ΠΑΝΕΠΙΣΤΗΜΙΟ ΜΑΚΕΔΟΝΙΑΣ
ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
ΤΜΗΜΑΤΟΣ ΕΦΑΡΜΟΣΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ

“Criminal Telecommunications Network Analysis – Key Player Approach”

Διπλωματική Εργασία

του

Αλετρά Δημητρίου

Θεσσαλονίκη, 01/2018

CRIMINAL TELECOMMUNICATIONS NETWORK ANALYSIS – KEY PLAYER APPROACH

Αλετράς Δημήτριος

Πτυχίο Οικονομικών Επιστημών – Πανεπιστήμιο Μακεδονίας - 2016

Διπλωματική Εργασία

υποβαλλόμενη για τη μερική εκπλήρωση των απαιτήσεων του

ΜΕΤΑΠΤΥΧΙΑΚΟΥ ΤΙΤΛΟΥ ΣΠΟΥΔΩΝ ΣΤΗΝ ΕΦΑΡΜΟΣΜΕΝΗ ΠΛΗΡΟΦΟΡΙΚΗ

Επιβλέπων Καθηγητής:
Χατζηγεωργίου Αλέξανδρος

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 23/10/2017

Ονοματεπώνυμο 1	Ονοματεπώνυμο 2	Ονοματεπώνυμο 3
....	....	....

Αλετράς Δημήτριος

....

Περίληψη

Σε αυτήν την διπλωματική εργασία περιγράφεται η λειτουργία και χρήση της δικτυακής εφαρμογής “IntelX” η οποία δημιουργήθηκε από τον συντάκτη και υλοποιεί ένα σύνολο αλγορίθμων για την αναγνώριση “παικτών κλειδιών” σε εγκληματικά δίκτυα τηλεπικοινωνιών. Οι αλγόριθμοι αναγνωρίζουν τους “παίκτες κλειδιά” με βάση δύο διαφορετικές προοπτικές, την προοπτική του κατακερματισμού του δικτύου και την προοπτική της διαρροής πληροφοριών στο δίκτυο. Στην πρώτη, αυτό που ενδιαφέρει τις διωκτικές αρχές είναι το να προκαλέσουν “ζημιά” στο δίκτυο, καθιστώντας το δυσλειτουργικό ή ακόμη και ανενεργό για κάποιο χρονικό διάστημα. Στην δεύτερη, αναζητούνται οι “παίκτες” του δικτύου οι οποίοι είναι οι καταλληλότεροι για να τεθούν υπό παρακολούθηση ή οι παίκτες στους οποίους να μπορούν να διαρρεύσουν σκόπιμα οι διωκτικές αρχές ψευδείς πληροφορίες έτσι ώστε αυτές να διαχυθούν γρήγορα στο υπόλοιπο δίκτυο. Η εφαρμογή αυτή μπορεί να είναι το έναυσμα για την υλοποίηση σύγχρονου λογισμικού το οποίο θα υλοποιεί στατιστικές μεθόδους για την αναγνώριση μοντέλων, προτύπων και συμπεριφορών, και τελικά να βοηθήσει στην πρόληψη και καταπολέμηση του εγκλήματος.

Λέξεις Κλειδιά: social network analysis, key player, criminal networks, enforcement priorities, intelligence priorities, network targeting, centrality, cohesion, fragmentation

Abstract

This master thesis, describes the use and design of the IntelX Web Application, which was developed by the author to implement various algorithms on criminal telecommunication networks in order to identify key players in the network, based on two concepts. The first being the key player’s ability to disrupt the network upon his “removal”, and the other being the key player’s ability to diffuse information to the network or possess a lot of information about the rest of the network. This application can serve as a stepping stone in an effort to develop modern crime fighting software that incorporates statistical methods that can be used to trace suspicious communication patterns, key-players and maybe help the authorities figure out more complicated networks and fight or predict crime more effectively.

Key Words: social network analysis, key player, criminal networks, enforcement priorities, intelligence priorities, network targeting, centrality, cohesion, fragmentation

Contents

Figures Index

Tables Index

1 Introduction

1.1 Problem – Importance of the Study

Crime Analysis is probably as old as crime itself. It involves data correlation, pattern recognition, analysis and description of these patterns and eventually a recommendation regarding measures to be taken to fight crime. The general rules may be the same, but science and technology have come to offer a great deal of help. Modern software assists with the representation of complex criminal networks that may be comprised of hundreds of suspects. Statistical analysis paints us the bigger picture concerning criminal activity and where it occurs more frequently, and huge databases contain related data from all over the world. And as much as can be said about crime solving, it has been the analyst's goal to develop methods that can detect or even prevent crime from happening. Therefore, algorithms have been developed to detect criminal networks or maybe suggest which network actors play which part. Modern software that implements such algorithms is exclusive to the authorities that are willing to pay hundreds of thousands of dollars/euros, and quite often it implements only the most basic algorithms that only calculate simple measures (usually common centrality measures). It is important to try to develop "smart" and modern software that can help, even by a little, law authorities in their struggle against crime.

1.2 Purpose - Goals

The purpose of the "IntelX" web application is to pinpoint the key players following two different approaches. First, we would like to identify which target would be more suitable for apprehension if we suspect an imminent threat, like for instance a terror attack. We do that by identifying the actor that, upon removal, disrupts/fragments the network the most. Secondly, we would like to identify which target would be more suitable to diffuse information into the network.

Both approaches need new measures since the out of the shelf centrality measures are inadequate. The "IntelX" application uses the measures introduced by Stephen P. Borgatti, and further investigated by Daniel M. Schwartz and Tony (D. A.) Rouselle.

1.3 Structure

In Chapter 2, the known for social network analysis centrality measures are discussed (Degree, Closeness, Betweenness, EigenVector) as well as the devised from

Stephen P. Borgatti and Daniel M. Schwartz and Tony (D.A.) Rouselle measures that identify key players or sets of players in a network based on two different concepts; Enforcement Analysis and Intelligence Analysis.

In Chapter 3, the design and use of the IntelX web application is discussed. In specific, the general database design, the importing and exporting functions, querying the database as well as visualizing the database's data as a graph are discussed. Finally, the analysis methods and their results presentation are presented.

Chapter 4 contains this thesis conclusions and limitations, as well as future implementations planned to make the IntelX application a complete tool for criminal telecommunication networks analysis.

2 Social Network Analysis in Crime Fighting

Social Network Analysis (SNA) is useful in Organized Crime or even Terrorist network's mapping as it focuses on the fluid interactions among the network's participants (actors/nodes) based on empirical data, avoiding the theoretical preconceptions about the way these networks operate. It is fairly complex however to achieve a successful analysis, as it requires good intelligence (information), high technical expertise, a good grasp of the analysis measures in play and a possible cooperation between law enforcement agencies and data analysts.

It is important to identify measures that can be helpful when trying to identify key players within a criminal network. To do that, we need to look into social network theory and examine which centrality measures help and in which way.

2.1 Centrality Measures

Social Network analysis measures are a vital tool in understanding the way our actors work and how they connect with each other. To investigate these connections we will use graph theory and look into some centrality measures used to filter the abundance of data and give the analyst a clearer view of what his data sample has to say.

2.1.1 Degree Centrality

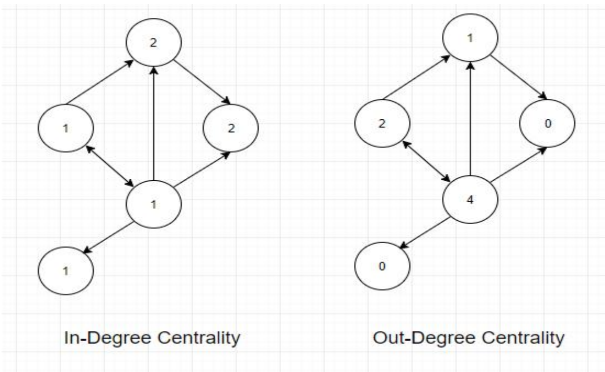


Figure 1

(Degree Centrality Measures)

Degree centrality indicates the number of connections a node possesses. In case we have a directed graph, we can investigate in-degree centrality and out-degree centrality. In-degree centrality measures the number of incoming edges onto a node while out-degree centrality measures the numbers of outgoing edges from a node. Depending on your type of data and whether your graph is directed or not you can either measure the degree centrality or explore the more detailed versions of in-degree and/or out-degree. Figure 1 illustrates an example of In-Degree and Out-Degree centrality measures. The centrality measure is marked with a number in the center of each node.

2.1.2 Betweenness Centrality

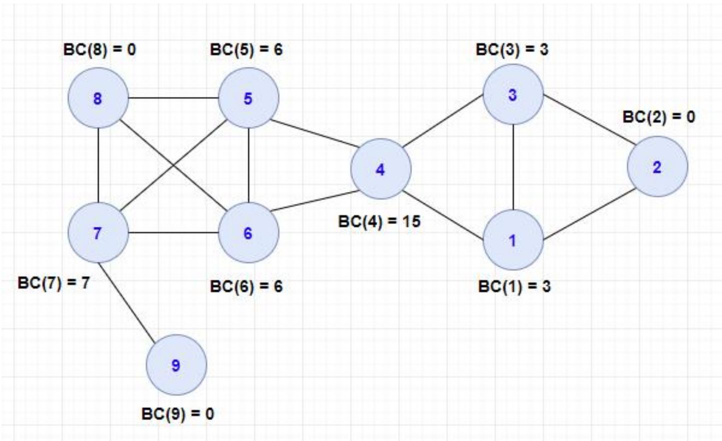


Figure 2

(Betweenness Centrality Measure)

Betweenness Centrality measures the number of times a node stands on the shortest path between other nodes (Freeman, 1977). Its calculation is not as straight forward as the degree centrality's, and is presented below:

$$g(u) = \sum_{s \neq u \neq t} \frac{\sigma_{st}(u)}{\sigma_{st}}$$

Where σ_{st} is the total number of shortest paths from node s to node t and $\sigma_{st}(u)$ is the number of those paths that pass through u. Figure 2 illustrates an example network with BC(i) (betweenness centrality) calculated for each node i. It is clear that even nodes with a lot of connections (high degree centrality) like node 8 for instance, can have a low betweenness centrality. Nodes that play the part of a "bridge" between larger groups of nodes are often called "brokers". The highest degree of betweenness centrality can be obtained in a star formatted network from the central node. In other words, to locate actors that connect larger groups of nodes, we pick the ones with the highest betweenness centrality measure.

2.1.3 Closeness Centrality

Closeness Centrality, as the name implies, measures the closeness between nodes. Closeness Centrality was introduced as a measure by Alex Bavelas (Bavelas, 1950) and it is calculated as the sum of the length of the shortest paths between a certain node and all others in a network (graph). It is calculated using the following function:

$C(x) = \frac{1}{\sum_y d(y,x)}$ where d(y,x) is the distance between nodes y and x. People often use the normalized version which is the average length of the shortest paths between two nodes x and y $C(x) = \frac{N-1}{\sum_y d(y,x)}$ where N denotes the number of nodes in the network.

For large networks the – 1 part becomes inconsequential so it is dropped resulting in $C(x) = \frac{N}{\sum_y d(y,x)}$.

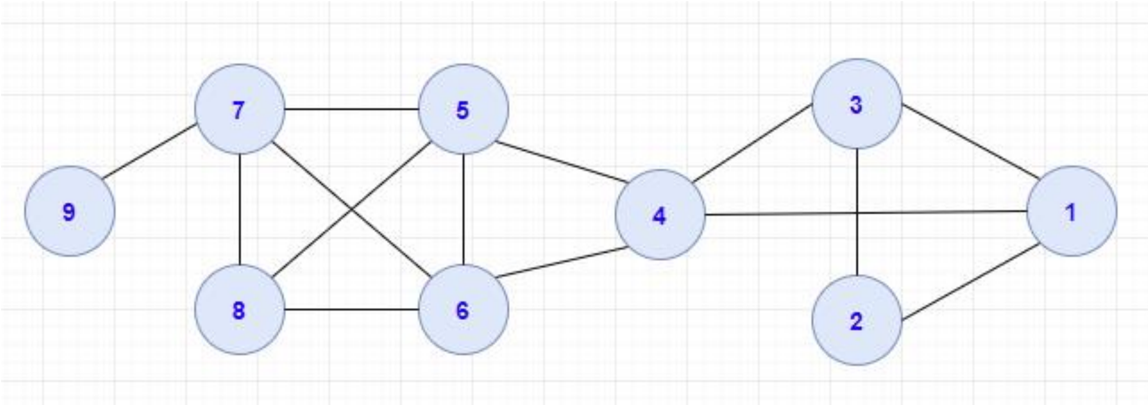


Figure 3

(Closeness Centrality Measure)

Figure 3 illustrates a simple network. To calculate the Closeness Centrality for node 3 for example:

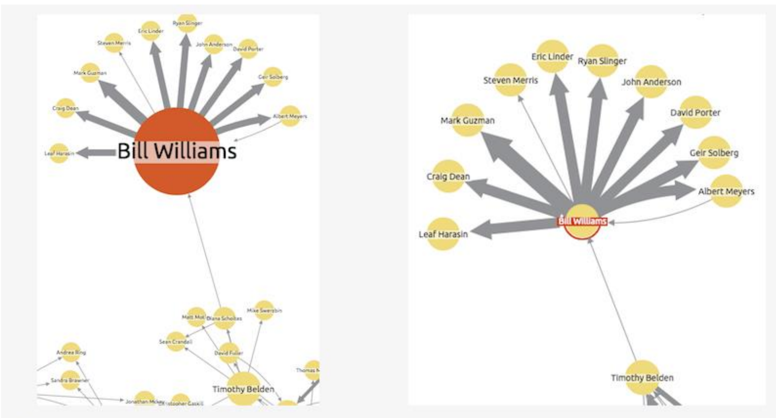
CC(3) = (9-1) / (1+1+1+2+2+3+3+4) = 8/17 = 0.47

CC(4) = (9-1) / (1+2+1+1+1+2+2+3) = 8/13 = 0.62

As it seems node 4 is "closer" to all the other nodes than node 3 since its closeness centrality is bigger.

2.1.4 Eigenvector Centrality

Eigenvector centrality, often called eigen centrality, is used to determine a nodes influence over the rest of the network (Newman, 2010). A high eigen centrality measure means that a node is connected with many nodes that also have high scores.



Degree centrality (left) and EigenCentrality (right)

Figure 4

(Eigenvector Centrality Measure from Cambridge Intelligence)

Eigen Centrality can be calculated using the following function:

$$X_u = \frac{1}{\lambda}\sum_{t\in M(u)} x_t = \frac{1}{\lambda}\sum_{t\in G} a_{u,t}x_t \ ,$$

where G is a given graph G:=(V, E) with |V| vertices and A = ($a_{u,t}$) is an adjacency matrix where $a_{u,t} = 1$ when vertex u is linked to t and $a_{u,t} = 0$ when they are not. M(u) is a set of all the neighbors of u and λ is a constant. Figure 4 illustrates an example of how Eigen centrality differs to degree centrality. On the left image, Bill Williams is marked as important because of his high degree centrality since he has many connections to other nodes. On the right image however, the same node is ranked sufficiently lower using the Eigen centrality measure, since he has only one tie with the rest of the network and that connecting node is not as closely connected as well (Cambridge Intelligence).

So, a node may have a high degree score (i.e. many connections) but a relatively low Eigen Centrality score if many of those connections are with similarly low-scored nodes. Also, a node may have a high betweenness score (indicating it connects disparate parts of a network) but a low Eigen Centrality score because it still has some distance from the centers of power in the network (Cambridge Intelligence). The combination of both a big number of connections and highly connected adjacent nodes is what awards the node with a high Eigen Centrality measure.

2.2 Borgatti's Measures

Stephen P. Borgatti, in his research titled "Identifying sets of key players in a social Network" (2006) tries to point out that the existing centrality measures are not always suitable for criminal network analysis and thus new, other than the off-the-shelf, centrality measures must be introduced. He introduced two separate concepts of key player analysis, one being the so called "enforcement" analysis and the other the "intelligence" analysis. He insisted that we should be looking for the reasons we need to use centrality measures before deciding which one to use (or create).

The first has to do with identifying the key player/s that upon removal, maximize the networks disruption. To form an example, suppose that a terrorist attack is imminent and there is a known terrorist cell in the country. This kind of analysis would point out the optimal assailants that by being removed (apprehended) they would damage their terrorist cell the most, leaving them unable to complete the imminent attack. The same algorithm could be applied in a drug trafficking ring, where by apprehending key players, the law authorities can render the entire network inactive for a small period of time at least.

The second analysis concept that Borgatti discusses, is the intelligence analysis which has to do with identifying the ideal graph actor that is used to diffuse information most effectively. Imagine that the crime fighting authorities need to learn the most about a known criminal network. Which actor is the best to put under surveillance? On the same time this actor would be the ideal candidate used to diffuse false information into the network.

2.2.1 Borgatti's Enforcement Analysis

In Borgatti's concept of enforcement analysis, he attempts to identify this "key player" by measuring the cohesiveness of the network before the targets removal and after, and trying to figure out which player's removal impacts the network's cohesion the most. The most suitable centrality measure for this purpose appears to be, at least at first, the betweenness measure. Betweenness measure sums the proportion of shortest paths from one node to another that pass through a given node (Freeman, 1979). Consequently, removing a node with high betweenness causes many pairs of nodes to disconnect from the network or at least stay connected through a longer path. Borgatti however, does not consider this measure to optimally locate the key player and illustrates the following example in Figure 5. In this example, node 1 has the highest centrality between all standard measures, including betweenness. However, the removal of node 1 has no effect on network disruption. Distances between nodes increase, but if our goal is the fragmentation of the network, removing node 1 is ineffective. If however, we remove node 8, we split the network in two smaller networks. Node 8 has lower centrality on all measures. It is clear then, that a measure that calculates fragmentation upon removal must be devised.

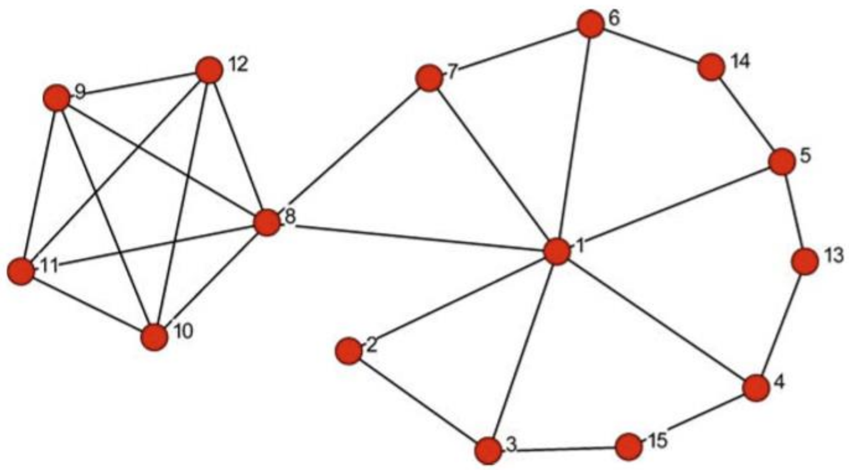


Figure 5

(Borgatti Enforcement Analysis Example)

Fragmenting the network however, is not enough. Dividing the network to two pieces for example is not the optimal solution because we would not take into account the size of the sub-networks, or the distance between nodes after the network is split. A measure that considers network fragmentation, sub-network size and lengthening the node distances is needed.

Borgatti came up with a suitable measure which is calculated by the following formula: $D_F = 1 - \frac{2\sum_{i>j}\frac{1}{d_{ij}}}{n(n-1)}$

Equation 1

Where: $1/d_{ij}$ → denotes a degree or reachability that varies from 0 to 1. In case where a node is disconnected from the others, the distance would be infinite so $1/d_{ij}$ is 0.
n → represents the number of nodes (Borgatti, 2006).

With this formula, both the fragmentation and the distances between nodes are taken into account. D_F takes the value of 1 when all nodes are isolates. This makes it clear that the higher D_F is the more fragmented our network. We can see the formula in action in Figure 6. Both cases (a and b) illustrate graphs consisted of two sub-networks of equal size. It is clear however that in case b, Borgatti's fragmentation is higher since the distances between node 1 and node 5, for instance, are longer.

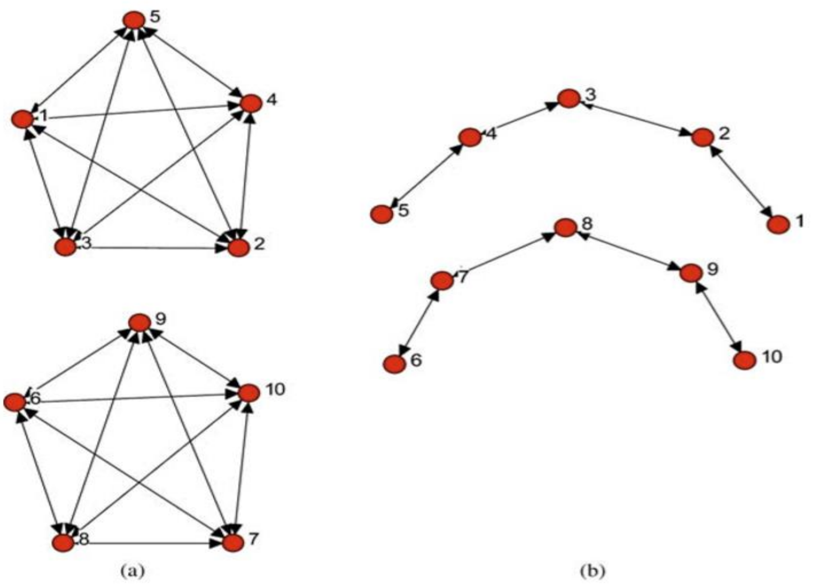


Fig. 5 (a) $^{D}F = 0.556$ and (b) $^{D}F = 0.715$

Figure 6

(Borgatti Enforcement Analysis Example)

2.2.2 Borgatti's Intelligence Analysis

As a starting point Stephen P. Borgatti defined a function that calculates the cohesion between members of one set of nodes (the kp set) and members of another.

$$C_K = \sum_{i \in K, j \in V-K} a_{ij}$$

In this function $a_{ij} = 1$ if the node i is adjacent to node j and $a_{ij}=0$ otherwise. K represents a set of nodes while (V – K) represents the rest of the network. As Borgatti notes however, this approach is essentially double counting ties to the same individuals, ignoring the structural equivalence of set members. For that reason he came up with the following function:

$$C_K = \sum_{j \in V-K} \bigcup_{i \in K} a_{ij}$$

In this function the operation U represents a non-specific aggregation function like taking the minimum or maximum. If U is the maximum function, then C_K is defined as the number of distinct nodes outside of K that members of K are adjacent to (Borgatti, 2006). This is identical to Everett and Borgatti's (1999b) notion of group degree centrality. But in order to incorporate the notion of distances as well, Borgatti followed a simple approach. He termed the m-reach measure, where he replaces adjacency with reachability such that ${}^{m}r_{ij} = 1$ if i can reach j via a path of length m or less, and ${}^{m}r_{ij} = 0$ otherwise (Borgatti, 2006). If the U operation is the maximum function, then the m-reach measure can be expressed like this:

$$C_K = \sum_{j \in V-K} \bigcup_{i \in K} m_{r_{ij}}$$

M-reach then, is a count of the number of unique nodes reached by any member of the kp-set in m links or less (Borgatti, 2006). The disadvantages of this approach are that

a) It assumes that all paths of length m or less are equally important (when the shorter length paths are most likely more important).
b) All paths longer than m are irrelevant.

The measure which Borgatti finally suggested regards all distances within the KP set to be unity, and let the summation occur over all nodes (Borgatti, 2006). He also normalized the measure to range between 0 and 1. The final measure is shown in Equation 2: D_R achieves the maximum value of 1 when every outside node is adjacent to at least one member of the kp-set while the minimum value of 0 is achieved when no kp-set member belongs to the same component as any node outside the kp-set (when the kp-set is isolated) (Borgatti, 2006).

$$D_R = \frac{\sum_j \frac{1}{d_{Kj}}}{n}$$

Equation 2

(Borgatti, 2006)

2.3 Daniel M. Schwartz and Tony (D.A.) Rouselle's Measures

While Borgatti's approach focuses solely on node positions, Schwartz and Rouselle's model incorporates the relative strength each node has, as well as the strength and importance of the node relationships. IntelX application incorporates this relationship and node strength by assigning a weight property on each node and edge.

According to Schwartz and Rouselle, Borgatti's approach to centrality focuses on the initial purpose of the measure itself, "why centrality is important in the first place" (Schwartz & Rouselle, 2009). Borgatti therefore shows that traditional centrality measures fail to capture the difference between enforcement and intelligence purposes. While Borgatti's approach recognizes this distinction however, it fails to incorporate weight attributes (Schwartz & Rouselle, 2009). It treats all nodes as equally strong and their relationship of equal importance.

2.3.1 Schwartz and Rouselle's Enforcement Analysis

To cover the factors that Borgatti did not with his methods, Schwartz and Rouselle developed their concept of Network Capital (NC). NC takes into account both the cohesiveness of the network and the actors and their between relationships strength.

$$Network\_Capital = \frac{Node\_Scores + Connection\_Scores}{N + [N(n-1)RSL]}$$

Where:

N denotes the total number of network actors;

RSL denotes the resource sharing level: an analyst-determined weight that reflects the percentage of resources that actors make available to other network actors and can vary between 0 and 1.0;

Node _ Scores denote the sum of each nodes weight;

Connection _ Scores denote the sum of the direct connection scores and the indirect connection scores;

(Schwartz & Rouselle, 2009)

The NodeScores variable is calculated using the following formula:

$$NodeScore_i = \frac{\sum_{n=1}^{NAW_i} AW_{ni}}{NAW_i}$$

Where **i** denotes a network actor, $\mathbf{AW_1}$, $\mathbf{AW_2}$, **…** , $\mathbf{AW_{NAW_i}}$ denote one or more attribute weights attributed to network actor i, each of which ranges from 0 through 1.0, $\mathbf{NAW_i}$ denotes the number of attribute weights attributed to actor I (Schwartz & Rouselle, 2009). Of note, Schwartz and Rouselle's approach allows for a usage of more than one weight attributes for each network actor. Using this method, Schwartz and Rouselle treat the network isolates in a really different way than Stephen P. Borgatti did, since an isolated actor with a high weight attribute can still be a candidate for being picked up as an enforcement target, although not being connected to any other actors. His "importance" might be so high regardless of the fact that he has no ties (not known at least) with any other network actors.

As noted above, a network actor can contribute to the Network Capital both with his presence and individual importance (NodeScore) and with his direct and indirect relationships with other actors (ConnectionScores). The reason why its connections matter is that an actor will never use his resources all by himself. There is a push and pull relationship between him and other actors as he might make his resources available for use by others and also "borrow" resources from others as well (Schwartz & Rouselle, 2009). This resource sharing however is bound by a theoretical and a practical factor. The practical factor has to do with how strong the ties are between two or more actors, thus how easy it is for them to exchange resources (link weight attribute). The theoretical factor has to do with the amount of resources that are made available for exchange by the actors. For this reason Schwartz and Rouselle used the RSL variable which ranges from 0 to 1.0. An RSL value of 0.1 equals a 10% of those resources being available. Let us consider two examples shown in Figure 7;

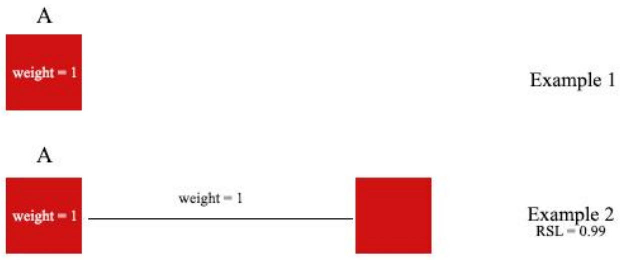


Figure 7

(Schwartz and Rouselle's Enforcement Analysis Example)

In Example 1, actor A contributes to the NC by adding his own importance of 1. In Example 2 however, he adds his own score of 1 and since RSL = 0.99, he adds a total score of 1.99 since 99% of his resources are used by the other actor as well (and their tie's weight = 1) (Schwartz & Rouselle, 2009). Network actors can be directly connected with one link or indirectly connected with more than one links. To calculate the direct connection score, we use Schwartz and Rouselle's formula:

$$DirectConnection_{ij} = \left(\left[\frac{\sum_{n=1}^{NAW_i} AW_{ni}}{NAW_i}\right] * RSL_i\right)\left(\frac{\sum_{m=1}^{NLW_{ij}} LW_{mij}}{NLW_{ij}}\right)$$

Where:

i and j denote network actors;

AW_1, AW_2, … , AW_{NAWi} denote one or more attribute weights attributed to network actor i, each of which ranges from above 0 to 1.0;

NAW_i denotes the number of attribute weights attributed to network actor i;

RSL_i denotes an analyst-determined weight for actor i ranged from above 0 to 1.0;

LW_1, LW_2, … , LW_{NLWij} denote one or more link weights attributed to the connection between network actors i and j, each of which varies from above 0 to 1.0.

NLW_{ij} denotes the number of link weights applied to the connection between network actors i and j.

(Schwartz & Rouselle, 2009)

Figure 8

(Schwartz and Rouselle's Enforcement Analysis Example)

To calculate the indirect connection score from A → C in Figure 8 we need to multiply the direct A → B connection score of 0.85 with the RSL value and the B → C link's weight. IndirectScore = 0.85 * RSL (=0.99) * 0.75 = 0.6311 (Schwartz & Rouselle, 2009). Notice that the more links between A and C, the lower their indirect score is, since it is multiplied more times with numbers smaller than 1. Furthermore, notice that the stronger the tie between B and C, the bigger the indirect score between A and C.

The formal notation for the indirect score is this: $IndirectConnection_{ij} = \max(p * q)$

Where **p** denotes a direct connection score between node i and j, **q** denotes the score for a series of one or more directly connected actors, beginning at actor l and ending at actor j, where each connection in the series is calculated by successively multiplying the previous score (p for the first such connection) by each RSL and link weight in the series (Schwartz & Rouselle, 2009).

2.3.2 Schwartz and Rouselle's Intelligence Analysis

Just like in enforcement concept, Schwartz and Rouselle suggest a measure called Intelligence Worth (IW). Intelligence Worth takes into account the nodes positions as well as intelligence weights on edges and attribute weights on nodes. Intelligence Worth is calculated pretty much the same way that Network Capital is. The attribute weights could represent the nodes intelligence gap. The less intelligence the authorities have, the higher the attribute weight for that node. Network Capital is network oriented while Intelligence Worth is node oriented. In the first, we are looking for the node that upon removal maximizes the network capital loss while with the latter we are looking for a node with the highest intelligence worth. Intelligence Worth is calculated with the following equation: $IntelligenceWorth_i = \frac{NodeScore_i + ConnectionScores_i}{NTA + [(N - NTA)RSL]}$.

Where **NTA** denotes the number of targeted nodes (the number of nodes targeted for surveillance), **i** denotes a network actor and **N** denotes the total number of nodes (Schwartz & Rouselle, 2009).

The equations used for this calculation are the same as in Network Capital calculation.

$$NodeScore_i = \frac{\sum_{n=1}^{NAW_i} AW_{ni}}{NSW_i}$$

Where **i** denotes a node, $\mathbf{AW_i}$ denotes one or more attribute weights and $\mathbf{NAW_i}$ denotes the number of attribute weights attributed to node i.

$$DirectConnection_{ij} = \left(\left[\frac{\sum_{n=1}^{NAW_i} AW_{ni}}{NAW_i}\right] * RSL_i\right)\left(\frac{\sum_{m=1}^{NLW_{ij}} LW_{mij}}{NLW_{ij}}\right)$$

Where:

i and j denote network actors;

$\mathbf{AW_1, AW_2, \ldots, AW_{NAWi}}$ denote one or more attribute weights attributed to network actor i, each of which ranges from above 0 to 1.0;

$\mathbf{NAW_i}$ denotes the number of attribute weights attributed to network actor i;

RSL_i denotes an analyst-determined weight for actor i ranged from above 0 to 1.0;
LW_1, LW_2, … , LW_{NLWij} denote one or more link weights attributed to the connection between network actors i and j, each of which varies from above 0 to 1.0.

$$IndirectConnection_{ij} = \max(p * q)$$

Where:
p denotes a direct connection score between node i and j;
q denotes the score for a series of one or more directly connected actors, beginning at actor l and ending at actor j, where each connection in the series is calculated by successively multiplying the previous score (p for the first such connection) by each RSL and link weight in the series (Schwartz & Rouselle, 2009).

The only difference between the calculation of Network Capital and Intelligence Worth lies in what the attribute weights symbolize. Schwartz and Rouselle tried to introduce a revised model based on Borgatti's concept of Enforcement and Intelligence Analysis. And while they managed to take into account the edges/links importance the attributed weights require too much knowledge about the network, knowledge that sometimes is what the law authorities are looking for, in the first place. Assigning weights requires deep insight into the network and some changes might trigger a need for massive reassignment of weights on the whole network.

3 The IntelX Web Application

3.1 Introduction

The IntelX application is a web application hosted online (http://addictivelabs.gr). Its front end is built with html + css and javascript, while its backend is built using php. The data is stored in a MySQL database. The files can be found on github here https://github.com/Add1ctiv3/MasterThesis. It is a network analysis and visualization project that implements centrality measures as well as some of Borgatti's and Schwartz and Rouselle's approaches. You can try the online version using these credentials:

Username: guest

Password: guest1!

The libraries used are:

- Vis.js (http://visjs.org), a library that visualizes data in the form of graphs, timeline etc.
- Jquery DurationPicker (https://github.com/Tartarus762/jquery-duration-picker), a smart looking jquery duration picker.
- Jquery ContextMenu (https://travis-ci.org/swisnl/jQuery-contextMenu), a smart looking jquery context menu.
- FileUpload (https://innostudio.de/fileuploader/), a really useful plugin that lets its user upload files through an html5 input element.
- jQRangeSlider (http://ghusse.github.io/jQRangeSlider/), a smart looking jquery range slider.
- Slim (http://slimimagecropper.com/), a beautiful jquery image cropping plugin.
- SlimScroll (http://rocha.la/jQuery-slimScroll), a jquery scrollbar plugin.
- TimePicker (http://trentrichardson.com/examples/timepicker/), a plugin that incorporates a timepicker in the jquery UI datepicker plugin.
- Toastr (https://github.com/CodeSeven/toastr), a plugin that shows beautiful notifications.
- PowerTip (https://stevenbenner.github.io/jquery-powertip/), a plugin that shows beautiful tooltips on html elements.
- Some general php classes that calculate the shortest path between two nodes using Dijkstra's algorithm (https://github.com/fisharebest/algorithm).

IntelX application is, for now, designed to visualize telecommunication networks only. The network's nodes will always be telephone numbers and the edges will always represent telecommunications (SMS or Telephone Calls).

3.2 Database Design

The MySQL database keeps stored telecommunications data, telephone numbers, datasets, people information (owners or users of telephone numbers), users, template settings and files. The most important tables are the ones that store telecommunication, telephone numbers and datasets data. Those tables can be seen below on Figure 9. Every telecommunication is associated with a dataset. Every telephone number is also associated with a dataset. As we can see below, the datasets table has a name field, a creator and editor field, a creation and edit timestamp field and last but not least, a visibility field. The visibility field can take either one of two values, public or private. If the value is private then the dataset and its contents are only visible to its creator, otherwise to everyone.

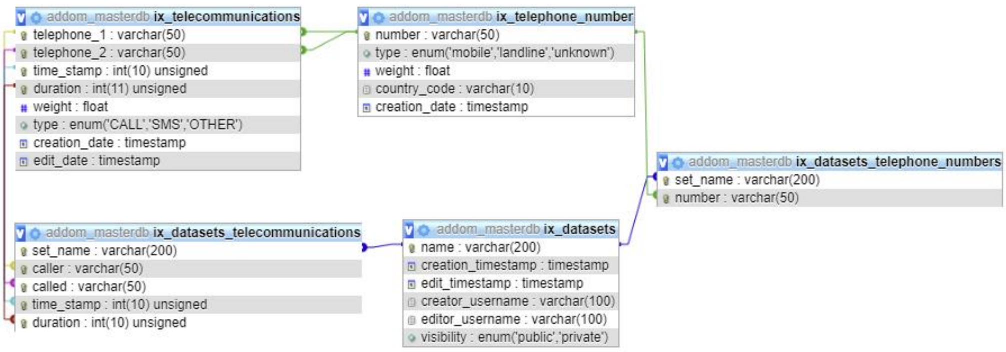


Figure 9

(IntelX Application Database Schema)

3.3 Inserting Data

There are two available ways to insert data into the IntelX database. Either by uploading a .csv file containing telecommunication or people information data or by importing telecommunication data using an import form built in the web application itself.

To upload a .csv file the user presses the “Upload Data” link on the left side menu (Figure 10). In the floating panel that appears the user can either process an already uploaded file in the first tab (Uploaded Files) or upload a new .csv file through the second tab (Upload File). By clicking on the “Upload File” tab, the user can press the “Choose File” button and navigate to locate the desired file. If a non .csv file is selected, the user gets a warning. Upon selection of the appropriate file, the upload begins automatically and once it is over, the application navigates to the first tab “Uploaded Files” automatically. There, the uploaded file with the current date and time as its name is visible. There are three possible options for each uploaded file. The user can either rename the file by pressing the far left button (which is next to the uploaded files name), process its contents to insert them into the database by clicking the middle button (the cog button), or delete the file by clicking the right button (with the delete symbol).

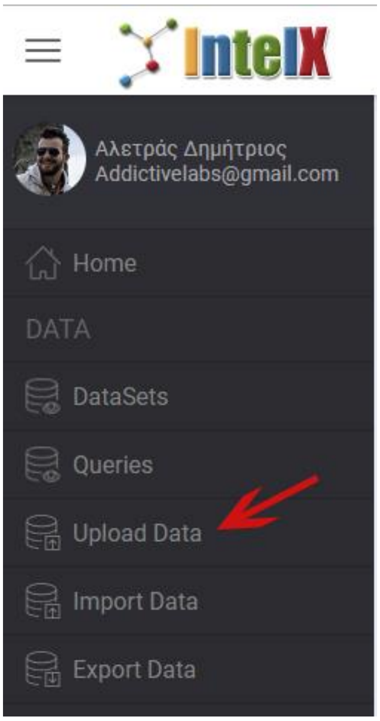


Figure 10

(IntelX Application Left Sidebar Menu)

These buttons can be seen in (Figure 11). When the user clicks the “Rename” button, an input field appears where the desired file name can be typed in. To finalize the rename the user clicks the “Rename” button next to the input field. To cancel, the user can click the “Cancel” button. By clicking the “Process Button”, a floating window appears (Figure 12). Through the dropdown list the user can choose whether to import telecommunication data, or people data (telephone owner/user data). Below the dropdown list the desired csv file delimiter is provided. By default the “ ; ” is selected delimiter. Finally, the user can select one of the saved import templates. By right clicking on one of the templates the user can also delete it by clicking on the “Delete” option. By clicking the “Next” button, a new floating window appears (Figure 13). This

Figure 11

(IntelX Application Uploaded File Processing Options) window includes the most important part of the importing procedure. The first 100 rows of the selected and uploaded csv file are available in the new floating window.

Above each column of the data, there is a dropdown list which assigns the database field that each column of data represents. Above each column of data an “Add Filter” button also exists which is used to manipulate each column separately. Finally, the user can choose to ignore a number of lines from the importing process by changing the value above the data table. The predefined value of zero can be changed to one, for example, if the user wishes to skip the file’s first line which might contain headers. For the database field assignment, the dropdown list contains three option groups, if the telecommunication data import is chosen in the previous step. The telephone 1 option group, which

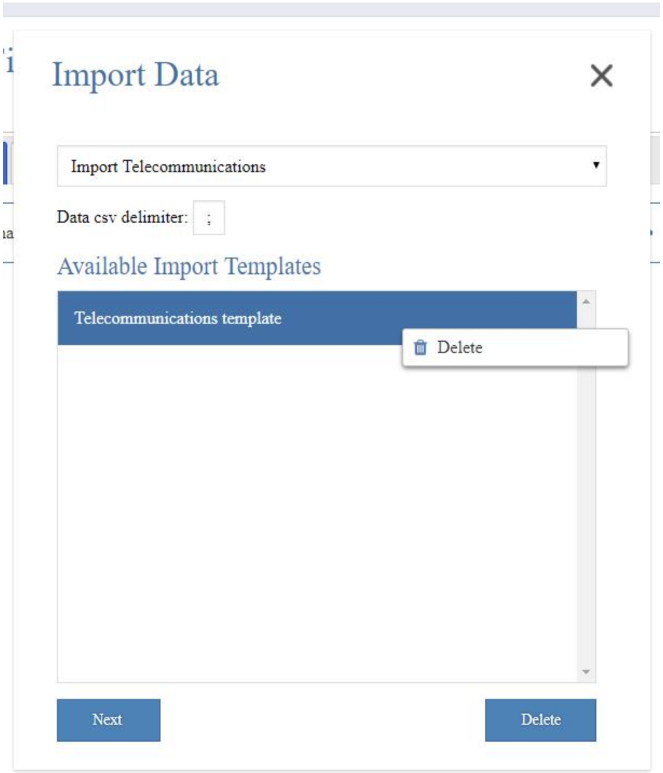


Figure 12

(IntelX Application Type of Import Dialog)

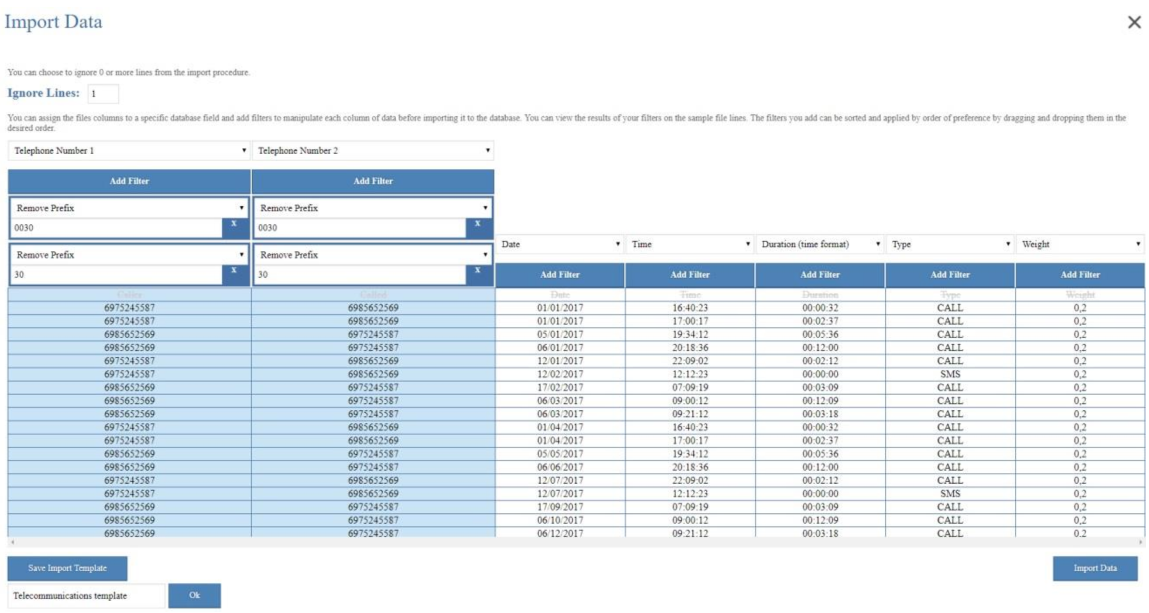


Figure 13

represents the called number of the communication, and the telecommunication fields group. The mandatory fields for the telecommunications import are the Telephone 1 number, the Telephone 2 number, the date, time or timestamp (which includes the date and time in the same column) and the duration of the communication. The weight option is not mandatory and if left unassigned, the default value for each telecommunication is 0,5. Upon pressing the "Add Filter" button, a dropdown list of available filters appears. The available filters are:

- Remove Prefix → If a value is provided, the application iterates through the data and removes the given value when found as a prefix. (If the value 0030 is provided for example, the telephone number 00306978676768 will result in 6978676768).
- Remove Suffix → If a value is provided, the application iterates through the data and removes the given value when found as a suffix. (If the value " Str." is provided for example, the address value "Balder Str." will result in "Balder").
- Add Prefix → If a value is provided, the application iterates through the data and adds the given value as a prefix. (If the value "+30" is provided for example, the telephone number 6978655647 will result in +306978655647).
- Add Suffix → If a value is provided, the application iterates through the data and adds the given value as a suffix. (If the value " Str." Is provided for example, the address value "Balder" will result in "Balder Str.").
- Remove Fixed Prefix → If a numerical value X is provided, the application iterates through the data and removes the first X characters. (If the value 2 is provided for example, the value 0030697823 will result in 30697823).
- Remove Fixed Suffix → If a numerical value X is provided, the application iterates through the data and removes the last X characters. (If the value 2, the value 0030697823 is provided for example, the resulting number will be 00306978).
- Exclude if Contains → If a value is provided, the application iterates through the data and excludes the row in which the value is found from the import process.

- Replace With Null → If a value is provided, the application iterates through the data and replaces the value with “”. (If the value “aaa” is provided for example, the value cccaaabbb will result in cccbbb).

Multiple filters can be applied on each column and all of them are applied in the order they appear. The filters are draggable, so the user can change their order regardless of the order in which he added them. After the assignment of columns to database fields is over, the user can choose to save these specific import settings as a template by clicking the “Save Import Template” button on the bottom left corner of the floating panel. A template name is provided and the “Ok” button is clicked to save it. This process should be done if the user imports files with the same column number and order frequently. By clicking on the “”Import Data” button, a new floating dialog appears, requesting the dataset name with which the data will be associated (Figure 14). Again, the user can either select a pre-existing dataset or create a new one. By pressing the “Continue” button, the import process begins. After the import process is complete a new floating panel appears which illustrates the import statistics (imported rows, duplicate values, errors etc.). A log which includes those statistics and the specific errors that might appear is also created, and is accessible by clicking the “here” link on the statistics floating panel.

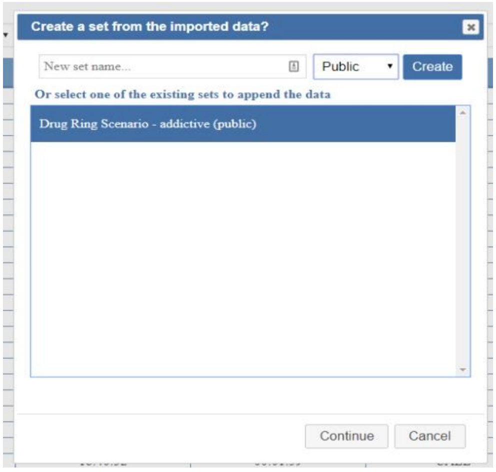


Figure 14

(IntelX Application Dataset Selection)

The second way to import data into the database is by using the “Import Data” function, by clicking the corresponding link on the left side menu of the application. The user can either import telecommunication data or people – telephone numbers association data, depending on which input fields he fills in. The respective fields can be seen in Figure 15. An error message that will ask for the necessary fields will be displayed if the user tries to import telecommunication or people data without filling in the mandatory fields. The mandatory fields for a telecommunication are all the input fields except for the type and weight. The mandatory fields for a person – telephone association are the telephone number, a surname, a name and an id number. If any of the data already exist, a notification will be shown.

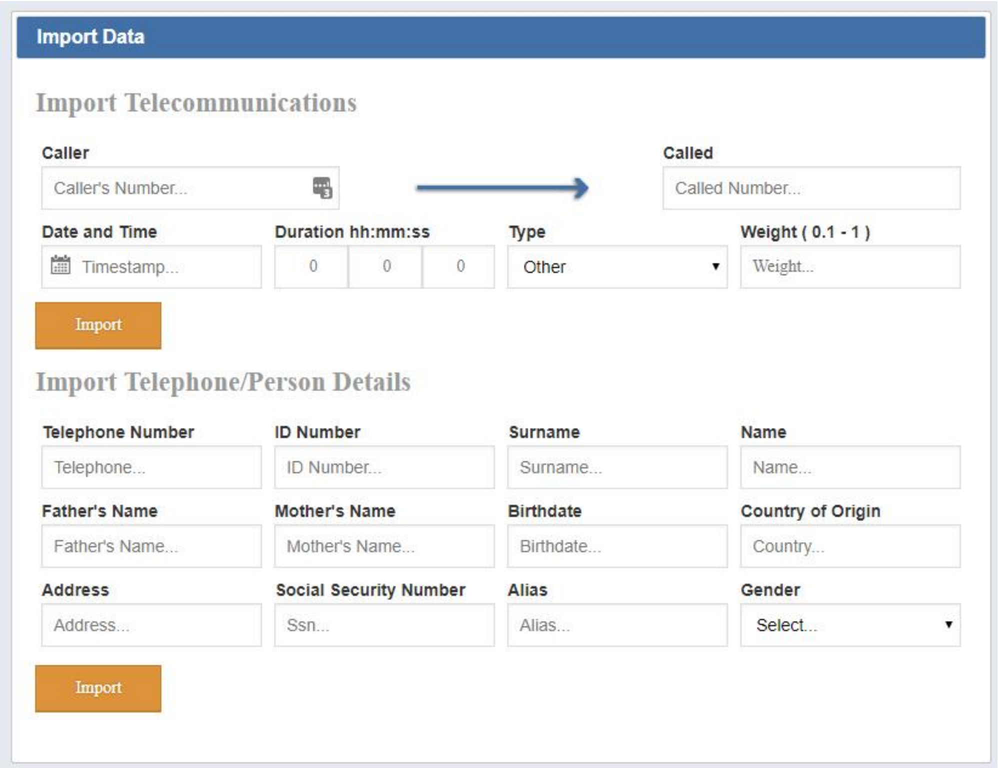


Figure 15

(IntelX Application Import Data Form)

3.4 Exporting Data

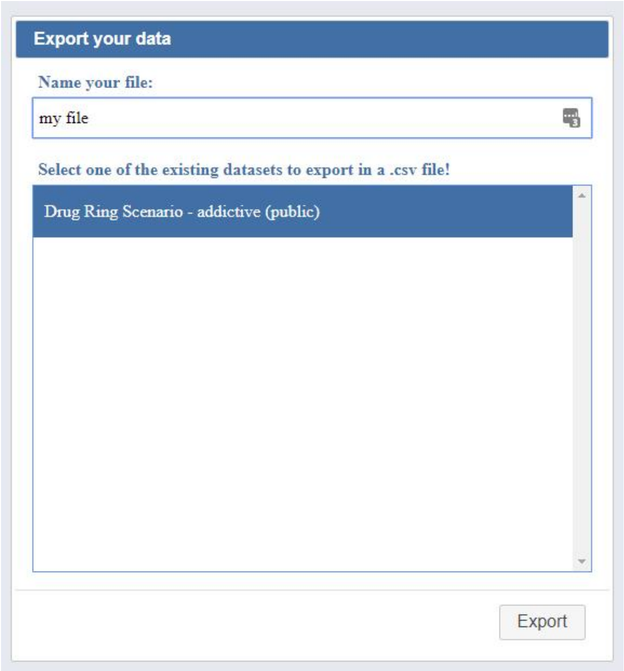


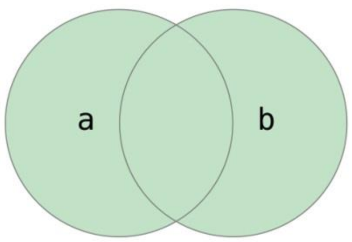
Figure 16

(IntelX Application Export Data Dialog)

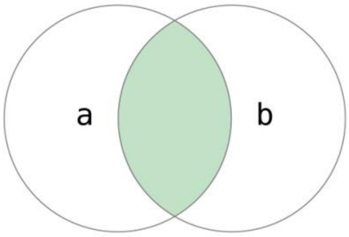
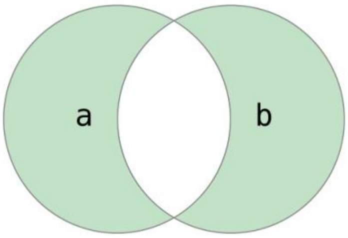
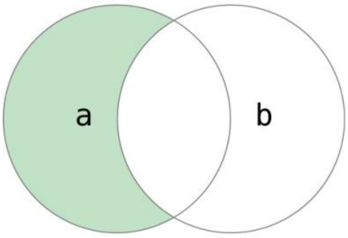
To create a .csv file containing your selected datasets telecommunication data, the user needs to click on the “Export Data” link on the left side menu of the “IntelX” application. There, the user provides a file name in the initial input field and selects one of the existing datasets whose contents he wishes to export. The user then clicks the “Export” button and the file download begins. The export data floating panel is shown in Figure 16. The fields than get exported are: Caller Telephone Number, Called Telephone Number, Timestamp (dd/mm/yyyy hh:mm:ss format), Duration (number of seconds), the telecommunications Weight (0.1 – 1) and finally the type (SMS or CALL). The user can process data easily with any application that can open and edit a .csv file.

3.5 Datasets

In this section, the Datasets viewing, editing, deleting and usage is described. To visit the Datasets section the user needs to click the "Datasets" link on the left side menu. A single floating panel contains all the existing datasets. The creation timestamp and the user that created the dataset are visible on each dataset. One or more datasets can be selected by left clicking on each one of them. When clicked, the user can see two activated buttons on the bottom left side of the panel. The first button is the "Delete" button which, as its name states, is used to delete a dataset after confirmation. More than one datasets can be deleted at once. Important thing to notice however is the fact that deleting a dataset will NOT delete its contents. The deletion only affects the data–dataset association with the risk of having "orphan" data (data that is not associated with any dataset) as leftovers. The datasets are mainly used for association purposes in the IntelX application. The second button, which only activates when one dataset is selected and deactivates when more are selected, is the "Visualize" button. By pressing it, the application navigates to the Visualization section of the application and the dataset's contents are added on the canvas. The final button left to explain is the "Combine Datasets" button. Upon pressing it, another floating panel appears. It makes no difference having selected datasets or not. In the dropdown list, on top of the panel that appears, the user can select how the datasets will be combined. The available options are:



Union → As the image implies, this option generates a new set containing the contents of all the selected datasets. Sets can be added in order to be unified by pressing the circular blue button with the white cross on the right side of the panel. There are at least two datasets necessary for this process, but there is no upper limit. The more datasets the user adds and the bigger they are, the longer this process will take to complete.

	Intersection → As the image implies, this option generates a new set containing the records common to both the designated datasets. A limit of two datasets has been set since the database queries required for this process are using complicated joins which become really intensive with more datasets. Any result can still be achieved by using some more steps and datasets.
	Asymmetric Difference → As the image implies, this option generates a new set containing the records **NOT** common to both the designated datasets. A limit of two datasets has been set since the database queries required for this process are using complicated joins which become really intensive with more datasets. Any result can still be achieved by using some more steps and datasets.
	Subtracting → As the image implies, this option generates a new set containing the datasets “a” records but **NOT** those that also exist in dataset “b”. A limit of two datasets has been set since the database queries required for this process are using complicated joins which become really intensive with more datasets. Any result can still be achieved by using some more steps and datasets.

(Table 1)

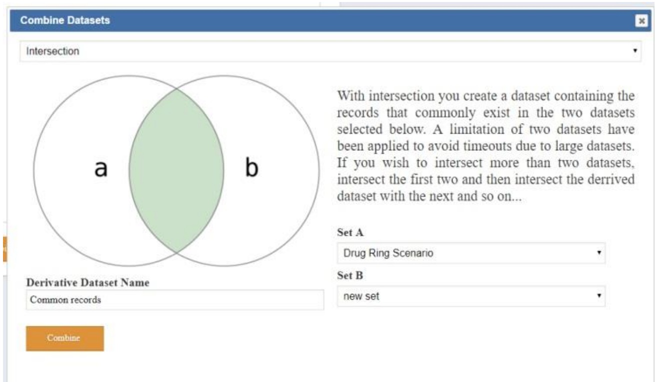


Figure 17

(IntelX Application Combine Datasets Dialog)

The derivative datasets name is provided in the input field on the bottom left side of the floating panel and the process starts after the user presses the button "Combine" (Figure 17). Upon completion, the new dataset is created and can be accessed through the queries section of the IntelX application which we will go through next. Finally, empty dataset can be created by clicking on the add new set circular blue button on the bottom right side of the initial floating panel by providing a dataset name and clicking the "ok" button. Data cannot be added to this dataset at this point however. The only way to add data to a dataset is through the importing process and through the querying process.

3.6 Queries

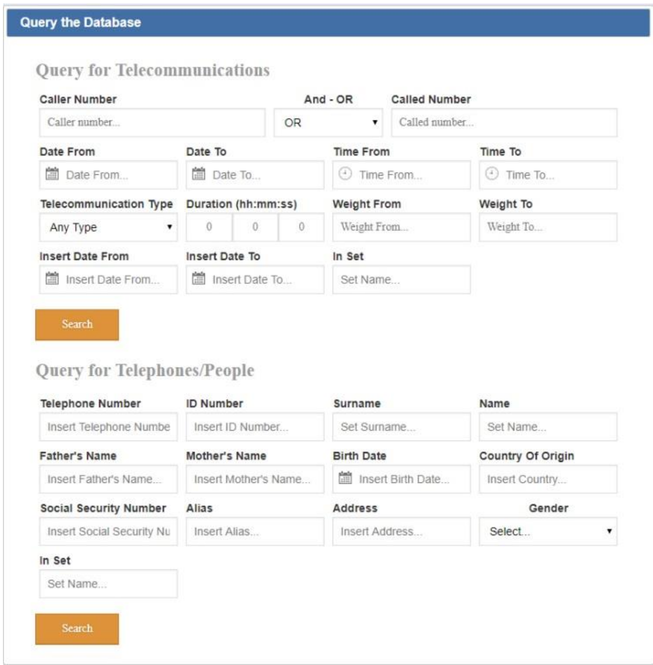


Figure 18

(IntelX Application Query Form)

In this section the process of querying the database is described. To visit the queries section the user needs to click on the "Queries" link on the left side menu of the IntelX application. Upon left clicking on the link, a floating panel appears containing multiple query input fields that will help the user query the database for records effectively (Figure 18). The user can specify a telephone number being the caller or the called number, or specify specific caller and called numbers by using the AND option from the And – Or Dropdown List. A date or time range can be specified, the telecommunication type, the duration (as a minimum duration), a weight range as well as an insert to the database date range. Finally, the user can specify the dataset in which he wishes to search for his records by filling the dataset input field. There is an auto-fill list containing all the existing datasets so as the user types the dataset name, he can select it from the list that appears to ensure that the dataset name that is being typed is the correct one.

The database can also be queried for specific telephone numbers or the people data that are associated with it, by using the second query form appearing in Figure 18. The database can be queried based on the surname, name, father's or mother's name, birthdate, country of origin, social security number, alias, address, gender and of course a specific telephone number. Finally, the user can search for people in specific datasets by specifying the dataset name in the respective input field.

Upon querying the database, a floating panel appears with the query's results (Figure 19). On the upper left corner, the total number of records found is indicated. The results are paginated with 600 records appearing in each page for performance reasons.

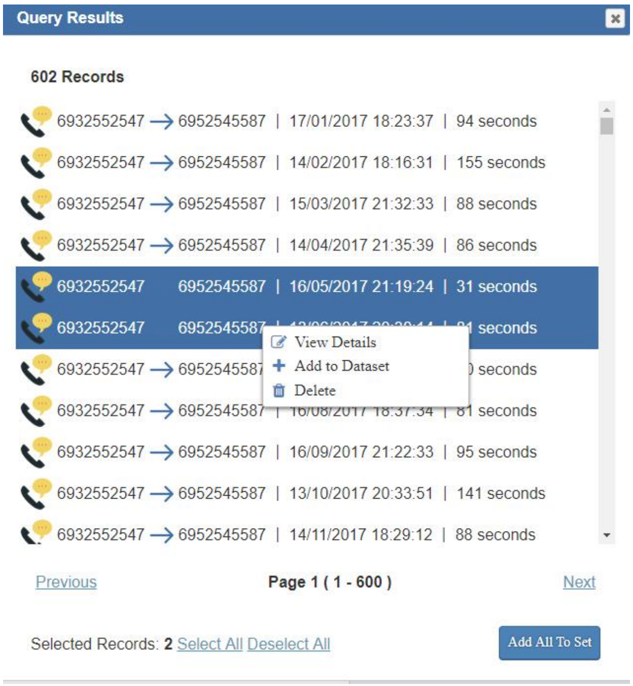


Figure 19

(IntelX Application Query Results Dialog)

By clicking the "Previous" and "Next" links you can navigate through all the query result pages. The "Select All" link selects all the records in the current page and the "Deselect All" link deselects them. You can select as many records as you wish by left clicking on each one of them, or selecting them all with the method explained above. You can right click on one of your selected records and select among three options. The "View Details" option only applies to the record you right clicked on, and triggers a new floating panel which indicates the records details (Figure 20). By clicking the "Edit" button on the Details floating panel, input fields appear containing the current values which you can then change to your liking (Figure 21). By pressing the "Edit" button again, you can verify your changes and by pressing the “Cancel” button, you can cancel the edit process. If there are more data associated with the telephone number, a different table will appear in the details floating panel, containing the corresponding data (Figure 22).

Figure 20

IntelX Application Telecommunication Record Details Dialog

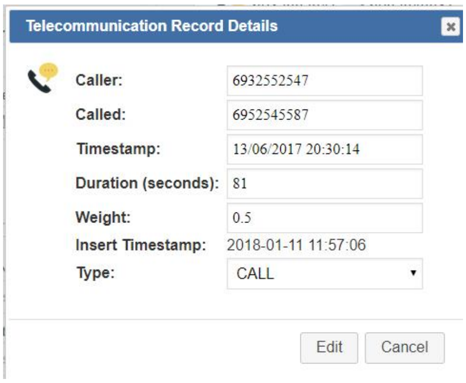


Figure 21

IntelX Application Telecommunication Record Edit Details Dialog

Figure 22

(IntelX Application Telephone Number Record Details Dialog)

By left-clicking on the red X mark on the data table in Figure 22, you can disassociate the data from the telephone number, while by clicking the edit icon right next to the red X, you can edit the persons data and confirm your choices by clicking the OK button.

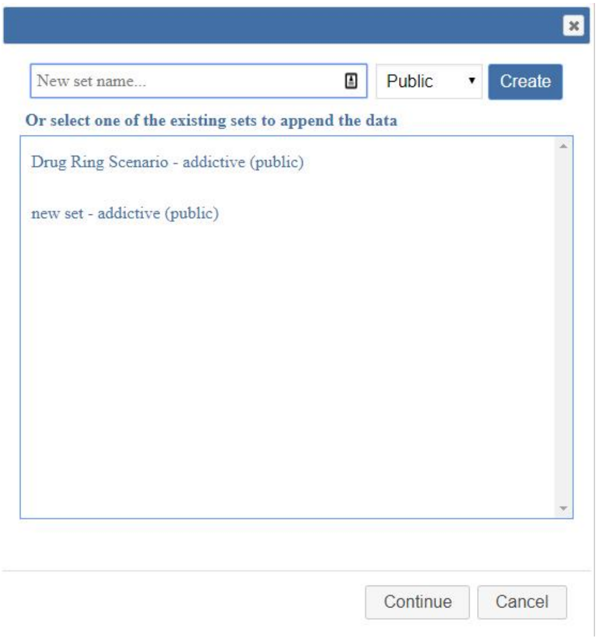


Figure 23

(IntelX Application Dataset Association Dialog)

The second option from the right-click menu, the "Add to Dataset" option triggers another floating window which lets you select which dataset to associate your selected records with (Figure 23). This panel should seem familiar since it is also used in the importing phase to select the dataset your records will be imported in. You can either create a new dataset to associate your data with, or use one of the existing. The third option in the right-click menu is used to delete your selected records after you confirm your action on the confirmation box that is generated. Finally, by clicking the "Add All To Set" in Figure 19, you can associate all your results, regardless of the page in which they appear, to a specific dataset by triggering the Figure 23 floating panel.

3.7 Graphs – Key Player Approach

In this section the visualization and actual analysis capabilities of the IntelX application are described. To visit this section, the user needs to left click on the "Key Player Approach" link, under the GRAPHS section in the left side menu.

When the application navigates in this section, the user encounters and empty html5 canvas which is going to be the graphs actual canvas. A menu appears on the right side of the screen with 8 buttons (Figure 24). Each button has a title which appears when the mouse pauses over it. The first button (Import From Dataset) is used, as the title implies, to import data from a dataset. Upon clicked, a familiar floating panel appears (Figure 23)

Figure 24

Right Side Bar Menu

and allows the user to select a dataset whose records will be imported in the canvas represented as a graph. The user can select one dataset and click on the “Import” button on the right bottom corner of the floating panel. The data will be imported briefly into the canvas as a graph (Figure 25).

3.7.1 Canvas Functions

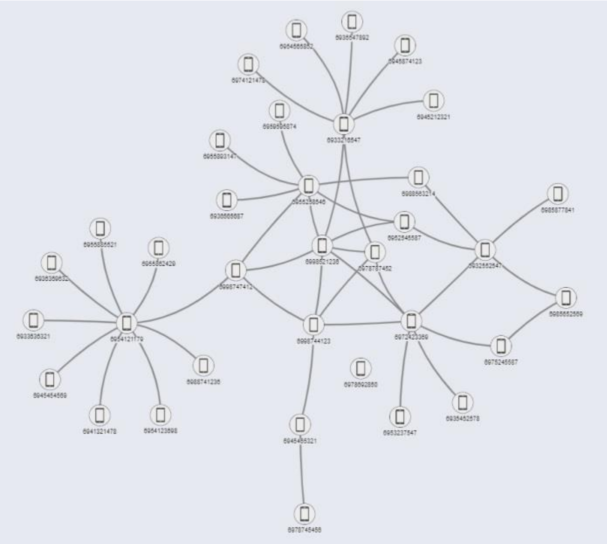


Figure 25

(IntelX Application Network Example)

The graph generated resembles the one in Figure 25 (depending on the data of course). Before moving on with the menus buttons, the import process must be explained. The process is easier explained with an example.

Let’s assume a dataset which contains two different telephone numbers with five telephone calls. So, for instance, Telephone 1 calls Telephone 2 five times. The directions of the calls are irrelevant at this point. So let us assume that the authorities that evaluate the calls importance, mark one of the calls made between Telephone 1 and Telephone 2 as crucial, because a drug trade is discussed, with a weight of 0,9 . The rest of the calls are meaningless as they discuss everyday matters so they are marked with a weight of 0,2. When we import the dataset containing the above telecommunications and telephone numbers, a graph of two nodes/vertices connected with **one** edge appears. One would expect five edges between the two vertices. The reason why only one is imported is simply because at this kind of analysis (key player approach) we only need to establish the relationship between these two vertices and how strong/important their tie is. For that reason, when the user selects the dataset to import, the edges between every pair of vertices are merged as one, retaining the maximum weight found between all the separate edges. So in the above example for instance, the edge that connects Telephone 1 and Telephone 2 on the graph is going to have a weight of 0,9. As an additional implication of the merging, all call directions are lost as well. The reason why the maximum weight between all communications between every pair or vertices is retained, is the fact that in criminal investigations, possible assailants rarely use the telephone for their business and if they end up doing so, they usually use code names and keep the conversations as brief as possible. For that exact reason the application uses the maximum weight, since an average weight might undervalue the vertices relationship in case they only talked about “business” once in ten calls/communications.

There are a number of functions that can be achieved by double clicking or right clicking on the canvas or on edges/vertices. When double-clicking on a telephone number for instance, a floating panel with the numbers details appears. The telephone numbers’ details can be edited on the spot by clicking the “Edit” button and confirming the changes. In case the telephone number is changed, the changes are also applied on the graph by changing the telephone numbers label. By double-clicking on an edge, a floating panel similar to the one that displays a query results appears. It displays all the telephone calls or sms messages that occurred between the connected vertices. By right clicking on each of those telecommunication records and selecting the “View Details” option, another floating panel appears containing the right-clicked telecommunications details. Those details can be edited by clicking the “Edit” button and confirming the changes afterwards. The changes made are also applied to the graph (only a weight change might have an effect on the graph). By right clicking on the canvas (you cannot right-click on a specific node/vertice or edge) you can select either one of three options:

i. “Remove Selected From Network”, which will remove the selected nodes and edges from the graph. The same result can be achieved by pressing the delete button on your keyboard.
ii. “Expand selected from Database” which will search the database for records that do not appear in the network and import it. If for instance Telephone 1 has communicated 3 other telephone numbers and only one of those appears in the current network, this option will import the rest of the telephone numbers and their merged telecommunications in the graph.
iii. “Undo” which will cancel the last delete, copy, cut, paste action. The same result can be achieved by pressing the Ctrl+Z combination of keys on the keyboard.

3.7.2 Search

 By pressing the “Search for a telephone number” button (2nd button) on the right side menu, a floating panel appears which contains an input field for a telephone number. The user can type the telephone number he is interested in finding and if it exists inside

the network/graph he is working on, the application will zoom in the exact position of the telephone number in the graph.

3.7.3 Multiselect Enable/Disable

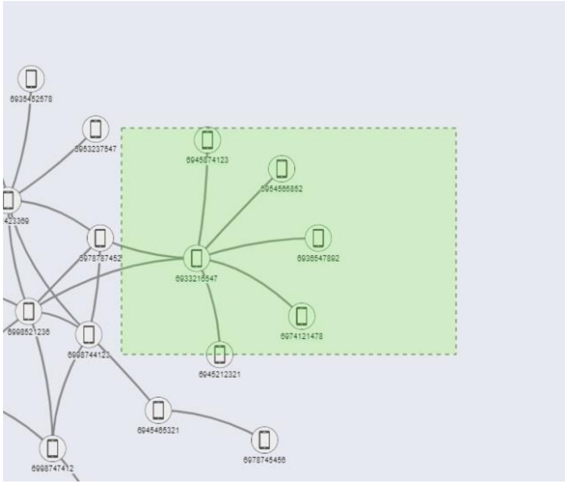
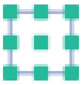


Figure 26

(IntelX Application Multiselect Feature)

By pressing the third button on the right side menu, the user enables, or if already enabled, he disables the multi-select feature. When enabled, the user can hold the right mouse button pressed and drag the mouse to select multiple node and edges in the graph (Figure 26). When done selecting, the user can press the multi-select button again to deactivate it and allow for the right-click function of the canvas to work again. You might notice that when holding your right mouse button pressed, the stabilization movement of the network momentarily stops for the graphics to be drawn and resumes when you release your mouse button.

3.7.4 Selection Expansion



By pressing the 4th button on the right side bar menu (the one with the title "Select connected nodes and edges") the user can expand his selection by a number of levels. Any number of telephone numbers can be selected before pressing the expansion button. The number of levels for the selection expansion is specified and the expansion starts after the "Ok" is pressed. By default, level 1 is typed in the input field, so only the nodes connected to the selected nodes will be added to the selection. Figure 27 illustrates the level 1 expansion while Figure 28 illustrates a level 2 selection expansion.

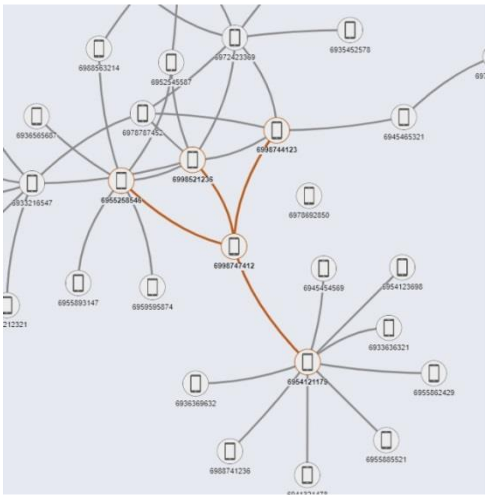


Figure 27

(IntelX Application Selection Level 1 Expansion)

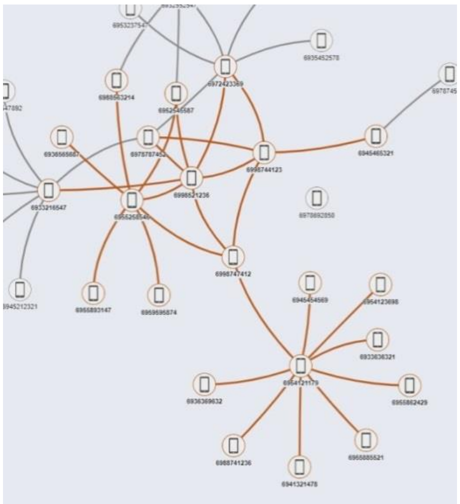


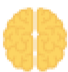
Figure 28

(IntelX Application Selection Level 2 Expansion)

3.7.5 Path Tracing

 By pressing the 5th button on the right side menu bar (the one with the "Trace connection path" title), the user can discover all available paths connecting two or more telephone numbers. The path may seem obvious in small graphs, but try to imagine a graph containing all the calls made in a telecommunications cell's radius. It would include thousands of calls to say the least, so if you wish to examine if two numbers are connected and in which way, you can use this feature and perhaps cut the selection (using Ctrl + X), clear the canvas and then paste it so you can examine them up close. If the selected nodes have no connections you will get a notification message saying so. If you include more than one telephone number in your initial selection, then the algorithm will try to locate a path including all selected telephone numbers.

3.7.6 Borgatti's Enforcement Analysis

To use Borgatti's fragmentation formula for the purpose of Enforcement analysis in a graph in the IntelX application, the user needs to have an existing network/graph and press the  button in the right side bar menu. When pressed, the graph's nodes and their connections are "grouped" in a javascript adjacency list like in the following example: Suppose we have a graph like the one in Figure 29. Upon pressing the Fragmentation analysis button shown above, a javascript object looking like this is sent to the back-end.

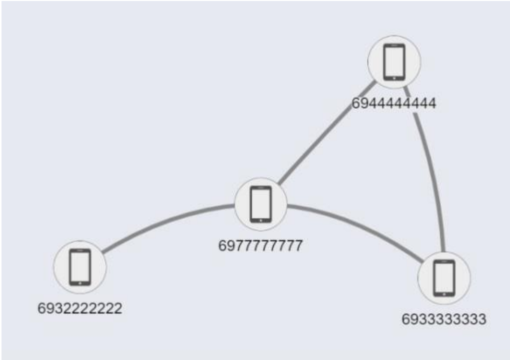


Figure 29

(IntelX Application Network Example)

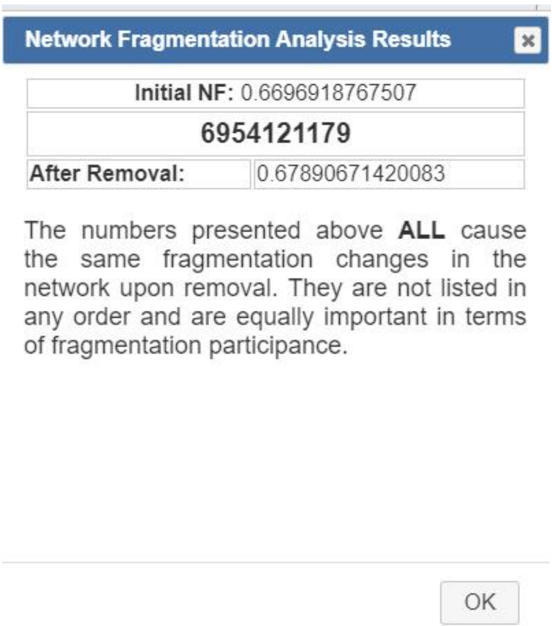


Figure 30

IntelX Application Fragmentation Analysis Results Dialog

{
  “6932222222”: [“6977777777”],
  “6977777777”: [“6932222222”, “6944444444”,
                 “6933333333”],
  “6944444444”: [“6977777777”, “6933333333”],
  “6933333333”: [“6977777777”, “6944444444”]
}

There, using dijkstra’s algorithm, the shortest distance for every pair of nodes is measured and Borgatti’s formula is calculated. The results are stored in a php array and the pair of node-fragmentation values with the highest fragmentation measure is returned to the front-end. The results are presented to the user with a floating panel (Figure 30). In the panel shown in Figure 30, you can see the initial Network Fragmentation, the node that upon removal causes the maximum network disruption and the network fragmentation after the nodes removal. In case more than one nodes cause the same disruption, you will notice more than one telephone numbers in the table. You can easily locate the target telephone number/node on the graph since it is enlarged and marked with a red border for as long as the results floating panel stays visible. You can also left click on the nodes label on the results floating panel and the application will zoom into the node’s position on the graph. In case you remove the target node (by pressing the Delete button on your keyboard or right clicking on it and selecting the “Remove selected from network” option from the menu, you can recalculate the fragmentation by pressing the same right side bar menu button. The measure will of course differ since the target node will be missing.

3.7.7 Borgatti's Intelligence Analysis

To apply the Borgatti's Intelligence Analysis Algorithm, you need to have a network and press the corresponding button in the right side bar menu. Upon pressing the above button, and assuming we have a network like the one pictured in Figure 29, a javascript object representing the current graph/network in the form of an adjacency list, is sent to the back end. The object sent looks a lot like the following part of code:

```
{
  “6932222222”:  [“6977777777”],
  “6977777777”:  [“6932222222”, “6944444444”, “6933333333”],
  “6944444444”:  [“6977777777”, “6933333333”],
  “6933333333”:  [“6977777777”, “6944444444”]
}
```

There, the application loops through each pair of nodes, calculating the sum of their minimum distance (obtained using dijkstra's algorithm). After the iteration is completed and the results are stored in an array, the three highest values are returned to the front end. The results are displayed as enlarged nodes with red stroke on the graph and in a floating panel like the one viewed in Figure 31. Notice that depending on the position the node holds, its circle is bigger on the graph. The user can view the three telephone numbers/nodes with the biggest reach in descending order as well as their reach right below the telephone number in the results table in Figure 31. By clicking on the telephone number on the results table in floating panel, the viewport focuses on the corresponding node on the graph to help the user locate it.

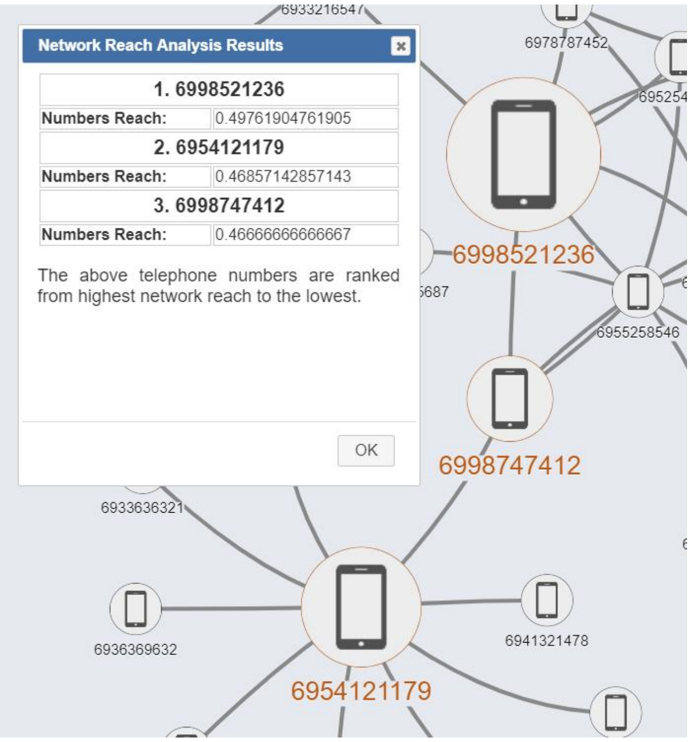


Figure 31

(IntelX Application Reach Analysis Results Dialog)

3.7.8 Daniel M. Schwartz – Tony (D.A.) Rouselle's Enforcement Analysis

To use Schwartz and Rouselle's formula in the IntelX application the user needs to left click the Enforcement Analysis button  on the right side bar menu. By doing so, a floating panel appears that requires from the user to provide an RSL value. This RSL value, just like discussed in chapter 2.3, represents the percentage of the amount of resources that the two nodes are willing to share. This is entirely up to the analyst who uses the application to decide, but usually a value between 0.3 and 0.45 is realistic. After the RSL value is filled in, the user presses ok and waits for the analysis to complete.

A javascript object like the following

```
{
 Edges: {
  0: {from: "…", to: "…", weight: "…"},
  1: {from: "…", to: "…", weight: "…"},
  .
  .
  .
 },
 Nodes: {
  Id: "…",
  Weight: "…",
  Connections {
   0: {from: "…", to: "…", value: "…", dataset: "…", id: "…"},
   …
  }
 }
}
```

is sent to the back end for processing. This particular algorithm is more time consuming than the previous since it is ran three times in a row. The first time determines the node that upon removal damages the network's capital the most. After removing that node it is run again and it determines the second most important "target". After determining it, it is run for a final time to find the third target. This happens because each time a node is removed, the connections are altered so the previous measures are no longer representative. After the algorithm is completed three times, a table with the results and their impact on network capital is returned to the front end (Figure 32).

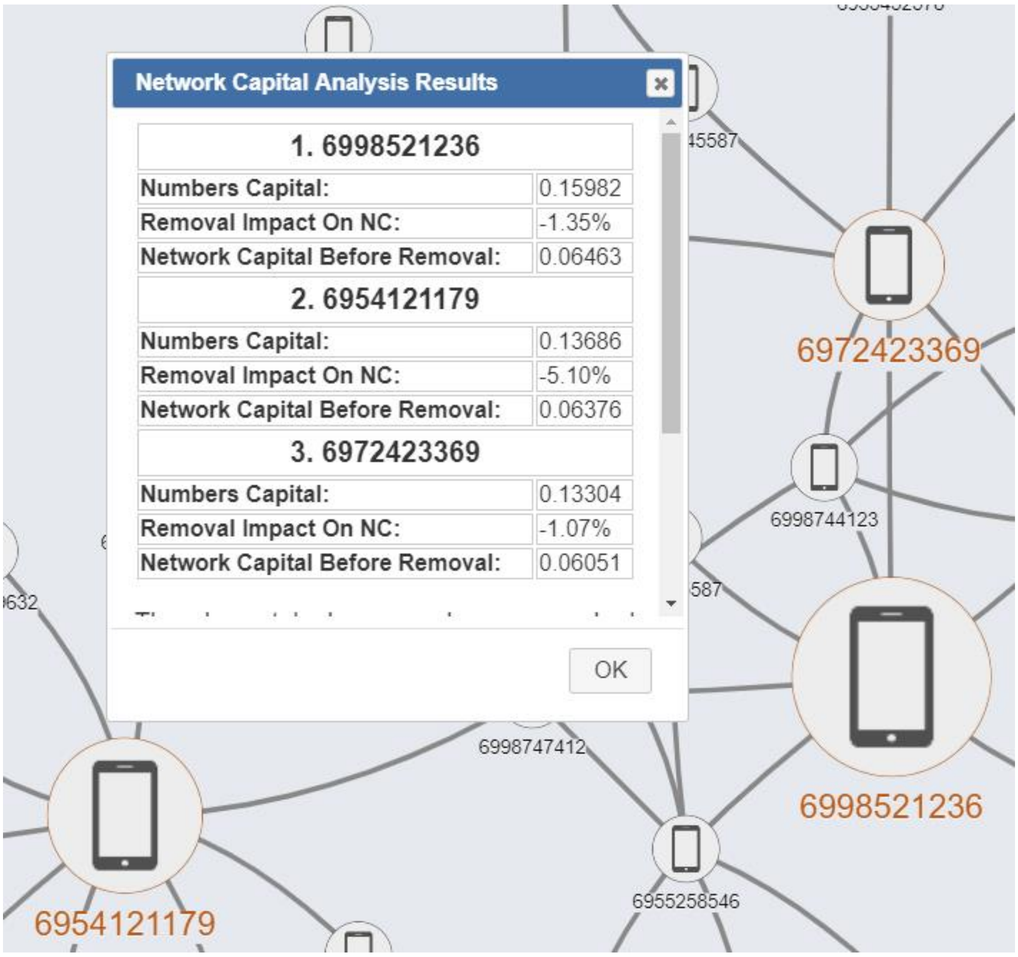


Figure 32

(IntelX Application Network Capital Analysis Results Dialog)

Notice that the floating table in Figure 32 contains the three telephone numbers that are the ideal targets, in terms of enforcement, and their removal impact on the network's capital. The first line, directly below the telephone numbers, denotes the numbers contribution to the NC, the second denotes the decrease in NC percentage and the third line denotes the network's capital before removing the telephone number from the graph. The telephone numbers are placed in such an order provided that they are removed from the network in their given order, in other words, in the example given in Figure 32, 6954121179 will be the second best target only if we remove the best target (6998521236) first. Likewise, 6972423369 will be the third best target only if you first remove 6998521236 and then you remove 6954121179. By left clicking on each telephone number on the floating panel, the viewport will zoom to that number on the graph to help the user locate it.

3.7.9 Degree Centrality

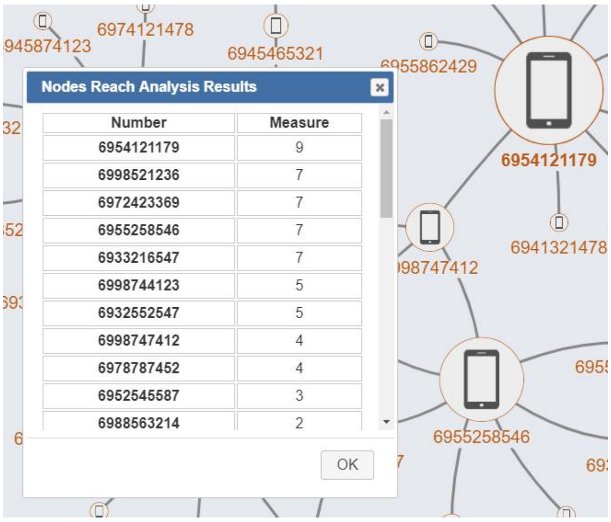


Figure 33

(IntelX Application Degree Centrality Results Dialog)

To calculate the node Degree Centrality, the user has to press the icon from the right side bar menu while having a network in the html5 canvas. By pressing this button, a javascript object containing all nodes and their connected nodes is sent to the backend for calculations. There, the number of connected nodes to each node is calculated and a list of nodes-degree measure is returned to the front end. The results are viewed in a table on a floating dialog (Figure 33). Depending on the measures value, the corresponding node is sized accordingly on the network graph. By left clicking on each telephone number on the results table, the viewport zooms in to help the user locate the telephone number on the graph.

3.7.10 Closeness Centrality

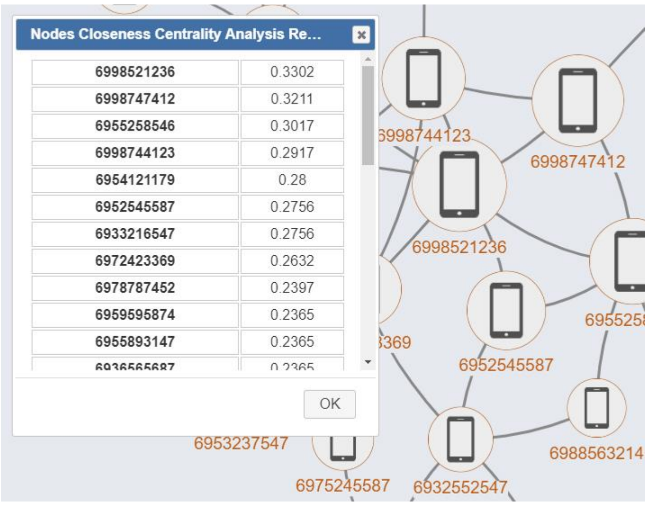


Figure 34

(IntelX Application Closeness Centrality Results Dialog)

To calculate the node Closeness Centrality, the user has to press the CC icon from the right side bar menu. By pressing this button, a javascript object containing all nodes and their connected nodes is sent to the backend for calculations. There, the sum of the length of the shortest paths between each node and the rest of the network's nodes is calculated. Then the normalized form is used and as a result, an array of nodes-closeness measures is returned to the front end. The results are displayed in a table on a floating dialog box (Figure 34). Just like in the reach analysis, the nodes are sized accordingly to match their closeness value. By left clicking on each telephone number on the results table, the viewport zooms in to help the user locate the telephone number on the graph.

3.7.11 Betweenness Centrality

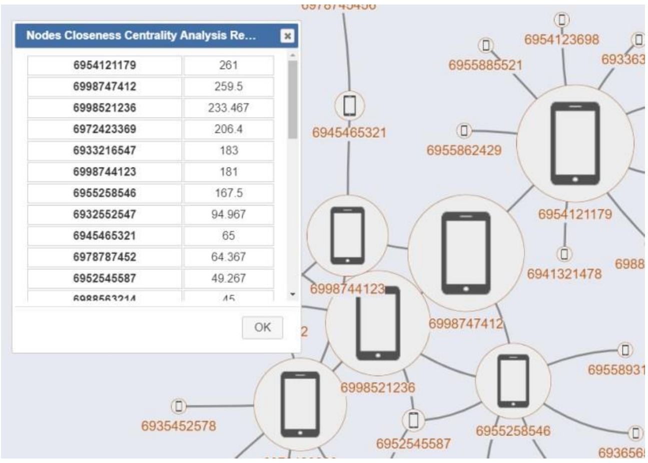


Figure 35

(IntelX Application Betweenness Centrality Results Dialog)

To calculate the node Betweenness Centrality, the user has to press the **BC** icon from the right side bar menu. By pressing this button, a javascript object containing all nodes and their connected nodes is sent to the backend for calculations. There, the algorithm calculates the sum of the number of shortest paths between each pair of nodes in which the node, whose betweenness we are calculating, is a part of, divided by that couple of nodes total number of shortest paths. An array of telephone numbers-betweenness measures is returned to the front end. The results are displayed in a table on a floating dialog box (Figure 35). Just like in the reach and closeness analysis, the nodes are sized accordingly to match their closeness value. By left clicking on each telephone number on the results table, the viewport zooms in to help the user locate the telephone number on the graph.

4 Epilogue

4.1 Conclusions

After describing the centrality measures we have concluded that to pinpoint a node with high connectivity, we need to use the Degree Centrality measure. To identify a node that is closer, in terms of hops, to the rest of the network we need to use the closeness centrality measure. To locate the nodes that tend to act like bridges, connecting larger groups of nodes, we need to use the betweenness centrality measure. The importance of a node in a network, in terms of it being connected to other important nodes, can be found by using the eigenvector centrality.

These methods have proven to be inadequate for some cases however, and the need for other measures has occurred. Stephen P. Borgatti devised two different measures used to identify key players in terms of Enforcement Analysis and Intelligence Analysis. Enforcement Analysis refers to the targets that upon being removed from the network, damage its cohesion by increasing its fragmentation and node distances the most, hence disorganizing the network. Intelligence Analysis refers to identifying the nodes that have the biggest reach in the network, able to diffuse information quickly or upon being placed under surveillance, they would give away a lot of intelligence. Borgatti did not account for edge weighting though, as he considered all edges to be equally important as well as all nodes. Schwartz and Rouselle devised a new model, based on Borgatti's theory which divided the analysis in enforcement purposes and intelligence purposes. They attributed weights on actors and their links and analyzed the effect that a nodes removal would have on the "network capital" and "Intelligence Worth" accordingly.

The majority of these methods of analysis have been implemented in the IntelX web application, to be used with telecommunications networks. The "Intelligence Worth" measure (Schwartz & Rouselle, 2009), have not been implemented since it is the developers opinion that it would not quite fit in the purposes of telecommunication analysis, where the target nodes Intelligence gaps can be filled in rather fast, resulting in constant changes of the attributed weights to either links or nodes. That would become impractical since it is in the nature of these networks to change constantly.

4.2 Limitations

Since the algorithms previously displayed are complex and, in terms of resource consumption, demanding, having a large network could result in high response times, not to mention instability of the graphical interface since the memory management is handled completely by the user’s browser. The IntelX application has been tested with networks that reach up to 2000 nodes and several thousands edges. For larger networks, the graphical interface becomes unstable, depending on the required function. Another possible limitation would derive from a huge data sample since the required data is returned with a sql query to the back end script (php script) where it is being processed, often iteratively. It is possible, for that reason, that the scripts may run out of memory in case the data sample is huge (> 500.000 records), depending on the server settings of course. The IntelX application can prove to be helpful for law enforcement authorities but it requires technical expertise, not to use it, but to interpret its results. Since its results are based on statistical measures, to interpret them you need to know what these measures describe exactly, otherwise the user could jump into false conclusions which to a criminal investigation could lead to serious problems.

4.3 Future Implementations

The IntelX application has ample room for development. More analysis algorithms can be implemented (like eigenvector centrality, pagerank, authoritativeness, hubness and more). Another addition would involve clustering algorithms that would allow the creation, use and analysis of clusters, thus enabling the analysis of larger networks. Implementing more visualization options would also be important, so as to be able to represent the data in different forms fitting every need like hierarchical view, timeline view, eventline view and more. The most important addition however, hence the most time consuming, will involve the ability to import, export, visualize and analyze different kinds of nodes instead of just focusing on telecommunications. It will be necessary to import people, places, vehicles and their in-between relationships (ownership, usage, friendship etc). That would require a lot of work and development hours but would somehow complete the IntelX application and render it a really useful tool for the law enforcement authorities.

5 Bibliography

Intelligence and Security Informatics. (2004). Tucson: Springer.

Bavelas, A. (1950). Communication Patterns in Task-Oriented Groups. *The Journal of the Acoustical Society of America*, 22(6):725-720.

Borgatti, S. P. (2006). Identifying sets of key players in a social network. *Computational & Mathematical Organization Theory*, 12:21-34.

Cambridge Intelligence. (n.d.). Ανάκτηση 2 11, 2018, από https://cambridge-intelligence.com/eigencentrality-pagerank/

Easly, D. (2010). *Networks, Crowds and Markets.* Cambridge University Press.

Freeman, L. C. (1977). A Set of Measures of Centrality Based on Betweenness. *Sociometry*, 40(1):35-41.

Newman, M. (2010). *Networks - An Introduction.* Oxford: Oxford University Press.

Ortiz-Arroyo, D. (2010). Discovering Sets of Key Players in Social Networks. Στο H. A. Abraham A., *Computational Social Network Analysis. Computer Communications and Networks.* London: Springer.

Schwartz, D. M., & Rouselle, T. (. (2009). Using social network analysis to target criminal networks. *Trends in Organized Crime,* 12:188-207.

Sparrow, M. K. (1991). The application of network analysis to criminal intelligence: An assessment of the prospects. *Social Networks*, 13:251-274.

Tsvetovat, M., & Kouznetsov, A. (2011). *Social Network Analysis for Startups.* O' Reilly Media, Inc.

Xu, J. J., & Chen, H. (2005). CrimeNet Explorer: A Framework for Criminal Network Knowledge Discovery. *ACM Transactions on Information Systems (TOIS)*, 201–226.

Appendix A

Here you can find the first 70 rows of the data sample used to provide some of the examples listed above. **No real person or telecommunication data have been used** in this master thesis, hence **<u>no private data violation has occurred</u>.** The data sample is quite bigger, so to refrain from presenting the whole table here, you can access it in the Github repository where the project lies, under the folder name "samples".

Table 2

Caller	Called	Date	Time	Duration	Type	Weight
6975245587	6985652569	01/01/2017	16:40:23	00:00:32	CALL	0,2
6975245587	6985652569	01/01/2017	17:00:17	00:02:37	CALL	0,2
6985652569	6975245587	05/01/2017	19:34:12	00:05:36	CALL	0,2
6985652569	6975245587	06/01/2017	20:18:36	00:12:00	CALL	0,2
6975245587	6985652569	12/01/2017	22:09:02	00:02:12	CALL	0,2
6975245587	6985652569	12/02/2017	12:12:23	00:00:00	SMS	0,2
6985652569	6975245587	17/02/2017	07:09:19	00:03:09	CALL	0,2
6985652569	6975245587	06/03/2017	09:00:12	00:12:09	CALL	0,2
6985652569	6975245587	06/03/2017	09:21:12	00:03:18	CALL	0,2
6975245587	6985652569	01/04/2017	16:40:23	00:00:32	CALL	0,2
6975245587	6985652569	01/04/2017	17:00:17	00:02:37	CALL	0,2
6985652569	6975245587	05/05/2017	19:34:12	00:05:36	CALL	0,2
6985652569	6975245587	06/06/2017	20:18:36	00:12:00	CALL	0,2
6975245587	6985652569	12/07/2017	22:09:02	00:02:12	CALL	0,2
6975245587	6985652569	12/07/2017	12:12:23	00:00:00	SMS	0,2
6985652569	6975245587	17/09/2017	07:09:19	00:03:09	CALL	0,2
6985652569	6975245587	06/10/2017	09:00:12	00:12:09	CALL	0,2
6985652569	6975245587	06/12/2017	09:21:12	00:03:18	CALL	0,2
6975245587	6972423369	15/01/2017	22:12:15	00:01:15	CALL	0,3
6975245587	6972423369	16/02/2017	21:45:12	00:00:59	CALL	0,3
6975245587	6972423369	14/03/2017	22:00:12	00:01:34	CALL	0,3
6975245587	6972423369	14/04/2017	22:00:12	00:01:34	CALL	0,3
6975245587	6972423369	17/05/2017	18:40:32	00:01:59	CALL	0,3
6975245587	6972423369	13/06/2017	18:43:32	00:01:59	CALL	0,3
6975245587	6972423369	15/07/2017	18:32:32	00:02:00	CALL	0,3
6975245587	6972423369	15/08/2017	18:23:32	00:01:59	CALL	0,3
6975245587	6972423369	15/09/2017	18:12:34	00:02:10	CALL	0,3
6975245587	6972423369	12/10/2017	21:02:34	00:03:18	CALL	0,2
6975245587	6972423369	17/11/2017	21:02:34	00:03:18	CALL	0,3
6975245587	6972423369	13/12/2017	22:01:52	00:01:18	CALL	0,3
6935452578	6972423369	16/01/2017	20:21:15	00:01:26	CALL	0,3
6935452578	6972423369	16/02/2017	18:17:19	00:02:26	CALL	0,3

6935452578	6972423369	16/03/2017	18:10:35	00:01:20	CALL	0,3
6935452578	6972423369	16/04/2017	21:33:21	00:02:10	CALL	0,3
6935452578	6972423369	17/05/2017	19:13:13	00:01:17	CALL	0,3
6935452578	6972423369	14/06/2017	18:24:57	00:02:35	CALL	0,3
6935452578	6972423369	16/07/2017	21:16:39	00:02:32	CALL	0,3
6935452578	6972423369	13/08/2017	21:13:31	00:02:28	CALL	0,3
6935452578	6972423369	15/09/2017	18:22:13	00:00:25	CALL	0,3
6935452578	6972423369	17/10/2017	18:26:30	00:00:29	CALL	0,3
6935452578	6972423369	15/11/2017	18:13:41	00:00:30	CALL	0,3
6935452578	6972423369	16/12/2017	21:13:16	00:02:34	CALL	0,3
6953237547	6972423369	16/01/2017	21:34:33	00:00:10	CALL	0,3
6953237547	6972423369	15/02/2017	21:15:46	00:00:12	CALL	0,3
6953237547	6972423369	15/03/2017	20:31:21	00:01:22	CALL	0,3
6953237547	6972423369	17/04/2017	19:31:25	00:00:33	CALL	0,3
6953237547	6972423369	13/05/2017	20:26:38	00:01:23	CALL	0,3
6953237547	6972423369	16/06/2017	21:18:18	00:00:21	CALL	0,3
6953237547	6972423369	13/07/2017	21:37:10	00:00:32	CALL	0,3
6953237547	6972423369	14/08/2017	18:29:57	00:02:18	CALL	0,3
6953237547	6972423369	15/09/2017	19:19:14	00:02:25	CALL	0,3
6953237547	6972423369	16/10/2017	19:16:58	00:01:24	CALL	0,3
6953237547	6972423369	13/11/2017	20:20:11	00:02:10	CALL	0,3
6953237547	6972423369	15/12/2017	19:15:23	00:02:18	CALL	0,3
6932552547	6972423369	16/01/2017	18:29:28	00:01:28	CALL	0,3
6932552547	6972423369	15/02/2017	18:33:48	00:02:24	CALL	0,3
6932552547	6972423369	15/03/2017	18:32:35	00:01:32	CALL	0,3
6932552547	6972423369	17/04/2017	18:20:38	00:00:30	CALL	0,3
6932552547	6972423369	17/05/2017	21:35:47	00:00:48	CALL	0,3
6932552547	6972423369	17/06/2017	19:30:26	00:00:33	CALL	0,3
6932552547	6972423369	17/07/2017	18:12:17	00:00:36	CALL	0,3
6932552547	6972423369	15/08/2017	19:29:34	00:02:35	CALL	0,3
6932552547	6972423369	16/09/2017	21:37:40	00:00:22	CALL	0,3
6932552547	6972423369	13/10/2017	20:10:30	00:00:23	CALL	0,3
6932552547	6972423369	13/11/2017	21:11:35	00:00:26	CALL	0,3
6932552547	6972423369	17/12/2017	18:17:39	00:00:31	CALL	0,3
6985652569	6932552547	13/01/2017	19:11:10	00:00:28	CALL	0,2
6985652569	6932552547	16/02/2017	19:21:37	00:02:20	CALL	0,2
6985652569	6932552547	15/03/2017	18:16:17	00:01:20	CALL	0,2

Table 2 illustrates the telecommunication data taken from a mock drug ring network for the purposes of presenting the IntelX application.

Table 3

Telephone	ID	SSN	Surname	Name	Birthdate	Gender	Fathersname	Mothersname	Alias	Address	Home Country
6975245587	AM-252565		Επίθετο 1	Όνομα 1	25/10/2017	M	Fathersname 1	Mothersname 1	Alias 1	Address 1	Greece
6985652569	AA-121478		Επίθετο 2	Όνομα 2	26/10/2017	F	Fathersname 2	Mothersname 2	Alias 2	Address 2	Greece
6972423369	AB-598836		Επίθετο 3	Όνομα 3	27/10/2017	F	Fathersname 3	Mothersname 3	Alias 3	Address 3	Greece
6935452578	K-142785		Επίθετο 4	Όνομα 4	28/10/2017	F	Fathersname 4	Mothersname 4	Alias 4	Address 4	Greece
6953237547	AN-963654		Επίθετο 5	Όνομα 5	29/10/2017	M	Fathersname 5	Mothersname 5	Alias 5	Address 5	Greece
6932552547	AK-134294		Επίθετο 6	Όνομα 6	30/10/2017	M	Fathersname 6	Mothersname 6	Alias 6	Address 6	Greece
6985877841	AM-012312		Επίθετο 7	Όνομα 7	31/10/2017	M	Fathersname 7	Mothersname 7	Alias 7	Address 7	Albania
6988563214	AN-090897		Επίθετο 8	Όνομα 8	1/11/2017	M	Fathersname 8	Mothersname 8	Alias 8	Address 8	Greece
6952545587	AA-123456		Επίθετο 9	Όνομα 9	2/11/2017	M	Fathersname 9	Mothersname 9	Alias 9	Address 9	Greece
6955258546	AA-908231		Επίθετο 10	Όνομα 10	3/11/2017	M	Fathersname 10	Mothersname 10	Alias 10	Address 10	Greece
6955893147	AZ-900234		Επίθετο 11	Όνομα 11	4/11/2017	M	Fathersname 11	Mothersname 11	Alias 11	Address 11	Boulgaria
6936565687	AZ-923871		Επίθετο 12	Όνομα 12	5/11/2017	M	Fathersname 12	Mothersname 12	Alias 12	Address 12	Greece
6959595874	AZ-909121		Επίθετο 13	Όνομα 13	6/11/2017	M	Fathersname 13	Mothersname 13	Alias 13	Address 13	Greece
6954121179	AK-901209		Επίθετο 14	Όνομα 14	7/11/2017	M	Fathersname 14	Mothersname 14	Alias 14	Address 14	Greece
6941321478	AK-897240		Επίθετο 15	Όνομα 15	8/11/2017	M	Fathersname 15	Mothersname 15	Alias 15	Address 15	Russia
6955885521	AE-121098		Επίθετο 16	Όνομα 16	9/11/2017	M	Fathersname 16	Mothersname 16	Alias 16	Address 16	Greece
6936369632	AE-534296		Επίθετο 17	Όνομα 17	10/11/2017	M	Fathersname 17	Mothersname 17	Alias 17	Address 17	Greece
6933636321	AN-109923		Επίθετο 18	Όνομα 18	11/11/2017	M	Fathersname 18	Mothersname 18	Alias 18	Address 18	Russia
6955862429	AK-100100		Επίθετο 19	Όνομα 19	12/11/2017	M	Fathersname 19	Mothersname 19	Alias 19	Address 19	Greece
6988741236	AK-210059		Επίθετο 20	Όνομα 20	13/11/2017	M	Fathersname 20	Mothersname 20	Alias 20	Address 20	Greece
6954123698	AK-309409		Επίθετο 21	Όνομα 21	14/11/2017	M	Fathersname 21	Mothersname 21	Alias 21	Address 21	Greece
6945454569	AN-102935		Επίθετο 22	Όνομα 22	15/11/2017	M	Fathersname 22	Mothersname 22	Alias 22	Address 22	Greece
6933216547	AA-100199		Επίθετο 23	Όνομα 23	16/11/2017	M	Fathersname 23	Mothersname 23	Alias 23	Address 23	Albania
6945874123	AA-204302		Επίθετο 24	Όνομα 24	17/11/2017	M	Fathersname 24	Mothersname 24	Alias 24	Address 24	Greece
6945212321	AB-456123		Επίθετο 25	Όνομα 25	18/11/2017	M	Fathersname 25	Mothersname 25	Alias 25	Address 25	Greece
6974121478	AK-245690		Επίθετο 26	Όνομα 26	19/11/2017	M	Fathersname 26	Mothersname 26	Alias 26	Address 26	Greece
6954565852	AB-234592		Επίθετο 27	Όνομα 27	20/11/2017	M	Fathersname 27	Mothersname 27	Alias 27	Address 27	Boulgaria
6936547892	AM-124243		Επίθετο 28	Όνομα 28	21/11/2017	M	Fathersname 28	Mothersname 28	Alias 28	Address 28	Greece
6998747412	AN-546023		Επίθετο 29	Όνομα 29	22/11/2017	M	Fathersname 29	Mothersname 29	Alias 29	Address 29	Greece
6978787452	AK-235012		Επίθετο 30	Όνομα 30	23/11/2017	M	Fathersname 30	Mothersname 30	Alias 30	Address 30	Greece
6998744123	AM-340854		Επίθετο 31	Όνομα 31	24/11/2017	M	Fathersname 31	Mothersname 31	Alias 31	Address 31	Serbia
6998521236	AK-123463		Επίθετο 32	Όνομα 32	25/11/2017	M	Fathersname 32	Mothersname 32	Alias 32	Address 32	Serbia
6945465321	AE-124342		Επίθετο 33	Όνομα 33	26/11/2017	M	Fathersname 33	Mothersname 33	Alias 33	Address 33	Greece
6978745456	AM-14550		Επίθετο 34	Όνομα 34	27/11/2017	M	Fathersname 34	Mothersname 34	Alias 34	Address 34	Greece

Table 3 illustrates the mock data of the telephone number owners for the purposes of presenting the IntelX application. You can also access this table as a csv file in the Github repository, under the folder “samples”.