Environment

AMPPS (Mac distribution) [36] is used for hosting the dashboard locally on the development machine. Main browser that has been used throughout the development phase is Chrome (for running the application) and its developer’s suit for debugging. Other browsers on which the application has been tested are Safari, Mozilla Firefox
and Internet Explorer. All the unit test cases are written and tested using QUnit version 1.20.0 [37].

5.2 Aggregating Events

This section explains various aggregations on events based on different user stories explained in chapter 4 section 4.4.1. Aggregation is applied on device dataset to find different usage trends; such trends are presented to end-users in different charts explained in chapter 4 section 4.3. Aggregation is done during the data transformation stage explained in section 4.3 and is based on type of chart and device selected or requested.

5.2.1 Categorical Usage of Device

To find the usage (online) of each device categorically and summarizing the trend requires, aggregating(summing) events after grouping them in categories (user activity). Applying simple aggregation on groups of such categorical events may lead to results exceeding the maximum daily usage (i.e. 24 hours) of a device. Moreover, if the user wants to know the categorical usage of a device in a specific time range the problem repeats simply by resulting in usage exceeding the selected time range (all these queries are interesting from the users point of view). This mainly happens due to the existence of concurrently occurring events, example of this could be receiving instant messages on more the one social networking application at the same time or downloading files from different torrent sources that may have started in the same instance; the dataset would have such flows or events classified under similar categories like social media and torrent. To make the above statement concrete we assume dataset of a single device and each event in the dataset to be represented as closed interval i.e. $I[s, t]$ where $d = t - s$. Here 's', 't' and 'd' are start-time, end-time and
duration respectively; in the dataset end time is not given, it is computed using the first seen timestamp and duration of the flow. If we consider \( W \) as a set of all intervals in a time range, where \( I \in W \) and \( X \) as a set of overlapping intervals in \( W \) then

\[
U = \sum_{i=1}^{n} W_i - \sum_{i=1}^{m} X_i
\]  

(5.1)

can be defined as usage or sum of all intervals without overlap in the same time range. To find categorical usage of a device overlapping intervals have to be removed, moreover intervals may overlap in various ways, below are the possible cases:

Figure 5.1. Illustrates various cases how two intervals may overlap, 1) \( b \) overlaps \( a \) from tail, 2) \( b \) overlaps \( a \) from head, 3) \( a \) and \( b \) super impose each other, 4) and 5) intervals do not overlap.

Figure 5.1 shows various scenarios when intervals \( a = [s, t] \) and \( b = [s', t'] \) overlap along with the cases when they do not overlap (case 4 and 5). Overlaps in an interval can be removed by either modifying the start-time or by modifying the
end-time of an interval i.e. if the overlap occurs in the tail (case 1 in the above figure) then the interval’s end-time occurring first (interval ‘a’ in case 1) has to be modified to end time of the overlapping interval (interval ‘b’ in case 1). Another problem in aggregating the intervals (grouped by category) is that data points(flows) or events in the dataset may not be in order and due which keeping track of overlapping intervals in a category becomes difficult.

Figure 5.2. Illustrates the order of intervals in the same category (colored blue) and resultant interval after removing the overlap (colored green).

Above Figure 5.2 depicts five intervals (that represent events from same category) in the order in which they appear in the dataset. To efficiently overcome these issues intervals can be sorted first and then scanned for overlaps.
The advantage of sorting (ordered based on start-time of intervals) the intervals is that overlaps can be removed in one pass of the list with fewer number of checks (not all the cases have to be checked illustrated in Figure 5.1), since intervals occurring later in the list cannot have start time less than the start time of the intervals occurring before in the list. Therefore, intervals can either overlap tails of intervals before them or superimpose them. Algorithm 1 below is for removing overlaps and computing usage in a category, input of this function is an unsorted list of events for a single device and the list of categories for which usage has to be computed.

The time complexity of the above algorithm is $O(n \log n)$, where ‘$n$’ is the number of events in the unsorted array, step 1 is the costliest since it involves sorting of events for the selected device.

5.2.2 Device usage

Device usage can be considered as the total time throughout the day when device was active online, this is computed in a similar way explained in section 5.2.1. Aggregation is done on the sorted list of events in the device after removing the
Algorithm 1 Algorithm categorical device usage

Input: eventList ← list of events in the device, categoryList list of categories for the device

Output: U collection of categorical usage of the device

Step I: Sort the eventList based on start time of the events, i.e. sortedEventList ← sort eventList (ascending order of time)

Step II: For categories in CategoryList

P (collection of sorted categorical events) ← Filter events from sortedEventList

Step III: Remove overlapping events for each category in collection P

Step IV: For each category in collection P

U (collection of categorical usage of device) ← find the sum of durations of events of category in collection P

overlaps (without considering categories). Algorithm 2 below is for computing total device usage. Time complexity of this algorithm is same as categorical device usage.

5.2.3 Aggregating other attributes of events

events(flow) in the dataset (NetFlow dataset) are multidimensional and can be used to answer other interesting trends such as daily data usage (upload or download) of device based on category or without category. "Categorical device data usage" and "device data usage" is computed in a very similar way already explained in previous sections, only modification here is when removing the overlapped events, attribute values for such events are also aggregated; at this level aggregation depends on the type of value for the attribute, in case of string type values it is aggregated by appending values as a list and in case of number type values it is aggregated by summing the values.
Algorithm 2 Device usage

*Input:* eventList $\leftarrow$ list of events in the device

*Output:* $U \leftarrow$ device usage

Step I: Sort the eventList based on start time of the events, i.e. sortedEventList $\leftarrow$ sorted eventList (ascending order of time)

Step II: Remove overlapping events from sortedEventList

$P$ (collection of sorted categorical events) $\leftarrow$ Filter events from sortedEventList

Step III: $U$ (usage of device) $\leftarrow$ find the sum of durations of events in sortedEventList

5.2.4 Aggregating events on user selected ranges

Figure 5.4 illustrates a scenario when information is required in a selected time range.

Figure 5.4. Illustration above is for a selected time range, intervals which are out of bounds are clipped or excluded during aggregation.
Events which are in the bounded region are considered for aggregation and are limited to values of attributes of event which are not string type or fixed throughout the duration of event (like port number, destination and source address etc.). In case of data usage, it is considered uniform throughout the length of the interval and percentage in the bounded region is accounted towards the computation. Process after this is similar to the aggregation explained in previous sections.

5.3 Visualization

This is the final section of this chapter and explains details of the dashboard which is been implemented. All the charts in the dashboard were implemented using D3.js. Figure 5.5 represents the dashboard with event map, different colors represent available categories for selected device, chart type and device can be selected from the selection menus and a button(hover) is provided for viewing event information (this button also works as a control to remove overlaps in event map, for other charts it only works as a control for tooltip). Each category in the device is represented as an event band in the event map and is referenced by the horizontal time axis (baseline). The axis is presented in the form of ticks to reduce ink and screen color. Panel below the event map is provided for zoom and focus(brush) and can be used to view the event activities in a specific time range. Initially when the Netflow summary page(dashboard) and controls are loaded. Data is loaded in the memory(background) and preprocessed for the use in different charts. Events from the selection menu such as device selection and chart selection triggers dispatcher to transform data and call the respective handlers to present the chart. Handler specific to the selected chart uses the transformed data and D3 functions to render the chart presented in the dashboard; this flow is explained in detail in section 4.4.
Figure 5.5. Netflow summary dashboard representing event chart with overlapping events, time ranges where category colors are intense are the busiest regions.

Figure 5.8 depicts a scenario when hover button is selected to enable tooltip, this functionality is also provided to remove overlapping events from each category (intensity of color sets to lower value throughout the band indicating events without overlap).
Figure 5.6. Illustrates the anatomy of the dashboard.

Figure 5.7. Illustrates the components in the control panel of the dashboard.
Figure 5.8. Dashboard representing event chart when overlapping events are removed, color intensity is low for events and individual events are clearly visible, on hovering over each event details can be viewed about it.

Figure 5.9. Dashboard representing events chart when a time range is selected from the brush above the chart.
Figure 5.10. Dashboard representing, categorical usage in hours, vertical axis represents hours.

Figure 5.11. Dashboard representing, categorical usage in hours, vertical axis represents hours, sort button is provided to dynamically change the chart with sorted bar chart.
Figure 5.12. Dashboard representing categorical data usage of the selected device.

Figure 5.13. Dashboard representing categorical data usage of the selected device in selected time range from the brush above the chart.
Figure 5.14. Dashboard representing total usage of the selected device, bar represents the usage in hours for a day and donut represents percentage of upload and download data.
CHAPTER 6

CONCLUSION AND FUTURE WORK

This research started with the goal of explaining daily Internet activity of device owners using decorated flow summaries in such a way that trends in data could be interactively highlighted. In the process we developed user stories which were specifically aimed towards user centric questions. Representing such data visually posed several challenges such as event overlaps and computing various usages to provide activity trends. This thesis demonstrates a way of aggregating overlapping events in order to compute the usage in different user activities. The visual framework which is a proof-of-concept provides a solution by interactively presenting trends based on user demands and such need based presentation of data overcomes problems associated with available visual space and data size. The dashboard presents data in the form of interactive charts designed based on user stories, it effectively explains the trend with the help of an interactive event map and different usage charts. The layout and design of the dashboard inherits concrete ideas and design patterns from the field of information visualization which has become a de facto standard for presenting data visually.

In the future the dashboard can also be used to reflect monthly trends. Computation of usage explained in this thesis can be further extended to monthly reports and the event map can effectively highlight busy days with the same visual layout. The solution provided here can be further extended to small devices and platforms, such as smart phones and phablets (crossover of phone and tablet). This extension would be valuable noting that small devices are accessible almost everywhere and are
faster and portable means of communication. The framework will have to accommodate touch based interaction as well as point based (ideal for web-based applications meant for laptops and desktops) though Bootstrap is still a good option to build the dashboard layout since it supports small devices well. Finally, due to the limitations of the available visual space in small devices alternative design schemes need to be explored and adopted to replace pie charts or donut charts which are used for device summary.
REFERENCES


[26] W. Javed, B. McDonnel, and N. Elmqvist, “Graphical perception of multiple time
series,” IEEE Transactions on Visualization and Computer Graphics, vol. 16,
no. 6, pp. 927–934, 2010.

[27] S. Few and Perceptual Edge, “The chartjunk debate,” Visual Business Intelli-

[28] Y. Qiang, M. Delafontaine, M. Versichele, P. De Maeyer, and N. Van de
Weghe, “Interactive Analysis of Time Intervals in a Two-Dimensional Space,”
http://ivi.sagepub.com/content/11/4/255.short

tool for visualizing and exploring events extracted from Ocean Observing System


US/docs/Web/JavaScript


BIOGRAPHICAL STATEMENT

Lalit Kumar Naidu joined the University of Texas at Arlington in Fall 2013 for MS in Computer Science. He received his M.Sc in Software Engineering from Sathyabama University in 2010. He worked as an IT Service Delivery Consultant for 3 years in Hewlett Packard in India. In the US, he worked as a graduate teaching assistant at UTA and interned at Capitalogix. His current research interests are in the areas of data analysis or mining, big data technologies, data visualization, web analytics and software development.