Social Network Analysis on “Game of Thrones”

Hoang-An Nguyen
University of Information Technology
hoanganblue@gmail.com

Trong-Van Duong
University of Information Technology
vanduong0504@gmail.com

Abstract

Social network analysis (SNA) is the process of investigating social structures through networks and graph theory. It characterizes networked structures in terms of nodes (individual actors, people, or things within the network) and the ties, edges, or links (relationships or interactions) that connect them. Furthermore, according to a hundred years' big data analytics, the social network gets fast-growing attention in recent 20 years in social science. In this project, we used the dataset from Andrew Beveridge about the famous HBO series “Game of Thrones”. From that dataset, we found some important characters based on three kinds of measures: geometric measures, spectral measures, and path-based measures. In these measures, we used different kinds of centralities: (1) Degree Centrality, Closeness Centrality for geometric measures, (2) Eigenvector Centrality, Pagerank for spectral measures, and (3) Betweenness Centrality for path-based measures. We further found groups of nodes in the network by two community detection algorithms: Modularity and K-means.

1. Introduction

Books are entertainment tools widely used in our daily life. In addition to entertaining interests, their applications range from providing knowledge for middle-aged people to storytelling for children's education. Many famous books have been transformed into TV series and drew the attention of numerous people. One of these is “A Song of Ice and Fir” which has been transformed into the famous TV series “Games of Thorne” on HBO. In this project, we use amazing data analysis techniques to explore this famous book's dataset and then construct a network of all characters as every character is a node. We use Gephi as a tool for our network visualization. After that, we find some important characters and detect the group of community. The main goals of this project are:

(1) We construct the network of the data and visualize. The great visualization will provide more information about the character and the interaction between them.

(2) We propose to use various brands of measures to find some important characters and compare the results with each other. These measures are: Degree Centrality, Closeness Centrality, Eigenvector Centrality, Pagerank and Betweenness Centrality.

(3) We further introduce two community detection algorithms: Modularity Clustering and K-means Clustering that widely used in clustering problems.

2. Centrality Measures

Centrality is such an important index because it indicates which node takes up a critical position in one whole network. Central positions always get equated with remarkable leadership, good popularity, or an excellent reputation in the network. As soon as the node gets a higher centrality, it means that node gets closer to the network center, that higher power, influence, and convenience from the network may acquire [9, 19]. In centrality measures, there exist three types of measures: (1) geometric measures, (2) spectral measures, and (3) path-based measures. We decide to use some basic algorithms to find a central node in the graph network. For a better understanding and visualization, we take an undirected network graph as an example to interpret.

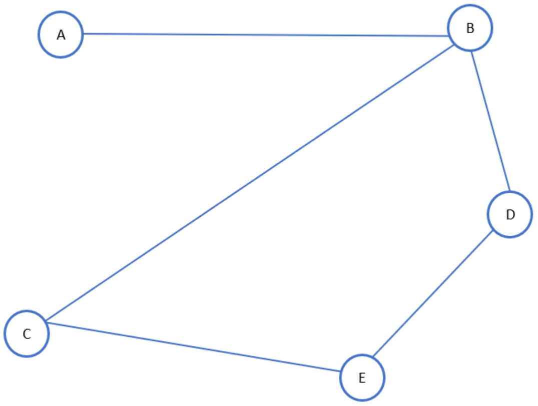

Figure 1: Example of network graph.

2.1. Geometric Measures

2.1.1 Degree Centrality

In a network graph, degree centrality is measured by the total amount of direct links with the other nodes, the fundamental formula Cd is Eq.(1). Since as time goes by, the size of the network may vary, to decrease this possible size effect to degree centrality measurement, Linton Freeman made standardized and put forward to Eq.(2)[9]

$$C_d(i) = \deg(i) \tag{1}$$

$$C'_d(i) = \frac{deg(i)}{n-1} \tag{2}$$

where:

$deg(i)$ = number of ties of node i
n = number of nodes

As example from Figure 1, the number of direct links connected with point B obviously is 3, in other words $C_d(A)$ is 3, after standardization $C'_d(A)$ is equal to 0.75. In the same way we can calculate C_d and C'_D of point A, C, D and E. See the results on Table 1.

	A	B	C	D	E
C_d	1	3	2	2	2
C'_d	0.25	0.75	0.5	0.5	0.5

Table 1: Calculations of degree centrality

2.1.2 Closeness Centrality

Closeness centrality is meant to measure one node to the other nodes' sum distances. If the length of node X's shortest paths with other nodes in the network is small, then node X has a high closeness centrality[18, 1]. It stands for the convenience and ease of connections between the focused node and the other nodes. The fundamental formula C_c is Eq.(3). Gert Sabidussi suggests multiply C_c by $(n-1)$ where n is number of nodes, so that we could get standardized as Eq.(4)[17].

$$C_c(i) = \frac{1}{\sum_{j=1}^{n} d(i,j)} \tag{3}$$

$$C'_c(i) = \frac{n-1}{\sum_{j=1}^{n} d(i,j)} \tag{4}$$

where:

$d(i,j)$ = an shortest path between node i and j
n = number of nodes

Back to the network graph Figure 1, B stand both next to C and E, so the distance to B and C is the same 1. On the other hand, we can see that C stands 2 "step" away from A and D. So $C_c(B) = \frac{1}{1+1+2+2} \approx 0.1667$ and $C'_c(B) \approx 0.6667$.

	A	B	C	D	E
C_c	0.125	0.2	0.1667	0.1667	0.1428
C'_c	0.5	0.8	0.6667	0.6667	0.57142

Table 2: Calculations of closeness centrality

2.2. Spectral Measures

2.2.1 Eigenvector Centrality

Eigenvector centrality measures a node's influence. It starts by measuring each node's degree score – which is simply a count of the number of links that node has to other nodes in the network. However, eigenvector centrality goes a step further than degree centrality in Section 2.1.1. It goes beyond the first-degree connections to count how many links their connections have, and so on through the network. We call the importance of each node its centrality score, and to measure this we'll want the centrality score to be proportional to the sum of the scores of all nodes which are connected to it. This way, if a node is connected to many important node, it will also be an important node, and if it is connected to only a few unimportant nodes then it won't be important. Writing this mathematically gives Eq.(5)[14].

$$C_{eig}(i) = \frac{1}{\lambda} \sum_{j \in M(i)} x_j \tag{5}$$

where

λ = normalization constant
$_{j \in M(i)}$ = sum over all j that nodes i, j are connected

Another way to write Eq.(5) is using the adjacency matrix. If we let x be the vector that has the centrality score of node i in component i, then can be rewritten in vector notation as the eigenvector equation:

$$\mathbf{Ax} = \lambda \mathbf{x} \tag{6}$$

where:

$\mathbf{A}$ = adjacency matrix of network
$\mathbf{x}$ = eigenvector
λ = eigenvalue

In general, the matrix $\mathbf{A}$ will have multiple eigenvalues, but the Perron-Frobenius theorem[5] guarantees that if we demand all the components of the eigenvector $\mathbf{x}$ be positive, then only one eigenvalue that satisfies this requirement. Therefore we can assign a unique centrality score to each node. Power iteration[4] is one of many algorithms that may be used to find this dominant eigenvector $\mathbf{x}$.

Let retake Figure 1 as example. First, we need to calculate the eigenvalue λ for the adjacency matrix $\mathbf{A}$. The value of λ is:

$$\begin{bmatrix}0 & 0.6621 & -0.6621 & 2.1357 & -2.1357\end{bmatrix}$$

Next, we take the maximum value of $\lambda = 2.1357$ and find the eigenvector $\mathbf{x}$. The eigenvector $\mathbf{x}$ we find is also the eigenvector centrality of every nodes in the network.

	A	B	C	D	E
C_{eig}	0.5996	1.2807	1.0678	1.0678	1

Table 3: Calculations of eigenvector centrality

2.2.2 Pagerank

Invented by Google founders Larry Page, PageRank[15] is a variant of Eigenvector centrality in Section 2.2.1. Designed for ranking web content, using hyperlinks between pages as a measure of importance. It can be used for any kind of network, though. PageRank's main difference from eigenvector centrality is that it accounts for link direction. Each node in a network is assigned a score based on its number of incoming links. These links are also weighted depending on the relative score of its originating node. The result is that nodes with many incoming links are influential, and nodes to which they are connected share some of that influence. So, the equation is as follows:

$$PR(i) = \frac{d1-}{n} + d\sum_{j\to i}\frac{PR(j)}{deg_{out}(j)} \tag{7}$$

where

d = residual probability, usually set to 0.85
n = number of nodes
$j \to i$ = set of pages that link to p_i
$deg_{out}(j)$ = number of outbound(outdegree of node j)

Eq.(7) will try many iterations until the network is converging, as we can call that the rank of all nodes is unchanged. Power iteration[4] can be used for this purpose.

Again, take Figure 1 for Pagerank calculations. Note that our graph is undirected so that we will have a link in two directions. For example, we have node A and B, so our graph have a link from node A to B and B to A by contracts. Let d equal 0.85. When initialize time, we call that *0th* iteration, rank of all nodes will be 1. For the *1th* iteration, node A has only node B link to itself, $PR_{0th}(B) = 1$, $deg_{out}(B) = 3$ so $PR(A) = \frac{1-d}{n} + d\frac{PR(B)}{deg_{out}(B)} \approx 0.3133$. Take another example of node B. Node B has three link: A, C and D, $PR(B) = \frac{1-d}{n} + d\left(\frac{PR(A)}{deg_{out}(A)} + \frac{PR(C)}{deg_{out}(C)} + \frac{PR(D)}{deg_{out}(D)}\right) \approx 1.0131$. The results after *1th* iteration in Table 4

Iter	A	B	C	D	E
0th	1	1	1	1	1
1th	0.3133	1.0131	0.7420	0.7420	0.6607

Table 4: Calculations of pagerank

2.3. Path-based Measures-Betweenness Centrality

Betweenness centrality is to measure one node undertaking 'mediation' role in a network. If one node locates in the only way which others nodes have to go through, such as communication, connection, transportation or transaction, then this node should be important and very likely have a high betweenness centrality[8]. Betweenness centrality is a measure of centrality in a graph based on shortest paths. The betweenness centrality for each vertex is the number of these shortest paths that pass through the vertex. Fundamental formula C_b is Eq.(8), standardize C'_b by dividing $\frac{(n-1)(n-2)}{2}$ as Eq.(9).

$$C_b(i) = \sum_{s\neq v\neq t}\frac{\sigma_{st}(i)}{\sigma_{st}} \tag{8}$$

$$C'_b(i) = 2\frac{\sum_{s\neq v\neq t}\frac{\sigma_{st}(i)}{\sigma_{st}}}{(n-1)(n-2)} \tag{9}$$

where:

σ_{st} = number of shortest paths from node s to t
$\sigma_{st}(i)$ = number of those paths that pass through i

Back to the network graph FIGURE 1, take node C for example, we have $C_b(C) = 1$ and $C'_b(C) \approx 0.0833$. The whole results are in Table 5

	A	B	C	D	E
C_b	0	2	1	1	0.5
C'_b	0	0.1667	0.0833	0.0833	0.0416

Table 5: Calculations of betweenness centrality

3. Community detection

Community detection is an important research area in social network analysis where we are concerned with discovering the social network structure[10, 7, 12, 16]. A network is said to have a community structure if the network nodes can be easily grouped into sets of nodes such that each set of nodes is densely connected internally.Community structures are quite common in real networks. Social networks include community groups based on common location, interests, occupation[2]. Finding an underlying community structure in a network, if it exists, is important for several reasons. Communities allow us to create a large scale map of a network since individual communities act like meta-nodes in the network, making its study easier[13]. In this project, we use two common algorithms in network detection: modularity clustering and K-means clustering. In this project, we take modularity clustering

3.1. Modularity Clustering

Modularity is one measure of the structure of networks or graphs. It was designed to measure the strength of a network's division into modules, also called groups, clusters, or communities. Networks with high modularity have dense connections between the nodes within modules but sparse connections between nodes in different modules. There are different methods for calculating modularity. In this project, we take the Louvain method proposed by Blondel *et al*. [3]. The inspiration for this method of community detection is the optimization of modularity as the algorithm progresses. In the Louvain Method of community detection, small communities are first found by optimizing modularity locally on all nodes, then each small community is grouped into one node, and the first step is repeated. The method is similar to the earlier method by Clauset *et al*. [6] that connects communities whose amalgamation produces the largest increase in modularity. The value to be optimized is modularity, defined as a value in the range [-1/2,1] that measures the density of links inside communities compared to links between communities. For a weighted graph, modularity is defined as:

$$Q = \frac{1}{2m} \sum_{ij} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j) \tag{10}$$

where:

A_{ij} = edge weight between nodes i and j
k_i, k_j = sum of the weights of the edges attached to nodes nodes i and j
m = sum of all of the edge weights in the graph
c_i, c_j = communities of the nodes
δ = Kronecker delta function ($\delta(x, y) = 1$ if $x = y$, 0 otherwise)

In order to maximize this value efficiently, the Louvain method has two phases that are repeated iteratively.

The first phase of the algorithm, each node in the network is assigned to its own community. Then for each node i, the change in modularity is calculated for removing i from its own community and moving it into the community of each neighbor j of i. This value is easily calculated by two steps:

(1) Removing i from its original community.

(2) Inserting i to the community of j.

The second phase of the algorithm groups all of the nodes in the same community and builds a new network where nodes are the communities from the previous phase. Self-loops now represent weighted edges between communities, represent any links between nodes of the same community on the new community node, and links from multiple nodes in the same community to a node in a different community. Once the new network is created, the second phase has ended, and the first phase can be re-applied to the new network.

3.2. K-means Clustering

K-means clustering is a method of vector quantization, originally from signal processing, that aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean. The most common algorithm uses an iterative refinement technique. Its ubiquity is often called "the k-means algorithm"; it is also referred to as Lloyd's algorithm[11], particularly in the computer science community. Given an initial set of k means, the algorithm proceeds by alternating between two steps:

(1) Assign each data point to the cluster with the nearest mean: that with the least squared Euclidean distance.

$$J(V) = \sum_{i=1}^{c} \sum_{j=1}^{c_i} \left\| x_i - v_j^{(t)} \right\|^2 \tag{11}$$

where:

$\left\| x_i - v_j^{(t)} \right\|$ = Euclidean distance between x_i and v_j
k_i, k_j = the number of data points in i^{th} cluster
c = number of cluster centers.

(2) Recalculate means (centroids) for data points assigned to each cluster.

(3) Recalculate the distance between each data point and new obtained cluster centroids.

(4) If no data point was reassigned then stop, otherwise repeat from step (1).

The algorithm has converged when the assignments no longer change. The algorithm is often presented as assigning objects to the nearest cluster by distance. Using a different distance function other than Euclidean distance may prevent the algorithm from converging.

4. Experiments

We implemented our network in the NetworkX package and Python language. All experiments were performed on Google Colab. Besides that, we used Gephi, one of the state-of-the-art tools for the purposed of visualization of the network

4.1. Data

We analyze the "Game of Thornes" network, provided as a dataset from Andrew Beveridge. In the dataset, there are a total of five separate books that are corresponding to five chapters. In this project, we merely our network by using only the first book. This provides us more space to visualize our network better in Gephi. Here's how the author turned this complex saga into a collection of interaction networks. Each link corresponds to an interaction between the two characters. This interaction could be direct or indirect. Here are some of the types of interactions that the author method picks up:

(1) Two characters appearing together in the same location.

(2) Two characters in conversation

(3) One character talking about another character.

(3) One character listening to a third character talk about a second character.

(4) A third character talking about two other characters.

If two characters appear together, then clearly, they are linked. If one character mentioned another character, then that signifies some relationship between them. Again, they should be linked. Likewise, if two characters are spoken of together, they are linked in a third character's mind. For simplicity, the author decided to make these links "undirected" meaning that the links are mutual, even when one character references another character. The weight used in the author dataset means: if two characters each time their names appear "within 15 words" of one another, *e.g.*, character A talk with character B about 15 words, so weight is 1, 30 words weight is 2. Note that the author automatically removed any edges of weight 1 or 2, judging these connections to be incidental.

4.2. Construct the network

After exploring the data, we recognize 187 characters correspond to 187 nodes in the network, and 684 weighted edges, accounting for 7,366 interactions. With some data technique skills, the data collected was very impressive: no duplicate edges, no NA values. Finally, we use the Gephi tool to visualize the network more clearly. Figure 2 shows a visualization in Gephi.

4.3. Experiments on centrality measures

In our experiments about five measures in Section 2, we calculate the top five important characters in our data, or be known as the top five characters with high centrality. Results on Table 6.

Eddard Stark tops all five centrality measures and Robert Baratheon takes second place in all categories. The strong performance by Eddard and Robert reflects the main narrative of the book.

In eigenvector centrality, characters get full credit for the importance of their neighbors. In PageRank centrality, characters get a pro-rated share of each neighbor's importance, depending on the character's connection (character weight). Each centrality measure captures a different dynamic. High eigenvector centrality means that characters have powerful resources that they can attempt to tap in the future. When a character's eigenvector centrality is higher than their PageRank centrality, that character can develop in future volumes. However, that character must exert themselves to capitalize on this potential.

Characters with high betweenness are located on many shortest paths connecting pairs of characters. In other words, they are essential because they bind the various narrative threads together.

4.4. Experiments on community detection

First, we try using modularity clustering in Section 3.1. After detector seven communities in total, we set $K_{cluster} = 7$ in because when using K-mean clustering, we need to initialize the cluster first. Results on Table 7.

Community	Characters (Modularity)	Characters (K-means)
1th	24	10
2th	39	11
3th	23	126
4th	26	14
5th	20	17
6th	27	16
7th	2	3

Table 7: Communities detection in "Games of Thorne"

From Table 7, the number of characters in each community is not too many for modularity clustering. It averages about 23 characters per community. By contrast, the number of characters is concentrated largely on a single community in K-means clustering, as 126 characters of 161 characters are in *3th* community. This is understandable when K-means clustering will compute cluster centroids whenever its character change from a different community, so many characters are gathered around 1 centroid after many iterations.

5. Conclusion

In this project, we proposed three types of centrality measures and two types of community detection. Aiming to analyze from the famous book's attribute data, we propose (1) Degree Centrality, Closeness Centrality, Eigenvector Centrality, Pagerank and Betweenness Centrality finding the important characters, and (2) Modularity Clustering and K-means Clustering in community detection problem.

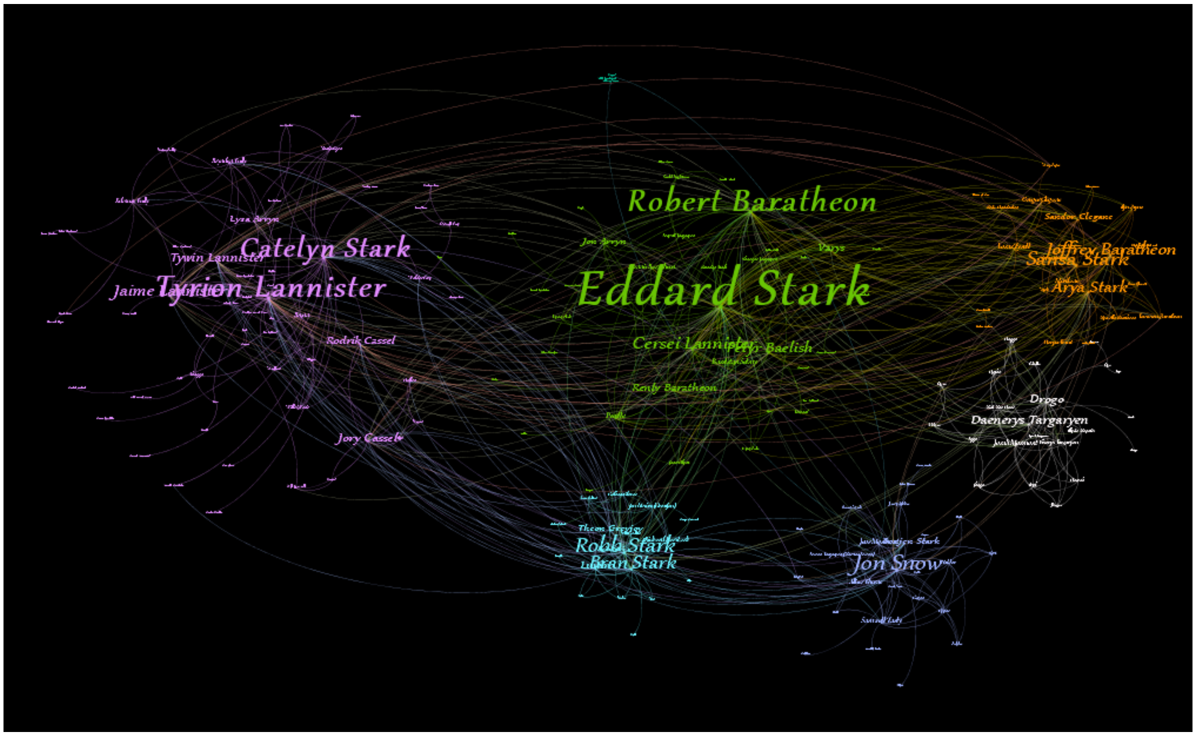

Figure 2: Network of "Game of Thornes" visualize with Gephi. Character names are sized by PageRank centrality. The colors are the number of communities by Modularity Clustering.

Rank	Degree	Closeness	Eigenvector	Pagerank	Betweenness
1	Eddard Stark	Eddard Stark	Eddard Stark	Eddard Stark	Eddard Stark
2	Robert Baratheon	Robert Baratheon	Robert Baratheon	Robert Baratheon	Robert Baratheon
3	Tyrion Lannister	Tyrion Lannister	Sansa Stark	Jon Snown	Tyrion Lannister
4	Catelyn Stark	Catelyn Stark	Tyrion Lannister	Tyrion Lannister	Jon Snow
5	Jon Snow	Robb Stark	Joffrey Baratheon	Catelyn Stark	Catelyn Stark

Table 6: Top five important characters in "Games of Thorne"

References

[1] M. A. Beauchamp. An improved index of centrality. *Behavioral science*, 10(2):161–163, 1965.

[2] P. Bedi and C. Sharma. Community detection in social networks. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 6(3):115–135, 2016.

[3] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008.

[4] P. Bonacich. Power and centrality: A family of measures. *American journal of sociology*, 92(5):1170–1182, 1987.

[5] K.-C. Chang, K. Pearson, and T. Zhang. Perron-frobenius theorem for nonnegative tensors. *Communications in Mathematical Sciences*, 6(2):507–520, 2008.

[6] A. Clauset, M. E. Newman, and C. Moore. Finding community structure in very large networks. *Physical review E*, 70(6):066111, 2004.

[7] S. Fortunato. Community detection in graphs. *Physics reports*, 486(3-5):75–174, 2010.

[8] L. C. Freeman. A set of measures of centrality based on betweenness. *Sociometry*, 40(1):35–41, 1977.

[9] L. C. Freeman. Centrality in social networks conceptual clarification. *Social networks*, 1(3):215–239, 1978.

[10] M. Girvan and M. E. Newman. Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12):7821–7826, 2002.

[11] S. Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982.

[12] F. D. Malliaros and M. Vazirgiannis. Clustering and community detection in directed networks: A survey. *Physics reports*, 533(4):95–142, 2013.

[13] M. E. Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical review E*,

74(3):036104, 2006.

[14] M. E. Newman. The mathematics of networks. *The new palgrave encyclopedia of economics*, 2(2008):1–12, 2008.

[15] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

[16] M. A. Porter, J.-P. Onnela, and P. J. Mucha. Communities in networks. *Notices of the AMS*, 56(9):1082–1097, 2009.

[17] G. Sabidussi. The centrality index of a graph. *Psychometrika*, 31(4):581–603, 1966.

[18] S. Wasserman and K. Faust. *Social Network Analysis: Methods and Applications*. Structural Analysis in the Social Sciences. Cambridge University Press, 1994.

[19] C. C. Yang. Chapter 3 - privacy-preserving social network integration, analysis, and mining. In C. Yang, W. Mao, X. Zheng, and H. Wang, editors, *Intelligent Systems for Security Informatics*, pages 51 – 67. Academic Press, Boston, 2013.