

The Infinite Canvas: Analyzing Social Networks in Virtual Space

Falko Schulz, SAS Australia

Nikola Marković, Boemsa United Kingdom

ABSTRACT

Virtual Reality (VR) technology has been the next big thing for a few years now, mainly focused on the consumer market with gaming and video playback applications. More recently, better hardware and usability have started making VR increasingly attractive for professional applications. Especially in the world of big data and advanced analytics, having access to an immersive, infinite, easily navigated visualization space opens an entire new world of data exploration and workflow.

This paper showcases one such immersive workflow for inspecting social networks in the context of financial fraud investigations - one that takes advantage of SAS® Viya® to transparently interact with its advanced analytics and machine learning components to create an intuitive, interactive, and productive analyst experience. Buckle up and enjoy the ride!

INTRODUCTION

Inferring meaning from data sets of network data, especially larger ones, can be challenging. Unlike most data visualization, which generally deals with communicating scalar values, graphic representations of networked data must instead rely on communicating entities, the connections between entities, and the meaning of those connections. The best-known way to do this is to project and visualize this data in a graph layout, with scalar properties of the data represented as secondary attributes.

A challenge with this approach is that most of us are traditionally accustomed to comparing data points to a numbered axis. With networked data, there are no axes; the *meaning* depends on an entity's context, its surroundings. The number of relevant nodes that we might need to compute in one pass to notice or interpret meaning in the data can vary wildly depending on the context, with immediate locality being just as relevant as the bigger picture. Because of its relative nature, the data is also often consumed by traversing the network node by node, and perspective and projection of the same network data can affect its interpretation.

It is for these reasons that the immersive, infinite plane of the VR canvas might hold greater potential for improving our capacity to interpret and compute network data over traditional data. However, just throwing VR at a problem like this doesn't work. Some of the biggest challenges with effectively implementing VR technology, particularly for professional applications where a user might want to spend a prolonged amount of time in VR, have been around nausea, disorientation, and the general discomfort resulting from immersing the user in another reality. Building for VR involves more than just programming a 3-D



Figure 1 – Network Rendered in VR

visualization - an *experience* must be created that allows the user not just to tolerate the transition in and out of the data space but to actually overcome the cost of crossing that threshold, to the point where they are truly benefiting from the transition into the virtual space.

To create this experience and realize this reality, the authors of this paper collaborated with a team of acclaimed VR and video game developers with extensive experience in creating immersive, comfortable virtual realities. The result of that collaboration is frequently referenced throughout the rest of this paper.

VIRTUAL REALITY

Overview

Virtual reality headsets have become popular in recent years, fueled mostly by the computer gaming industry. In terms of hardware, the market is dominated by two players: HTC (makers of the HTC Vive) and Facebook-owned Oculus, makers of the Oculus Rift.

Both of those current-generation headsets depend on a desktop-class computer and graphics card, to which they are tethered via a wired connection. In addition to this connection, a set of sensors must be configured to track the VR headset and associated hand controllers in 3-D space, so that the user's physical movements can be accurately translated and re-rendered in the Virtual Reality plane. This is especially important for the headset itself because the image of VR projected to the user through the headset must sufficiently match the orientation and movement of a user's head to avoid feelings of nausea.

Due to these requirements, current-generation technology is considered expensive and remains inaccessible, largely confined to the gaming community. However, by the time this paper is published, the Oculus Quest - a fully independent, stand-alone VR headset capable of tracking 6 Degrees of Freedom (6DoF) movement inside-out, without requiring any extra sensors - will be available with a drastically lower entry price than existing options. The Quest is predicted to sell over a million units in its first year and is widely expected to bring VR headsets into the mainstream.

There are some compelling reasons why that new VR headset could potentially excel as an alternative interface for network data exploration:

- VR headsets are more compact than computer screens, at least when compared with screens capable of displaying comparable volumes of data. They don't require a desk or workstation, and in many ways they are ergonomically superior.
- Viewing data using a VR headset is inherently secure. It is incredibly difficult, if not impossible, to take screenshots of sensitive data viewed in VR or for a bystander to examine the user's shoulder - a valuable feature in environments or industries where confidentiality is paramount.
- While they might feel awkward initially, headsets offer a far more natural, intuitive interaction mechanism than the (comparatively ancient) mouse and keyboard most of us are used to. This is



Figure 2 - Oculus Quest



Figure 3 - VR Controller

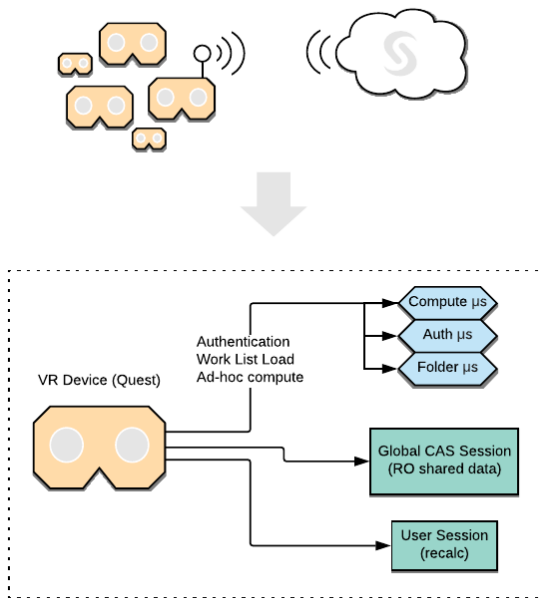
especially valuable in industries where the users might not have a strong technical background (for example, insurance fraud investigators, many of whom are former police officers and not necessarily experienced or comfortable with traditional user interfaces and peripherals).

- The headsets already operate by closely tracking the movement and behavior of the user. The ability to collect this data to profile, for example, the behavioral patterns of an experienced investigator provides an opportunity to encode their behavioral patterns by training machine learning-based models to emulate their decisioning behavior.

With these in mind, we set out to build an application that showcased the capabilities and open integration features of SAS® Viya® to take advantage of what could be the future of our interaction with data.

Application Architecture and Data Flow

The application is built using the Unity engine for front-end rendering and SAS Viya for everything else. SAS Viya manages authentication, data loads, layout calculations, and workflow/recalculation; and the Unity based application runs on the headset, acting as the display and UI layer.



Unity is a cross-platform 3-D engine that allows the same version of the front-end application to be portable across a variety of devices and operating systems, including both current-generation tethered VR headsets and the upcoming Oculus Quest. While it was built primarily for programming computer games, it has found a new role as one of the standards for VR/AR development.

The front-end application communicates with SAS® through RESTful API calls using built-in HTTP functionality. When the app starts, it connects directly to SAS Viya, authenticates, and then loads any data waiting for that user. The Unity app then renders the initial network and presents the user with their list of interesting parties or alerts, ready for processing.

Figure 4 - Communication Diagram

The Visualization Space

The space inside VR was designed with two ideals in mind: minimizing the discomfort experienced by the user and maximizing their capability to compute the data presented to them. Many of the features implemented specifically target the former; for example, the user cannot move within the space, but they can scale the data around them. There is no visible horizon, and the application explicitly doesn't try to fool the user into thinking that they are in the outside world. Instead, faint grid lines exist to ensure that the user remains oriented, while ensuring that the data-ink ratio remains high.

When a user puts the headset on, they find themselves in the center of a 20m³ (79 ft²) room, in front of a desk-height surface. The rendered data is presented floating above the middle of that surface as a network of nodes and edges. The user can then, with the hand controllers, grip the data, pick it up, hold it in their (virtual) hand, and inspect it from all sides as if it was a physical item they were holding in real life. By gripping the network with both hands, the user can easily scale the data up and down and quickly travel between different parts of the network without getting disoriented within the data. A trigger button allows the user to select nodes and bring up detail data or pin the node's card onto a virtual pinboard, either as a temporary work surface or as a destination for node categorization.

The position of the user within the room never changes; instead, they manipulate the data in the virtual space around them, through gripping, scaling, selecting, and siting. Due to the static nature, the experience works very well when the user is seated and exceptionally well in a reclining, swiveling office chair - which both opens up the overhead space and allows them to physically swivel around the data set they're viewing, an alternative to picking it up.

There are two additional features worth mentioning that lend themselves well to the technology but haven't yet been fully implemented.

The first is the use of voice recognition to aid in the interaction; the VR headset hardware is built with this in mind, and, because for most use cases the voice interaction will be used for fixed commands or for effectively selecting or locating nodes from a known set (rather than an open search), high accuracy and quality of experience should be achievable.

The second is collaborative data exploration. Because the user only ever scales the projected data, it is possible for two users to share the same relative space within the network, regardless of position or scale. A much-requested feature, this *might* be available by the time this paper is published.

SOCIAL NETWORK ANALYSIS

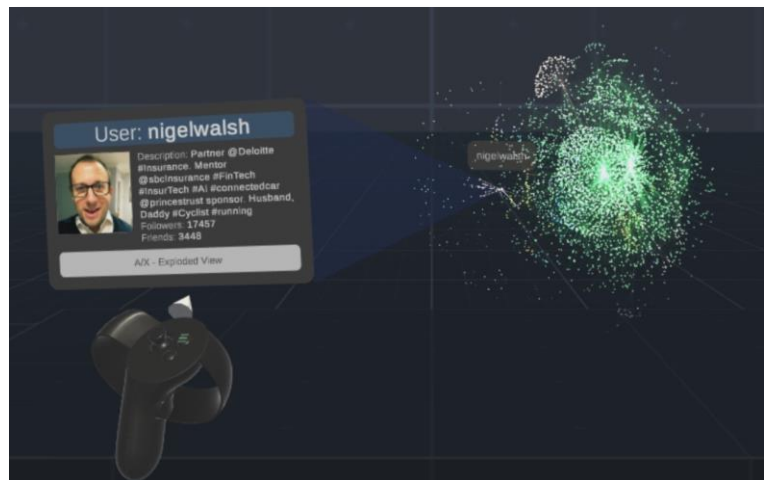


Figure 5 - Identity Card

Workflow

A peculiar aspect of complex systems, and one of the reasons they can be so difficult to predict, is the interplay between their structure and the behavior of their components (Schulz 2014).

This is particularly visible in social networks. Who you know affects what you do, and vice-versa. In this scenario, the network consists of actors – mostly people, but sometimes even automated agents or bots – linked by their relationships and the actions they enable.

The impact of what the actors do is largely a function of with whom they relate. Actors in critical places in the topology can affect the entire group in drastic ways.

So how do we go about understanding social networks? We begin by asking two basic questions:

- What is the underlying structure of the network? Do we have a single cohesive group or strong communities weakly linked by a few bridges?
- Who are the influencers? Who are the key actors in our network that trigger a group of suspicious transactions?

Together, these questions reveal how the macro and micro aspects of a network influence each other. Key actors shape the network structure (think about the hubs on a scale-free network), while communities define the scope of their influence.

Concepts

Analyzing networks and in particular social networks involves learning about some key concepts. Popular approaches include analyzing communities of people and how those are linked between each other along with key actor analysis.

Community Detection

Community detection, or clustering, is the process by which a network is partitioned into communities such that the links within community subgraphs are more densely connected than the links between communities.

The study of communities has great practical value as the nodes in these communities usually display common properties or have similar preferences – for example, the partition of the blogosphere along political lines] or gang membership among criminals (Dillow 2014).

Thus, identifying the communities in a network can further the understanding of how network function and topology affect each other. This has direct applications in marketing, sociology, and many other areas.

The following figure shows a snapshot of the financial network rendered in SAS® Visual Analytics® used for this paper. We can see a high number of identities connected by related bank transactions in the center as well as several small subclusters.

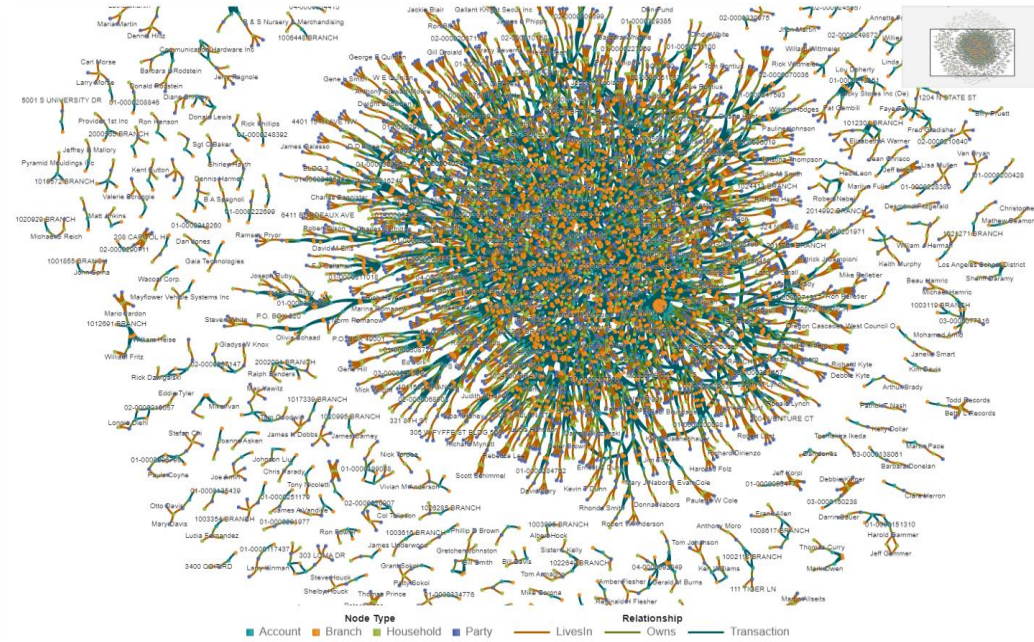


Figure 6 - SAS Visual Analytics - Fraud Network

Zooming in further reveals details such as the number of transactions and related weights (dollar amounts):

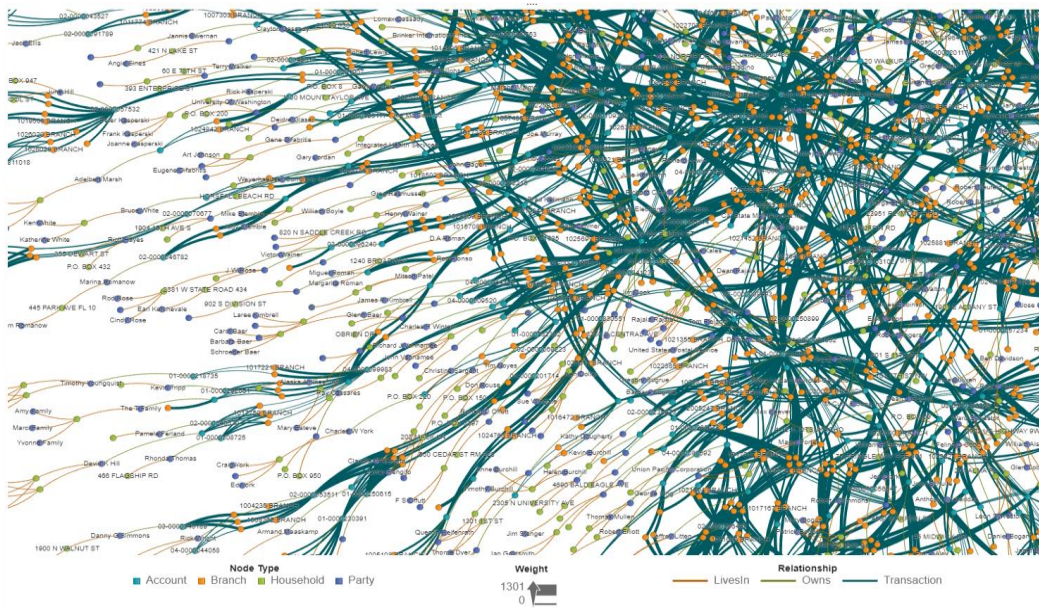


Figure 7 - SAS Visual Analytics - Subset Fraud Network

Such financial networks can be complex, and it is difficult to analyze the graph just by its structure. Let's dive in further to understand the concepts of social network analysis.

Key Actor Analysis

There are different ways to measure who is important, or central, to a network. Examples of such centrality measures include:

- *Betweenness* measures the number of shortest paths an actor is on, which indicates how often actors can reach each other through it. A high score indicates it is a likely path for information flows and might especially be interesting in a financial crime network, where such actors represent money flow.
- *Eigenvector* centrality is proportional to the centrality of an actor's neighbors. Google's PageRank algorithm is an example of this metric. A high score indicates the actor is popular among popular actors.
- *Degree* reflects how many actors are connected to a given actor. This is a simple metric that counts direct relationships (for example, the number of incoming/outgoing transactions).
- *Closeness* indicates the relative distance to all other actors. Closeness is based on the distance between actors, where distance is given by the shortest path between a pair of actors. A high score indicates the actor is close to everyone.
- *Influence* is a generalization of degree centrality that considers the link and node weights of adjacent nodes in addition to the link weights of nodes that are adjacent to adjacent nodes. This metric requires a simple traversal and therefore should scale to very large graphs.



Figure 8 - Property Filter Panel

Key Actor Analysis (Conway 2009) identifies critical nodes in a social network by plotting actors' scores for Eigenvector centrality versus Betweenness. Any actor with a high score on both measures is obviously an important node in the network. But given how these measures are expected to be approximately linear, any non-linear outliers are also considered to be key actors playing very specific roles.

An actor with high Betweenness but low Eigenvector centrality might provide the only path to a central actor. These are gate-keepers, connecting actors to a session of the network that would otherwise be isolated from the core.

On the other hand, an actor with low Betweenness but high Eigenvector centrality might have unique access to central actors. These are pulse-takers, well-connected actors at the core of the network.

The following figure shows a bubble plot to visually identify key actors. Highlighted is one of the pulse-takers that is linked to our financial network. It shows that this identity has unusually high numbers of transactions from a single bank account.

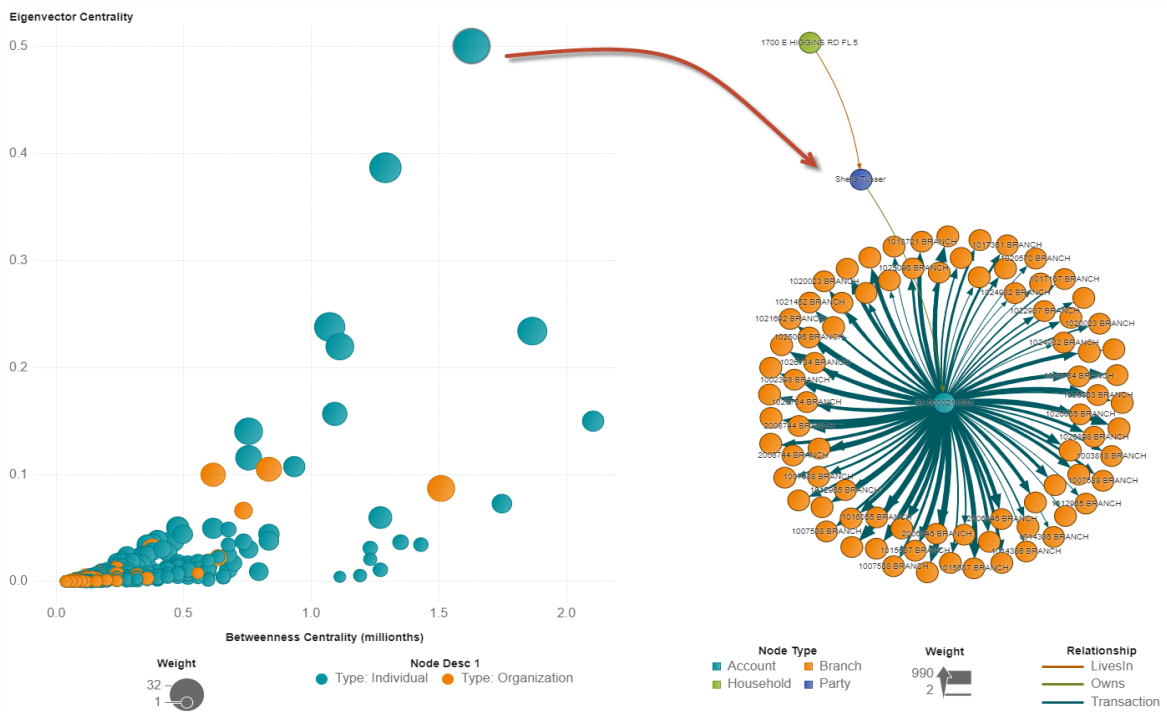


Figure 9 - Key Actor Analysis

Additional analysis typically performed for risk-based graph analysis includes shortest path analysis and cycle detection. Cycle detection is a powerful tool that helps identify when a node is indirectly related to itself through at least one other node. In financial crime, cycles can be used to detect things such as fraud rings, nexuses, or opportunities to layer illicit activities. Determining what cycle depth is suspicious requires feedback from actual investigations and analysis to define the optimal point of what cycle depth provides the most productive investigation (Overton 2018). Such tools differ slightly from previous pre-calculated network metrics in that they get triggered by the user and require one or more selected nodes. Support for both analysis tools is planned for future versions of the VR application.

SAS VIYA INTEGRATION

REST communication

For us to analyze data, the VR application needs to communicate to SAS Viya, in particular the compute engine SAS® Cloud Analytic Services (CAS). A popular communication technique uses REST-based interactions (SASViyaREST).

For demonstration purposes, we use simple HTTP requests using curl. When developing third-party applications, such calls are implemented and triggered from within the application. Before we can execute any specific data-related actions, a client has to authenticate and create a CAS session first.

While other authorization methods are supported, the following example shows a simple request by passing on the credentials as part of the request header. Such authorization would only be recommended for secure HTTPS communication.

```
curl -n --header "Authorization: Basic Y2FzOkdvNHRoc2Fz" -X POST
http://sasviya:8777/cas/sessions
```

The string `Y2FzOkdvNHRoc2Fz` represents the base64-encoded string of your credentials in the form of `username:password`. The result of such a call would return the CAS session ID, for example:

```
{ "session": "fea3fb43-4f85-3e45-afdc-cb2c1b88ac78" }
```

We will continue to use this session identifier as we execute new actions. A session can be closed by executing the following command, if required:

```
curl -n --header "Authorization: Basic Y2FzOkdvNHRoc2Fz" -X DELETE
http://sasviya:8777/cas/sessions/fea3fb43-4f85-3e45-afdc-cb2c1b88ac78
```

The VR application uses SAS Viya to calculate the underlying 3-D graph layout as well as related network metrics. The following section explains how related actions can be executed.

Calculating network metrics

The VR application uses a number of network metrics including all centrality measures as well as communities. Network metrics can be calculated using either the *Hypergroup* or *Network* CAS action. Because the Hypergroup action can give us both the metrics and the graph layout, we can use a single action call here. Centrality measures can be requested by specifying the argument `centality=true` and community detection is enabled using `community=true`.

Calculating the graph layout

For us to calculate a graph layout, we need to specify the input edge table along with the from/to columns. The actual structure of the call parameters is the same regardless of the input data. If you need to adjust or modify the graph layout, you will find many additional arguments in the related CAS action online documentation. Because our VR application uses a 3-D rendering engine, we also specify the option `threeD=true` to calculate the 3-D layout:

```
{
  "allGraphs": true,
  "createOut": "NEVER",
  "centrality": true,
  "scaleCentralities": "CENTRAL1",
  "community": true,
  "threeD": true,
  "graphPartition": true,
  "maxNodes": 100,
  "numericLabels": true,
  "inputs": [
    { "name": "from_id" },
    { "name": "to_id" }
  ],
  "table": { "name": "fraud_network", "casLib": "Public" },
  "edges": { "name": "fraud_network_edges", "replace": true },
  "vertices": { "name": "fraud_network_nodes", "replace": true }
}
```

Saving the request body into the calcNetworkLayout.json file allows us to execute the actual request:

```
curl -n
  --data "@calcNetworkLayout.json"
  --header "Authorization: Basic Y2FzOkdvNHRoc2Fz"
  --header "Content-Type: application/json"
  -X POST http://sasviya:8777/cas/sessions/[id]/actions/hyperGroup.hypergroup
```

The resulting tables contain the node positions (x/y/z) in our graph along with node and edge metrics. The VR application loads data using JSON and renders the graph. The following screenshot shows the same network graph as before but now rendered in VR:

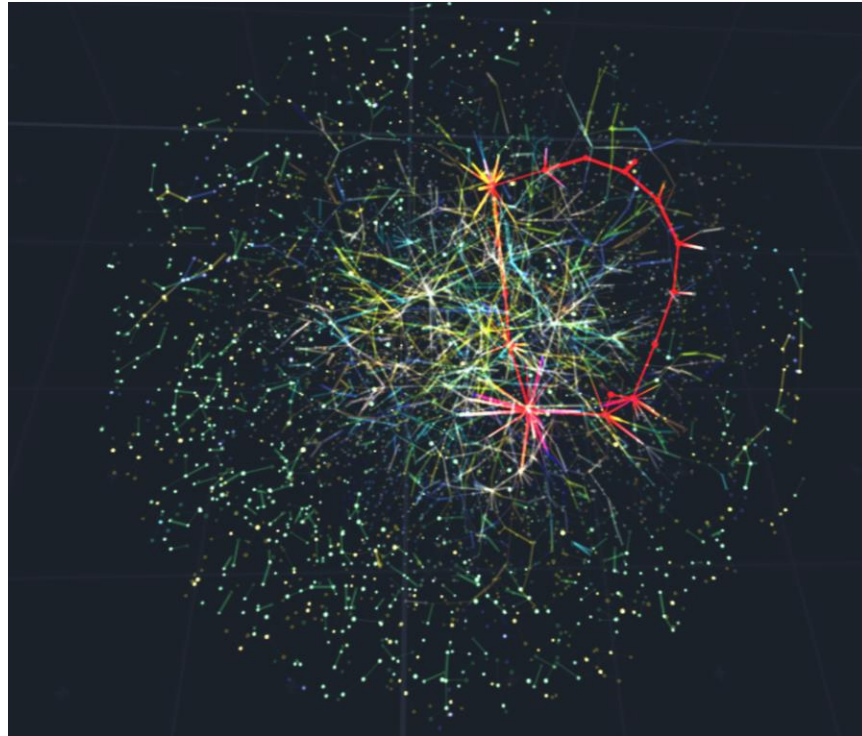


Figure 10 - Fraud Network Rendered in Virtual Reality

A complete list of supported arguments for the CAS Hypergroup actions can be found in the developer documentation (SASViyaHypergroupAPI).

USE CASE: FINANCIAL FRAUD INVESTIGATION

Overview

The previous section covers network analysis techniques in some detail in the context of social network analysis. Analysis of financial fraud data networks is very similar - the structure of the data is almost the same, and most of the same concepts apply. In fraud networks, rather than being linked by likes or shares, entities are linked by transactions, shared dwellings, bank accounts, or vendors.

Another difference when it comes to analyzing financial fraud networks is that the time taken to complete an analysis and make a decision can often be of critical importance - whether because of the immediate nature of the decision that needs to be made or because of the sheer volume of decisions that need to be made on a daily basis. The triage of alerts - the process of deciding whether an alert is worthy of investigation - is an example of a process that is both more workflow-oriented (in that it has discrete inputs and outputs) and more time-critical (due to the sheer number of alerts that need to be processed by analysts on a daily basis).

Virtual Reality Investigation Workflow

With fraud alert triage, each user is presented with a list of alerts that require confirmation or manual investigation - typically because the confidence of the model's decision is below a certain threshold. These are presented to the user on a pinboard - a virtual panel, positioned to the side of the user and just within reach, populated with a list of cards that represent alerts. The user can pick these cards up with the controller and be transported to the relevant part of the network, where they can replay or inspect the series of events (transactions) that led to a rule being triggered and an alert being created. The user can inspect the data available to them while narrating their thought process and ultimately pin the card to one of two panels on the opposite side of their workspace: a false alert pinboard or a pinboard to open an investigation.



Figure 11 - Card Holders in Virtual Reality

The advantage in this scenario is more in the features of the hardware than in the visualization: the ability to record voice narration as a record of the justification behind a decision is of significant benefit in regulated industries where a decision might be challenged down the line (that is, if a customer queries why their insurance policy was canceled), and the ability to use their hands to pick up cards from the fire hose, navigate to that part of the network, inspect with situational awareness, and make a decision is something that, although it takes training, becomes very obvious once a user is trained

and accustomed to working inside the virtual environment.

From a technical implementation perspective, behind the scenes the process looks something like this:

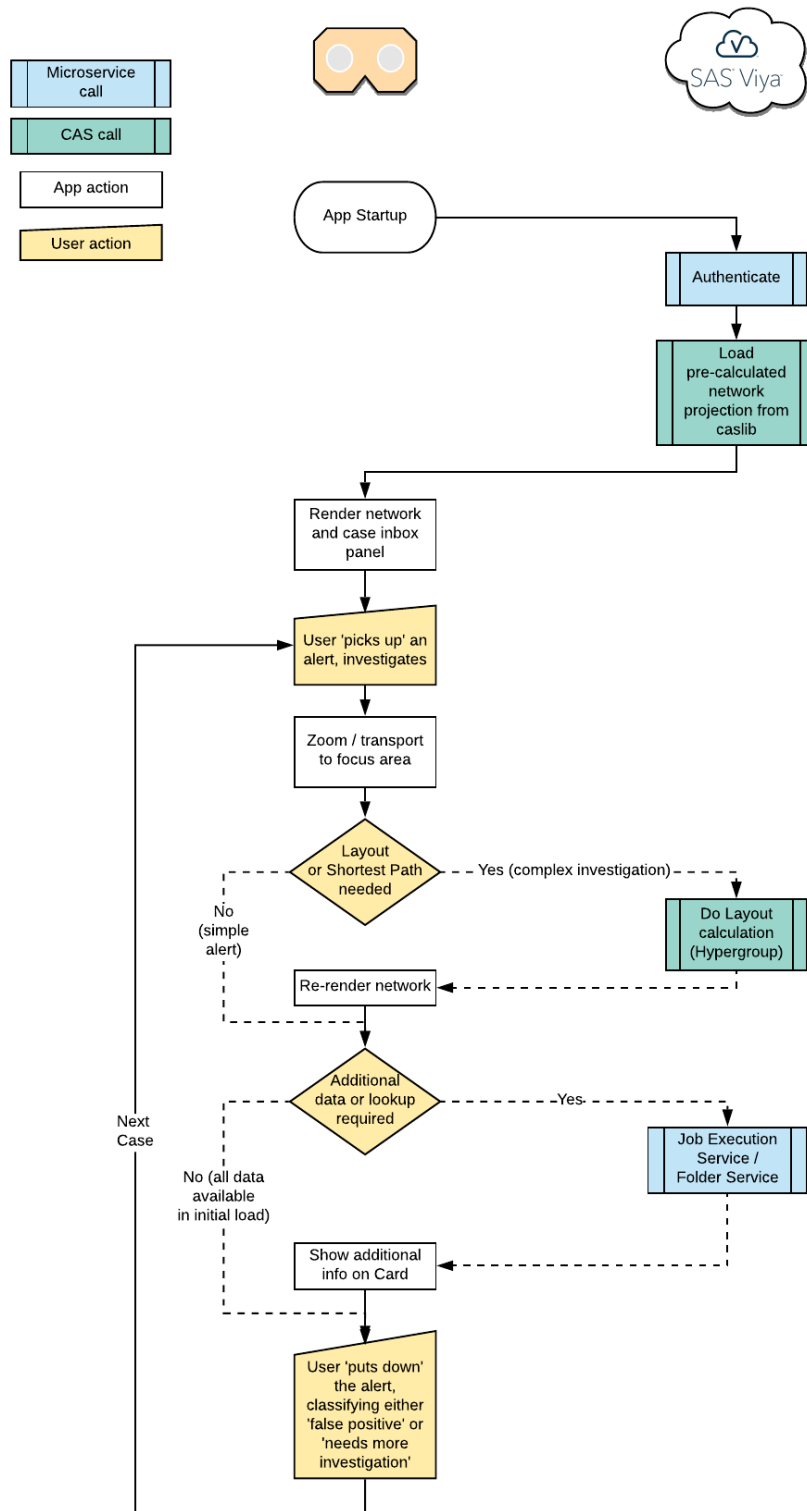


Figure 12 - Investigation Workflow

Another more general case, and one that warrants more exploration, is the ability to hand off a workflow to the VR headset from a full-featured desktop application when the user wants to explore a larger data set in the alternative space.

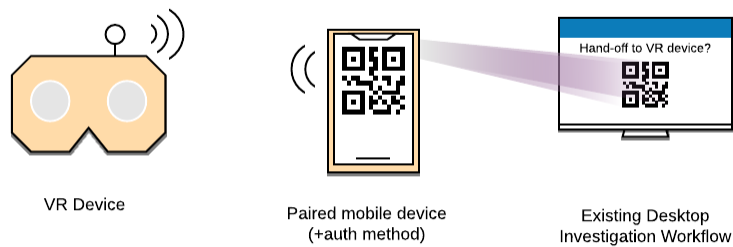


Figure 13 - Desktop Application Hand Off

The technical implementation of this functionality is currently planned to rely on session handoff using a QR code to expose a pre-authenticated endpoint, where the user scans their screen using their mobile device before putting the headset on. The current Oculus Quest device, much like its predecessor, the Go, is likely to lean on a paired mobile device for setting up the VR experience. The experience with this method should allow just enough time for the relevant data to be loaded from SAS Viya onto the device by the time the user puts it on, offering a completely seamless user experience. The user can tag nodes or mark certain bookmarks before putting the headset on and continuing with the rest of their investigation in VR. The virtual workspace can then contain the same sources and sinks that link it back to the original session, allowing the user to record their output from the exploration and get the best of both worlds.

CONCLUSION

Social network analysis is a complex undertaking, especially given the variety of graphs generated for industry-specific use cases. However, you are rewarded with knowledge about potentially hidden and previously unknown identity relationships - ultimately helping investigators to detect suspicious behavior and reduce the risk associated with financial crime. Experiencing graph analysis in such virtual space cannot only accelerate the investigation process and provide new insights into graph structure and underlying network metrics but offer drastic improvements in the way we interact with data, making it more accessible to a new generation of users. Powered by advanced analytics and machine learning algorithms using SAS Viya - this environment is suitable for both the non-technical, even technophobic, investigators as well as advanced data scientists.

REFERENCES

Conway, Drew. 2009. "Social Network Analysis in R." New York University. Department of Politics. Available http://files.meetup.com/1406240/sna_in_R.pdf. Accessed on March 18, 2019.

Dillow, Clay. 2014. "Building a Social Network of Crime - Can Software Distill Mayhem into a Database?" *Popular Science*. Available <http://www.popsci.com/article/science/building-social-network-crime>. Accessed on March 18, 2019.

Schulz, Falko. 2014. "Exploring Social Networks with SAS Visual Analytics." Available <https://blogs.sas.com/content/sascom/2014/02/19/exploring-social-networks-with-sas-visual-analytics>. Accessed on March 18, 2019.

Overton, Steve. 2018. "Discovering Insightful Relationships inside the Panama Papers Using SAS Visual Analytics." *Proceedings of the SAS Global Forum 2018 Conference*. Cary, NC: SAS Institute Inc. Available <https://www.sas.com/content/dam/SAS/support/en/sas-global-forum-proceedings/2018/1786-2018.pdf>. Accessed on March 18, 2019.

SAS Institute Inc. "Getting Started with SAS Viya REST APIs." Available <http://developer.sas.com/guides/rest.html>. Accessed on March 18, 2019.

SAS Institute Inc. "CAS Hypergroup Action API." Available <https://go.documentation.sas.com/?cdcId=pgmcdc&cdcVersion=8.11&docsetId=casanpg&docsetTarget=cas-hypergroup-hypergroup.htm&locale=en>. Accessed on March 18, 2019.

ACKNOWLEDGMENTS

This paper would not have been possible without the pioneering VR development work of Chris Cummings, San Shepherd and Olly Read. In addition, we would like to thank Steve Overton for his assistance regarding the underlying financial data mart and his critical role in performing fraud analysis, and Michael Thomas for his feedback on early prototypes and general support.

RECOMMENDED READING

- *Linked: The New Science of Networks*
- *Networks, Crowds, and Markets: Reasoning about a Highly Connected World*
- *Intelligent Realities For Workers Using Augmented Reality, Virtual Reality and Beyond (Michael D. Thomas)*

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the authors:

Falko Schulz
SAS Australia
Falko.Schulz@sas.com

Nikola Marković
Boemska
nik@boemskats.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.