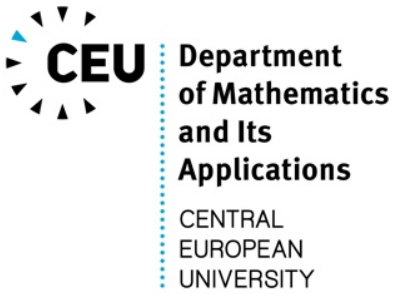
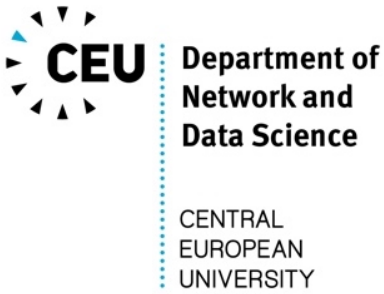
Detection of malicious domains via a large scale network analysis

Matej Kerekrety

Supervised by Gerardo Iñiguez

May 2019





Contents

Abstract

In order to protect users from spam, financial scams or malware, security companies, such as ESET,[1] tend to block dangerous domains and Internet Protocol (IP) addresses. Many of them are chronically known for spreading malware and thus blacklisted, while others are known as clean and whitelisted sources. However, most dangerous domains/IPs are unknown. The aim of this project is to assign a malware probability to domains/IPs using a large scale data on a temporal bipartite network. We model the associated reputation problem as a network interference and graph mining problem, where we construct layers of domains and IP addresses, and seed tthe network with empirical ground truth on malware sources. Then we run the voter model of information spreading to estimate marginal probabilities of domains/IPs being blacklisted. Our analysis provides an intuitive, scalable way of identifying previously unknown, dangerous sources online.

[1] An IT security company ESET which is a leader in antivirus and firewall products, see Introduction section.

Acknowledgements

First of all I would like to express my gratitude to my supervisor Gerardo Iñiguez for his constant support, bright advises and feedback through the whole process of this thesis.

Secondly, I would like to thank to company ESET for giving me the opportunity to work on this project, providing me data and technical support. Namely, I would like to thank to Jindřich Kubec for framing the topic of this project and for his advises in methodology of malicious domain detection. I am also very grateful to Juraj Šustek, who helped me in dealing with large dataset. I must sincerely thank Robert Paško for his help with implementing the model on real big data. Additionally, I would like to thank to Jakub Daubner, Martin Lackovič and Katarina Mayer for their advises, comments and feedback through the whole process of this thesis.

Finally, I must express my gratitude to Rebeka Nagyova, who helped me dealing with hardest time through the whole process of writing this thesis.

Introduction

Security companies offering antivirus and firewall solution, such as ESET, use various techniques to detect malicious content and catch it before it harms the user's machine. Most of these techniques are based on a precise understanding of malware software, collection of samples and comparison to the newly catched malware. Another novel trend in malicious domain and URL detection is to study the lexical patterns in domains and Uniform Resource Locator (URL) strings [17].

This thesis aims to approach this problem from a different direction. We represent the structure of the internet as a Passive Domain Name Server[2] (PDNS) graph which consists of hostnames or domains and the hosting IP addresses.

In order to improve the antivirus software, users are voluntarily sending information about threat detections catch on their machines. These detections contain information on the URL where the file was spotted, the corresponding domain, IP address and a time stamp.

In order to recognize domains and IP addresses which are known to be a source of malware, they are labeled as blacklisted, while others may be trusted and labeled as whitelisted. For simplicity we ignore other possible labelling such as phishing or Potentially unwanted application[3]. Labels are assigned based on both publicly known lists, such as *alexa.com*, and internal classifiers. However we do not have such information about 99% of domains and hosts.

In light of these issues, this thesis proposes a solution which assigns reputation scores to unknown domains and IPs in order to be blacklisted or whitelisted. We constructed a bipartite network of domains and IP addresses, seeded it with minimal ground information and propagated the information from the known nodes to the unknown ones.

The thesis has three chapters. The first chapter is a theoretical introduction, where we define our problem, introduce the voter model and discuss its theoretical properties. The second chapter deals with data and general statistics, where we describe the data used and it's parameters. The third chapter contains the main results of the project,where we present our results and the performance of

[2]Passive Domain Name Server is a system which records Domain Name Server (DNS) resolution data about the DNS severs and IP address hosts.

[3]*Potentially unwanted application* (PUA), as some software companies have a business strategy such that their products are automatically downloaded as a person visits their website. Antivirus software tend to block them as they look suspicious.

the voter model on our dataset.

Related work

There has been significant research done in the area of network interference for detecting malware and malicious domains. Scientists at Symantec created the Polonium model where they created a bipartite network of machines and files and used a Belief Propagation algorithm to estimate marginal probabilities of files being malicious [14]. Researchers in Hewlett-Packard used Belief Propagation to detect malicious domains on network of hosts (IP addresses) and domains [5]. Cisco researchers contributed in identifying botnet domains via signals transmitted through a tripartite network of machines, host names (domains) and host (IPs) [4].

Scientists have also been studying the voter model for decades. Eguíluz at al. have published several works on the dynamics of the voter model on various types of networks [2, 16, 8, 18]. Mobilia at al. [1] wrote On a role of Zealotry in the voter model, where they analysed the voter model on a complete graph with an initial fixed numbers of zealots[4], while a larger overview on the modelling of social dynamics was conducted by Castellano et al. in Statistical physics of social dynamics [15].

ESET

This project was done in close collaboration with the IT security company ESET which is a leader in antivirus and firewall products. ESET is widely known for antivirus programs such as ESET NOD32 or ESET Internet Security, it serves more than 110 millions users worldwide. As a proof of proficiency in cyber security, ESET has the longest unbroken run of VB100 awards [5] for malware detection of all Internet security vendors in the world.

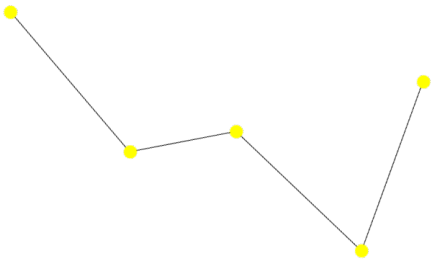


Figure 1: Cassiopeia constellation.

[4] **Zealot** - *a person who has very strong opinions about something, and tries to make other people have them too* (dictionary.cambridge.org)

[5] Virus Bulletin is an independent testing for security software.

The Kassiopea

The main contribution of our work is a novel domain reputation model Kassiopea built on a customized configuration of the voter model. Howerver, most of the similar works such as Symantec Polonium Technology [14] or domain reputation from Hewlett-Packard [5] are built on the Belief Propagation model which is a message-passing algorithm for graphical models based on idea of Bayesian networks [9]. We approached our problem from a different angle, as a random stochastic process where we spread the ground information from the known nodes to the unknown one.

The name Kassiopea stands for the fact that the Cassiopeia is a constellation formed by five bright stars which forms a bipartite network, see Figure 1.

Chapter 1

Theoretical framework

In present chapter, we firstly, we characterize the malicious domain problem as a network interference problem. Secondly, we define our network and it's characteristics. Thirdly, we present our version of modified voter model, existence of zealots and susceptible nodes, implementation and realization of the model. As voter model is widely used in physics, we define macroscopic parameters of *magnetization* and *density*.

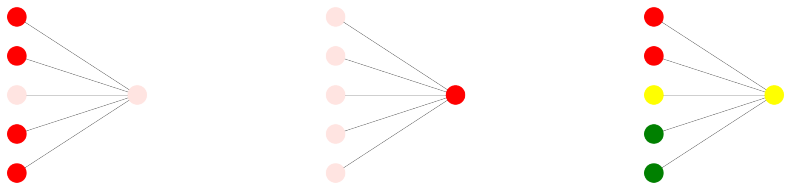


Figure 1.1: Propagation of blacklistness / whitelistness between domains and hosts.
Red - *blacklisted*, **green** - *whitelisted*, **light red** - *tend to be blacklisted*, **yellow** - *tend to be both blacklisted and whitelisted*

1.1 Domain blocking

One of the very basic techniques of how to protect people from spam, financial scams, malware software and other nuisances on the internet is to blacklist certain hosts or the domains. When a domain is labeled as blacklisted, users cannot access its content.

There are three empirically accepted hypotheses about the relationship of blacklisted / whitelisted domains and hosts [5, 12, 3, 6]:

1. An unknown domain which lies on the same host with mostly blacklisted domains tends to be blacklisted. *Equally for whitelisted.*
2. All domains on a blacklisted host are blacklisted. *Equally for whitelisted.*
3. A host with mostly blacklisted domains tends to be blacklisted as well. *Equally for whitelisted.*

Figure 1.1 shows how the blacklistness or whitelistness is propagated in domain-host interference.

1.2 Network

Definition 1 *Let $\mathcal{B} = (\mathcal{V}, E)$ be a network of vertices $\mathcal{V}$ and edges E. Let $\mathcal{X}, \mathcal{Y} \subset \mathcal{V}$, $\mathcal{X} \cap \mathcal{Y} = \emptyset$ and every edge connects a node from $\mathcal{X}$ with one node from $\mathcal{Y}$. Then we call $\mathcal{B}$ a bipartite network.*

We modeled the reputation problem as a network interference and graph mining problem, where we constructed a bipartite network with layers of domain and IP addresses. When a domain lies on an IP address, then those nodes are connected with an edge. Figure 1.2 shows a sketch of the two layers - domains and IP addresses.

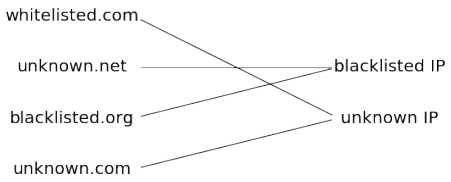


Figure 1.2: Sketch of bipartite network of domains and IP addresses with their states.

In addition, we constructed a projected network to the corresponding bipartite network and ran our reputation model on the projected version as well.

1.3 Bipartite projection

Definition 2 *Let $\mathcal{P} = (\mathcal{X}, \mathcal{F})$ be a network corresponding to the bipartite projection on set $\mathcal{X}$ of network $\mathcal{B}$. Let $N_{\mathcal{B}}(i)$ be a set of neighbors of node i in network $\mathcal{B}$. An edge $e = (x_i, x_j) \in \mathcal{F}$ if $N_{\mathcal{B}}(i) \cap N_{\mathcal{B}}(j) \neq \emptyset$.*

In order to compute a bipartite projection we need to find an adjacency matrix for network $\mathcal{B}$. However, in bipartite network the adjacency matrix has two zero blocks, threfore, we defined a biadjacency matrix B. Then by multiplying $B^T B$ we got an adjacency matrix for the bipartite projected network $\mathcal{P}$ [7, 1, 9]. Figure 1.3 shows the original bipartite projection and its projection on the domain's set.

Figure 1.3: Bipartite network and its corresponding projected network

1.4 The voter model

The voter model is a stochastic process that describes opinion dynamics of adopting states in between agents in the system. The system consists of N agents which are represented as nodes of a network, where those agents are connected by links. The agents can communicate with their neighbors only.

In our implementation, the agents had three states (labels, opinions) which had values $-1, 0, +1$. The agents change their opinions based on rules and processes described in the following sections.

1.4.1 Zealotry and susceptibility of nodes

In our implementation of the voter model in which we had a small fraction[1] of nodes that were zealots, i. e. nodes which do not change their states, while others were susceptible and willing to adopt different states [13].

In the network the ground information was represented by zealot nodes, while susceptible nodes were initially all the remaining unknown nodes. We denoted -1 as a *blacklisted* state, $+1$ as a *whitelisted* state and 0 as an *unknown* state. Initially, zealot nodes have states ± 1 and others i. e. susceptible nodes are in the *unknown* state 0.

At each time step an edge (n_i, n_j) is selected at random and node n_i takes the opinion of n_j in according to the rules.

- Susceptible 0 *unknown* can become $+1$ *whitelisted*
- Susceptible 0 *unknown* can become -1 *blacklisted*

[1] In our network the fraction of zealots is $\sim 1\%$, see chapter Results.

- Susceptible -1 *blacklisted* can become +1 *whitelisted*
- Susceptible +1 *whitelisted* can become -1 *blacklisted*
- Zealot -1 *blacklisted* remains always -1 *blacklisted*
- Zealot +1 *whitelisted* remains always +1 *whitelisted*

Each node, whether it is a susceptible or a zealot, is treated equally and has the same persuasion strength. We denote number of zealots in state ± 1 as $Z_{\pm}$, susceptible in states ± 1 as $N_{\pm}$ and susceptible unknown as N_0. The total number of nodes in the system does not change, $N = N_- + N_0 + N_+ + Z_- + Z_+$, the only changes are states of nodes.

1.4.2 Implementation

The dynamics of our implementation of the voter model consisted of the following steps.

1. Choose an edge at random (n_i, n_j); if both n_i and n_j are zealots, unknown or have the same state, do nothing.
2. If in the selected edge (n_i, n_j) there is at least one susceptible node then it adopts its state from the known susceptible or zealot node, otherwise nothing happens.
3. Repeat steps 1 and 2 *ad infinitum* or until consensus is reached.

Under consensus we understand a stable state when the number of changes is zero or relatively minimal[2].

Other configurations

We also have experimented with different configurations. In the original configuration a susceptible node adopts an option with the probability 1, we tried to improve and set the probability to various values as 0.5 or based on the degree of the j node:

$$p_j = \frac{w_j}{\sum\limits_{i \in N_G(j)} w_i}$$

where w_i is the weight[3] of the edge, $N_G(j)$ means the set of j node's neighbors. Other configurations led to a lower performance.

[2] There are various ways to measure that we have reached consensus. If the magnetization is close to it's extremes ($m(t) \approx \pm 1$), if the relative fraction of active links $\rho(t) \approx 0$ or if a relative number of changes in a block of iteration is small. See chapter Results.

[3] Under 'weight of edge' we understand the number of entries in our dataset which have a connection for the particular domain-IP connection.

1.4.3 Theoretical concept of reputation

In this section, we define the realization of the voter model and characterize reputation. We characterize reputation as statistical dependency between neighbor nodes.

When the voter model reaches the consensus and the process is stopped, call it realization of the voter model. In the end of each realization we remembered the lastly assigned labels. Afterwards, we ran the voter model with the initial setting (zealot and unknown susceptible nodes). During the whole process we ran a number of realizations (50-200) and the final reputation is a long term average of all lastly assigned labels at the end of each realization. In the Results section we present models with various numbers of realizations.

1.4.4 Macroscopic description

In order to define and measure when the model reaches consensus we have to define some macroscopic measures which could be used to describe the dynamics [2, 16, 8, 18].

Magnetization $m(t)$: average state in the network, defined as:

$$m(t) = \frac{1}{N}\sum_{i=1}^{N} x_i$$

$m(t) \in \langle -1, +1\rangle$ If $m(t) = \pm 1$ then the network reached one of the absorbing states ± 1.

Initial densities $\sigma_{\pm 0}$: initial density of zealots states $-1, 0, +1$ in the entire network at time t, defined as:

$$\sigma_{\pm 0}(t) = \frac{Z_{\pm} + N_{\pm 0}}{N}$$

Initial ratios of zealots $z_{\pm}$: initial ratios of zealots states, defined as:

$$z_{\pm} = \frac{Z_{\pm}}{Z}$$

Density of interfaces $\rho(t)$: fraction of links connecting neighbors of different states[4] or a number of *active links* in the network. An *active link* is a link such that one of the nodes could adopt its neighbor state under conditions defined in the implementation of the voter model.

$$\rho(t) = \frac{\#\text{ of actitve links}}{\#\text{ of links in the network}} = \frac{2\Big[\sum\limits_{\langle ij\rangle\in\mathcal{N}} \theta(|x_i - x_j|) - \sum\limits_{\langle ij\rangle\in\mathcal{Z}} \theta(|x_i - x_j|)\Big]}{\langle k\rangle N}$$

[4] We have to distinguish between zealot and susceptible nodes. If two zealots of opposite states are connected, then the link is not active as the state cannot be adopted either way.

Where $\langle ij \rangle$ means neighboring nodes, $\mathcal{N}$ and $\mathcal{Z}$ are the set of all nodes and the set of zealot nodes, respectively, $\theta(x)$ is Heaviside step function, defined as:

$$\theta(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases}$$

$\rho(t) \in \langle 0, 1 \rangle$ If $\rho(t) = 0$, then the network reaches its stable state.

Summary

In this chapter, we proposed three basic hypotheses on domain blocking, we defined and introduced our network and the concept of bipartite network and bipartite projection. We defined the voter model as a stochastic process with updating rules and defined our configuration with three states of nodes $(-1, 0, +1)$ and explained the concept of zealot and susceptible nodes. Lastly, we defined how we computed reputation and defined macroscopic measure as magnetization and density.

Chapter 2

Data and general statistics

In this chapter we present our data and its general statistics. In order to improve antivirus programs, users voluntarily send information about suspicious detection. When a suspiciously looking file is opened on a machine, in order to protect the computer, the antivirus program blocks it and may send a report. If the report is send, that this detection contains information about the URL where the file was spotted, the corresponding domain, the IP address and a time stamp. From this detection data we are able to construct a real-time bipartite network of domains and their hosts.

As data we considered nodes of domains, IP addresses with whitelisted, blacklisted or unknown states. An edge in the bipartite network corresponds to the information that a domain lies on a particular host.

2.1 Bipartite network and bipartite projection

Our original data formed a bipartite network of domains and hosts layers. We tried to run the reputation algorithm on both the original bipartite and projected simple network. Bipartite projection is understood as a network where two nodes are connected if they share a neighbor. In the case of a bipartite network, nodes of two disjoint sets are in the network, while in the projected network, there are only nodes from one set, either hosts or domains.[1]

2.2 Degree distribution

In order to characterize our network, we computed the degree distribution and the degree correlation for bipartite network. Degree distribution is understood as a probability distribution that characterizes the number of links of every

[1] In practice, we can create a projected network where both types of nodes are present but they are disjoint, (a domain cannot be connected with IP, but there is no edge between domain and IP).

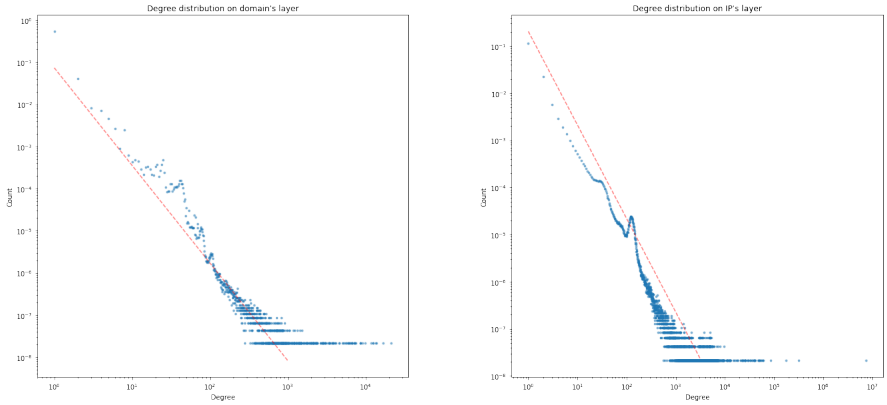


Figure 2.1: Degree distributions of domains and IP layers.

node. Based on figure 2.1 we tested our hypotheses that the network follows power law distribution, Equation 2.1 and 2.2.

$$p(k) \sim k^{-\alpha} \tag{2.1}$$

$$log(p(k)) \sim -\alpha log(k) \tag{2.2}$$

For the whole bipartite network (including both layers), Figure 2.2, has parameter alpha $\alpha = 2.0416$.

As we can see our exponent α is very close to the value 2, it is a ultra small world network, just above the anomalous regime[2]. Then considering Equation 2.3 we see that the first moment is finite, but for $\alpha \in [2, 3)$ the second moment diverges.

$$\langle k^m \rangle = \frac{\alpha - 1}{\alpha - 1 - m} k_{min}^m \tag{2.3}$$

2.3 Degree correlation

Degree correlation determines whether the hubs[3] tend to link with other hubs or are more likely to connect to nodes with lower degree. We categorize the network based on the parameter of distribution of *degree correlation function*, Equation 2.4,

$$k_{nn}(k_i) = \frac{1}{k_i} \sum_{j=1}^{N} A_{ij} k_j \tag{2.4}$$

[2] If the $\alpha \in (1, 2)$ then the network is not graphical. See Barabasi [11]

[3] *Hub* is a node with a relatively high degree.

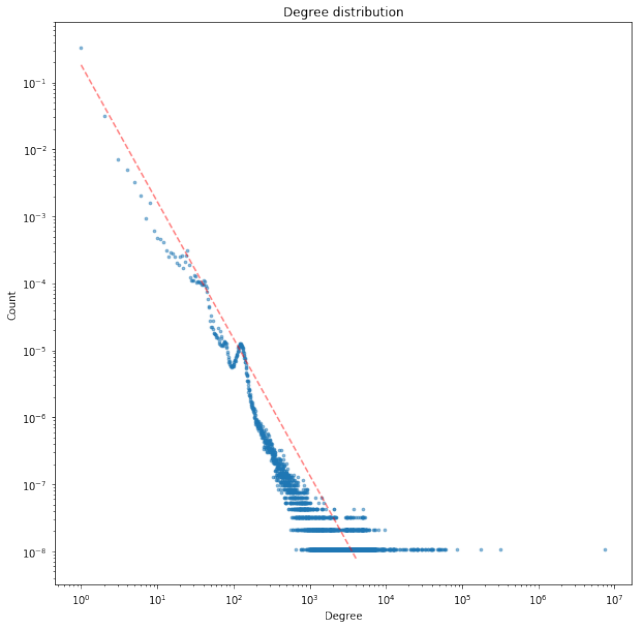


Figure 2.2: Degree distribution of the bipartite network.

where A stands for adjacency matrix of the network and k_i denotes the particular degree [11].

In Figure 2.3, we see that the exponent of power law distribution of *degree correlation function* has a clearly positive value. Therefore, we consider our network as **Disassortative Network**, which means that hubs prefer to link to low-degree nodes [11].

2.4 Macroscopic measures

In Chapter 1, we defined several macroscopic measures which describe the dynamics of the voter model.

Initial magnetization $m(t_0)$:

$$m(t_0) = \frac{1}{N}\sum_{i=1}^{N} x_i \approx 0.009$$

Initial densities $\sigma_{\pm}$):

$$\sigma_+ = \frac{Z_+}{N} \approx 0.0229$$

$$\sigma_- = \frac{Z_-}{N} \approx 0.0140$$

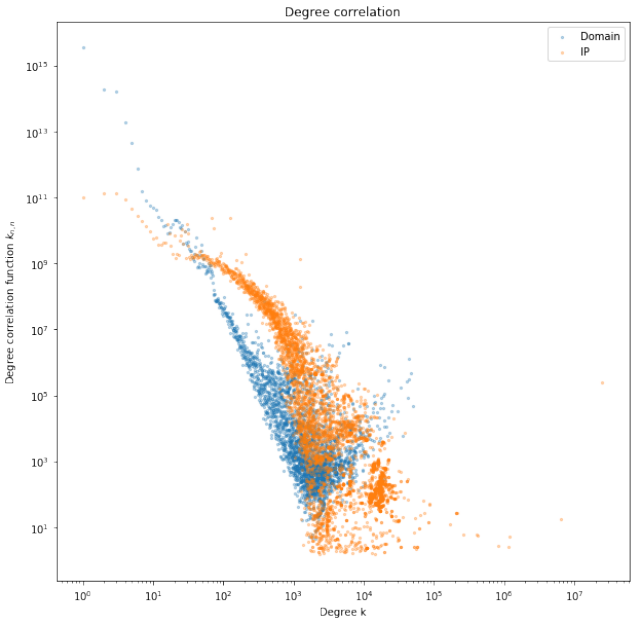


Figure 2.3: Degree correlation of the bipartite network.

Initial ratios of zealots $z_{\pm}$:

$$z_+ = \frac{Z_+}{Z} \approx 0.6207$$

$$z_- = \frac{Z_-}{Z} \approx 0.3793$$

Density of interfaces $\rho(t_0)$:

$$\rho(t_0) = \frac{\#\text{ of actitve links}}{\#\text{ of links in the network}} \approx 0.11$$

We also check the number of links between nodes of two zealots:

$$zealots\_links \approx 5.45 \cdot 10^{-4}$$

Number of links between two unknown nodes in the initial setting is:

$$unknown\_links \approx 0.8897$$

Summary

In this chapter we presented our network and its general statistics. We found that the degree distribution of our network follows power law distribution, due

to the fact that the exponent of power law distribution is an ultra small world network and just above the anomalous regime. Secondly, we observed that it is a disassortative network and the hubs tent to link to low-degree nodes. Finally, we found initial macroscopic measures of the voter model.

Chapter 3

Results

In this chapter, we present the results of the voter model's application to our dataset. Firstly, we describe its technical realization. Secondly, we define how we measured that the model reached a consensus. Thirdly, we present our results on accuracy, True Positive and True Negative rates.

We put significant efforts into studying performance between the bipartite and the projected network. We also tested how the topology and our ground information matters, for which we have created synthetic configuration models with the same properties as our network.

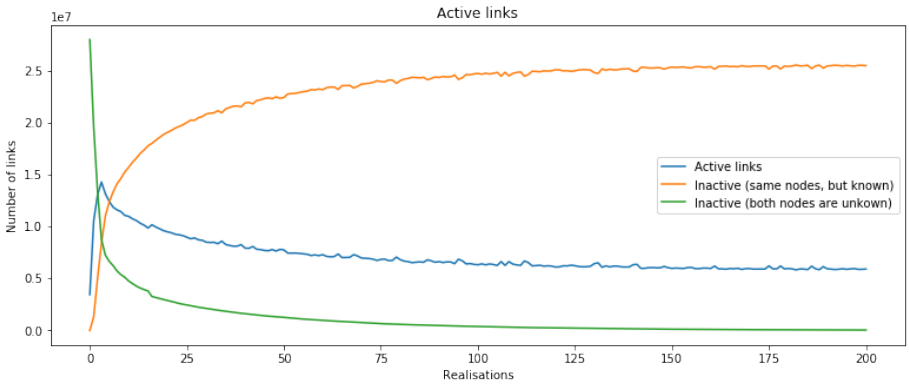


Figure 3.1: Evolution of absolute number of active and inactive links in time.

3.1 Technical realization

The voter model is a simple model and its implementation has the following structure. We define a function *persuade(egde)* which gets edge as an argument, based on the rules mentioned in chapter 1, we choose a node and assign an

adopted label. For implementation see Algorithm 1.

```
Input: edge = (node1, node2) a tuple of two nodes
Output: Update a node's label
Function Persuade(edge):
    if node1 is zealot and node2 is not zealot then
        node2 adopts node1 label
    else if node1 is not zealot and node2 is zealot then
        node1 adopts node2 label
    else if node1 is not zealot and node2 is not zealot then
        if node1 is not unknown and node2 is not unknown then
            choose node that adopts the label at random
        else
            if node1 is unknown and node2 is not unknown then
                node1 adopts node2 label
            else if node1 is not unknown and node2 is unknown then
                node2 adopts node1 label
            else
                both nodes are unknown, do nothing
            end
        end
    end
End Function
```

Algorithm 1: Persuade function

Secondly, we are choosing edges at random in two for loops of size number of batch and number of iterations in batch, respectively. As we can see the complexity $\mathcal{O}(M)$ is linear to maximal number of iteration. For implementation see Algorithm 2.

```
for number of batches do
    for size of a batch do
        choose an edge at random;
        Persuade(edge);
        if the process reached the consensus then
            break;
    end
end
```

Algorithm 2: Realization of the voter model

3.1.1 Network

We have performed our model on various scales of networks. Firstly, we proved our concept on a network of size $9 \cdot 10^4$ nodes and $1.3 \cdot 10^5$ with initial magnetization $m(t_0) = -0.0164$ and relative number of active links $\rho(t_0) = 0.089$. This network is used in sections *Comparison of bipartite and projected network* and *Configuration networks*.

Afterwards, we used a bigger network of size $1 \cdot 10^7$ nodes and $3.1 \cdot 10^7$ with initial magnetization $m(t_0) = 0.009$ and relative number of active links $\rho(t_0) = 0.11$. This network was used for all other analysis.

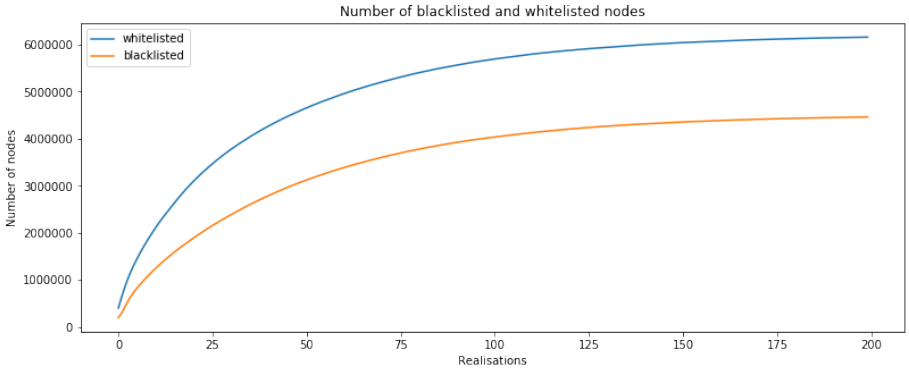


Figure 3.2: Evolution of number of nodes with blacklisted and whitelisted labels in time. (one realization)

3.2 Reaching consensus

In the original configurations of the voter model, where all agents are susceptible and have only two opinions ± 1, there are two stable states, where agents stop changing their opinions[1], which means that all agents have the same -1 or $+1$ opinion[2].

Magnetization $m(t) = \pm 1$ shows if the system reached the absorbing state, where all nodes have the same opinion, while relative fraction of active links $\rho(t)$ describes how many links can an update happen on. $\rho(t) = 0$ corresponds to fully ordered state. In the original model there is a direct relationship:

$$\rho(t) = 0 \Leftrightarrow m(t) = \pm 1$$

For scale-free uncorrelated networks [2] the relationship between magnetization and density relies on $\langle k \rangle$ average degree in the following way:

$$\rho(t) = \frac{\langle k \rangle - 2}{2(\langle k \rangle - 1)} \left(1 - m^2(t)\right)$$

However in our model there are zealots - nodes which never change their opinions. Therefore, $m(t) \neq \pm 1$ will never reach a fully ordered absorbing state and the same goes for the relative fraction of active links, $\rho(t) \neq 0$. Figure

[1] We used to called these states absorbing or fully ordered states.

[2] In particular, this is not true in general. Let's have a graph with two disjointed components where on one component all nodes have -1 opinion while on the second component $+1$.

3.1 shows how the number of active and inactive links evolves in time. We can observe that from a certain point in time, the number of active links stops its decay.

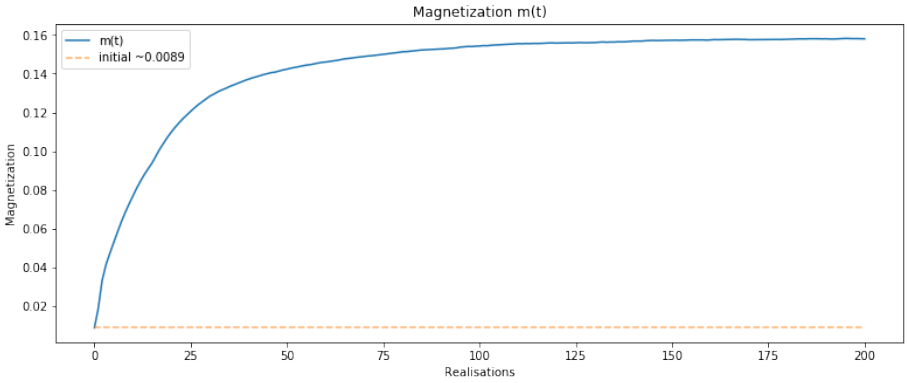


Figure 3.3: Evolution of magnetization $m(t)$ in time. (one realization)

A similar phenomenon can be observed for magnetization $m(t)$. Figure 3.2 shows the absolute number of nodes with blacklisted and whitelisted labels evolving in time. Figure 3.3 shows the relative magnetization evolving in time.

3.2.1 Relative fraction of updates

Based on empirical observation, we estimate the expected number of iterations required to reach consensus M[3], then we split it into b blocks or batches of size M/b. In every batch we count the number of changes and number of active and inactive links. We observe (see Figures 3.1, 3.3, 3.4) that the process tends to saturate at a certain number of active links, magnetization and relative number of changes, respectively.

More interestingly we observe that in each realization of our model, the number of active links, relative fraction of updates and magnetization saturated at a different level. We ran 10 independent realizations of our model.

Figure 3.6 shows paths for magnetization, relative number of updates, number of active links and number of inactive links with known nodes ± 1 for 10 different realizations.

3.2.2 Unknown nodes and links

Another variable to measure is the number of unknown nodes and links[4] where both nodes have not been labeled yet. Figure 3.1 shows the number of unknown links, while Figure 3.5 describes the decay of unknown nodes in the network.

[3]However, the number of edges is about $\sim 10^7$, therefore we estimated that the expected number of iterations would be $\sim 10^8$.

[4]Under unknown link we understand an inactive link where both nodes have state 0.

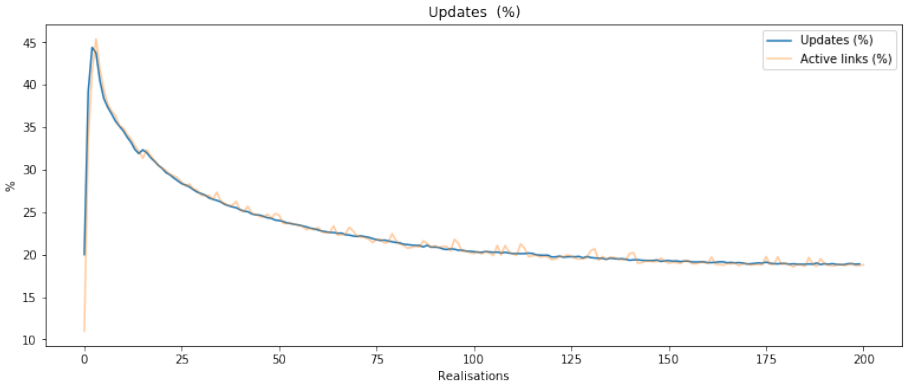


Figure 3.4: Evolution of relative number of updates in time is corresponding to the number of active links. (one realization)

As we can see the decay progresses very quickly, and for unknown nodes it progresses exponentially.

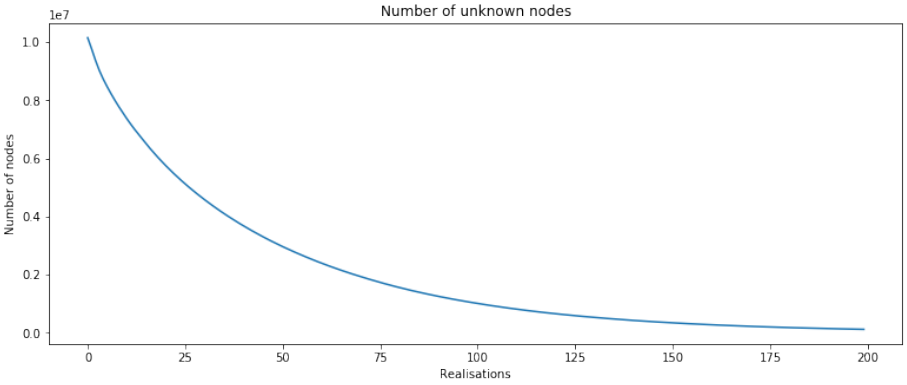


Figure 3.5: Number of unknown nodes in the network. (one realization)

In the end, we stop our model when the relative number of changes starts to saturate. Due to the fact that each realization saturates at a different level, we stop the realization when the derivation[5] is *relatively* equal to zero.

[5] In our discrete case we count cumulative difference.

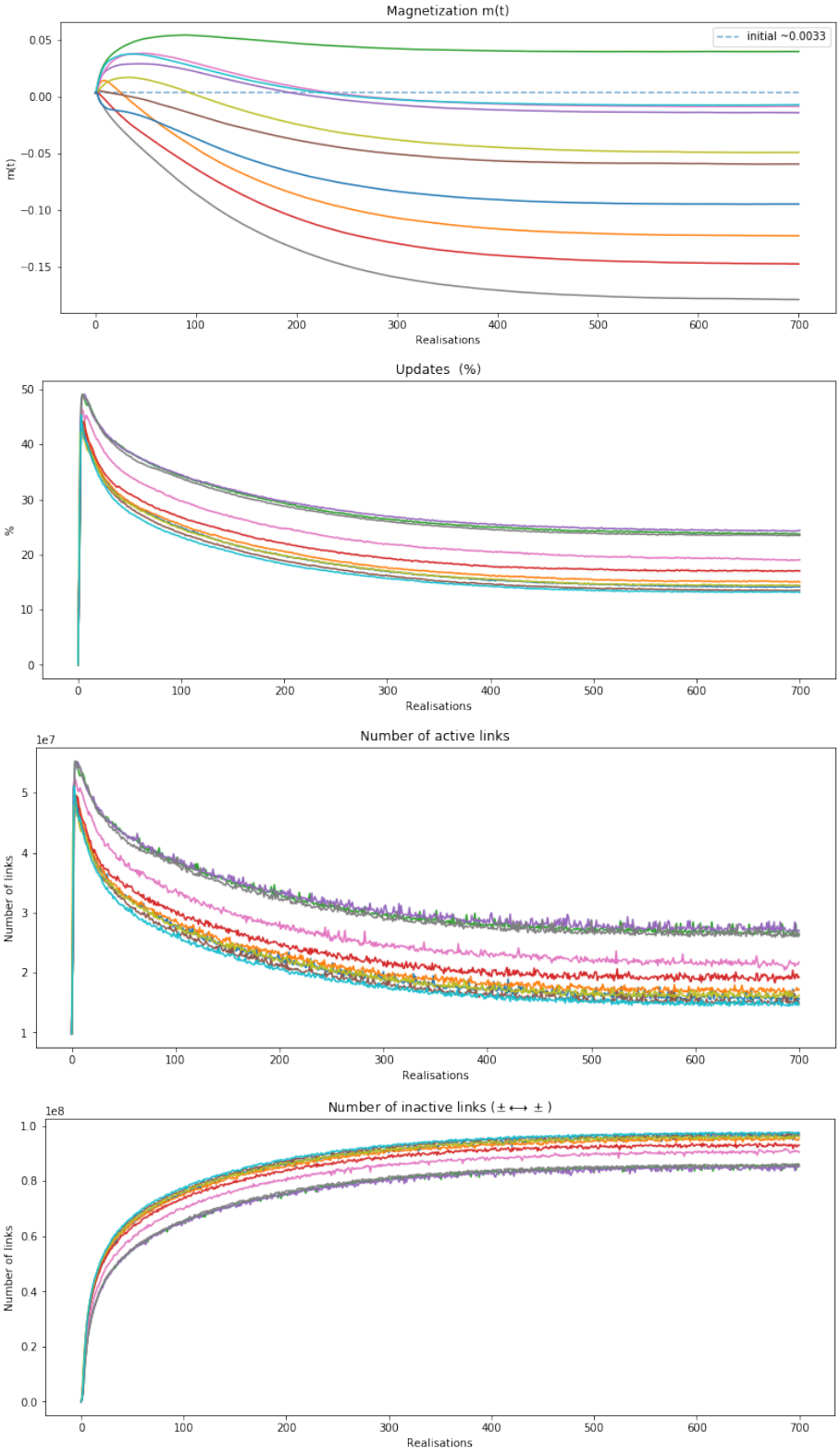


Figure 3.6: How the levels of magnetization, relative fraction of updates, number of active and inactive links saturate on different values for 10 realizations of the voter model.

3.3 Validation

In this section, we present how we tested the accuracy of how well can our model assign reputation and predict new potentially malicious domains. However, our problem does not belong to the traditional machine learning problems where it would be possible to set up training and testing sets. In order to check the performance, we use two validation methodologies.

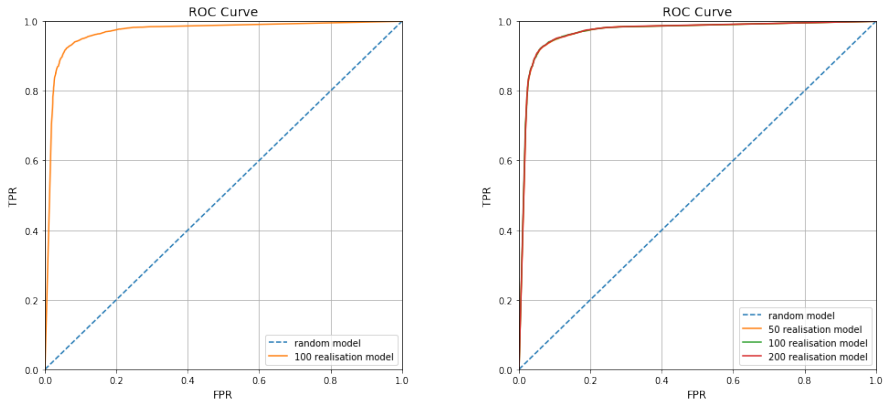


Figure 3.7: True positive rate and false positive rate for domains, using the ground data test set validation. Comparison of three models, varying the number of realizations *50, 100, 200.*

Firstly, we tried the methodology used in Manadhata et al. [5] and Faloutsos et al. [14] which shows the robustness of the model where we took $1/10^{th}$ from our ground data (zealot nodes) and labeled them as *unknown*, while the rest $9/10^{ths}$ of the ground data with the susceptible nodes was the training set. Using this validation methodology we achieved an overall accuracy of 92.64% with True Positive Rate (TPR) of 93.38% and True Negative Rate (TNR) of 91.86%[6]. Figure 3.7 shows the ROC curve, while figure 3.8 shows the probability distributions of *blacklisted / whitelisted* classes.

To investigate how the number of realizations changes the accuracy, we performed three different models with 50, 100 and 200 realizations, respectively. The difference in accuracy was minimal, around ±0.01%.

Secondly, we used valuation methodology which is based on the fact that domains change their states in time. It is possible for a domain to be labeled as unknown, but in a few days it may be *blacklisted / whitelisted* [7]. Thanks to

[6] Under positive outcome or when the rate is close to +1 we understand *whitelisted*, while negative outcome, rate is close to -1 we understand *blacklisted*.

[7] As we mentioned in the Introduction, domains and IP addresses are *blacklisted / whitelisted* based on external and internal classifiers.

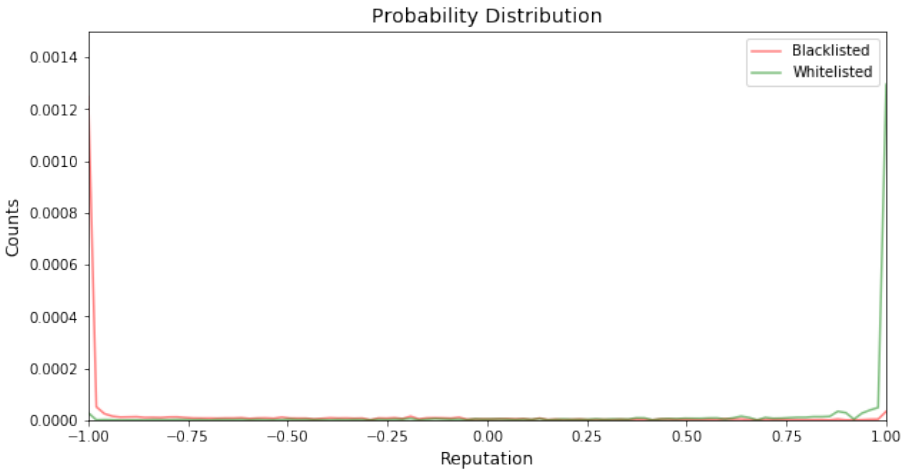


Figure 3.8: Probability distribution of *blacklisted* and *whitelisted* classes.

that we can select domains which were unknown when the model was run but now they are labeled. Using this validation methodology we achieved a general accuracy of 78.78% with TPR of 52.38% and TNR of 79.09%, see Figure 3.9.

3.4 The voter model on bipartite and projected network

In other works, the voter model was run on regular latices [8], complete networks [16], scale free uncorrelated networks [2] or on complete bipartite [18] networks, but none of them had such complicated structure and configuration as our network and model. Therefore, we tried to simplify our network and calculate the projection[8].

The opposite was true, as we found that the projected network is more dense, the accuracy is lower, there is loss of information and in terms of detection of malicious domains it is better to consider the original bipartite network.

3.4.1 Difference in computation efficiency

On a sample of bipartite network with the number of nodes $N \approx 9 \cdot 10^4$ and number of edges $|E| \approx 1.3 \cdot 10^5$, the projected network had number of edges $|\mathcal{F}| \approx 3.2 \cdot 10^7$.

The hypothesis of why this is true relies on the fact that our bipartite network is *disassortative*, which means that hubs tend to link to low-degree nodes. Figure 3.10 shows a network where an IP address (hub) is linked to five domains (low-degree nodes); the network will be projected as a complete graph of five domains.

[8]The projection is defined in chapter 1 as well as the projection algorithm.

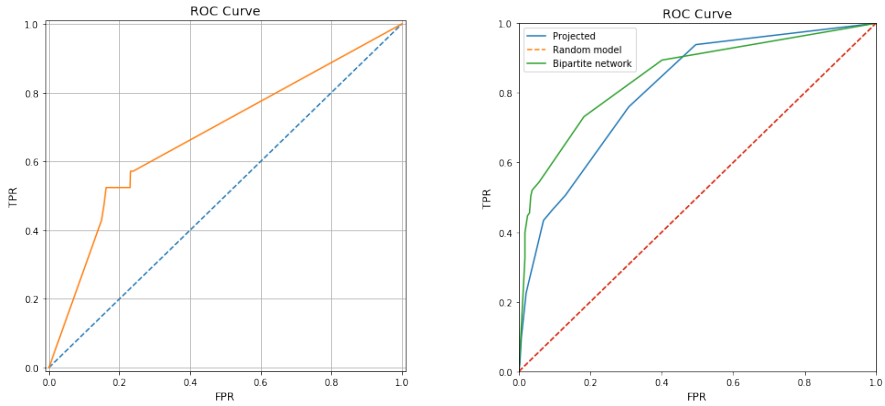


Figure 3.9: a) True positive rate and false positive rate for domains, using the newly labeled test set validation. b) Comparison of ROC curves of bipartite and projected network.

However, there is a high number of high cliques and many more edges, therefore the whole model runs for longer, plus a separate, not negligible, computational capacity for projecting is required.

3.4.2 Difference in accuracy

Higher complexity is not the only problem with the projected network. We have observed that the accuracy is lower (~ 0.15) compared to the original bipartite network. Figure 3.9 shows the difference on two ROC curves.

3.4.3 Difference in domain detection

In chapter 1, we introduced three hypotheses of how the blacklistness / whitelistness is being inherited. While the first and third are kept in the projected network, the second one (*i.e. All domains on blacklisted host are blacklisted.*) is omitted by the removal of the IP layer.

However, because the computation complexity is higher, accuracy is lower and there is information loss, we decided to use the bipartite projected network in our model.

3.5 Configuration models and synthetic networks

In order to understand the significance of our network's topology we built two synthetic models which have the same properties as our network. We kept the

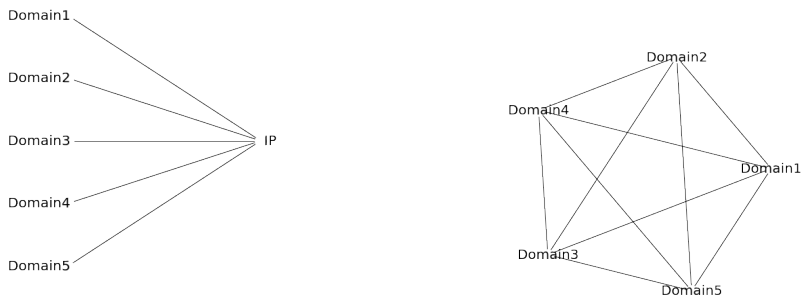


Figure 3.10: Bipartite network and its corresponding projected network.

degree distribution, number of zealots and their ratios (number of blacklisted / whitelisted nodes). The models have the following configurations:

Randomly shuffled zealots: We kept the network structure and topology as it is. Nodes and links were in the original configuration. We also kept the susceptibility of nodes, but we shuffled the zealot's labels at random.

Randomly shuffled links: We kept the nodes susceptibility and labels as in the original network but we shuffled the links at random keeping the degree distribution of each node.

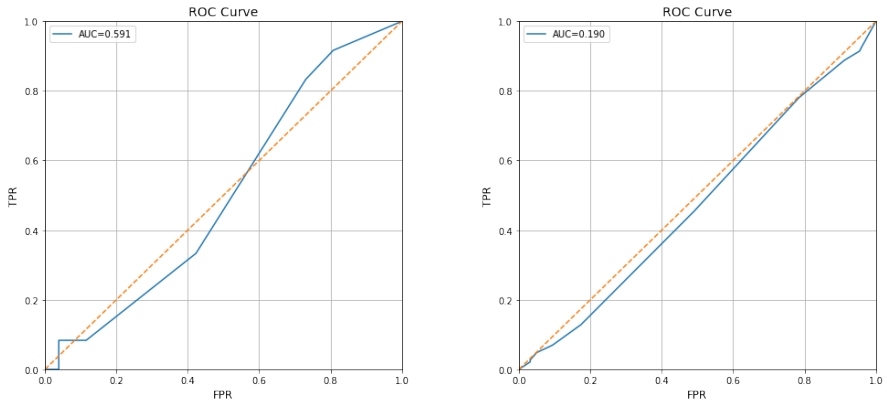


Figure 3.11: ROCs for the configuration modes: a) randomly shuffled zealot's labels b) randomly allocated edges, keeping the degrees.

On both configuration networks ran the voter model and checked the accuracy and ROC curves. Figure 3.11 shows that the performance on those

synthetic networks is close to a random.

Based on these configuration models we can assume that the topology and the position of zealots is significant.

Summary

In this chapter we presented results of our reputation model Kassiopea. We demonstrated the technical implementation of the algorithm. We observed the dynamics of the voter model on our network and based on that we defined when it reached the consensus.

The reputation model attained a high True positive and True Negative rate, 93,38% and 92.64% respectively. We discussed the difference in running the model on a bipartite and a projected network. Additionally, we tested how the model behaves on a synthetic network and we concluded that the topology and position of zealot nodes have a significant impact on the results.

Conclusion

In this project, we aimed to detect potentially malicious domains by assigning reputation of being blacklisted to unknown domains. We transformed it into a large scale network mining and interference problem and we proposed and implemented a novel Kassiopeia model. The Kassiopeia model is a special configuration of the voter model, which uses the existence of zealots as a ground information which is spreads to the remaining unknown nodes.

We performed our Kassiopeia model on large-scale network. The results show that Kassiopea attained a high True Positive rate TPR of 93.38% and True Negative rate TNR of 91.86% with overall accuracy of 92.64%.

Kassiopeia now assigns reputation to tens of millions of domains and predicts new malicious domains on a daily basis.

In addition, we described the stochastic dynamics of the voter model on our network. We defined consensus and characterized when the system reaches it.

We believe that our work has contributed in both to computer security research and to the mathematical understanding of a novel configuration of the voter model.

Bibliography

[1] Suman Banerjee, Mamata Jenamani, Dilip Kumar Pratihar: *Algorithms for Projecting a Bipartite Network*, (August 2017)
https://www.researchgate.net/publication/323067832_Algorithms_for_projecting_a_bipartite_network

[2] Federico Vazquez, Víctor M Eguíluz: *Analytical solution of the voter model on uncorrelated networks*, (June 2008)
https://iopscience.iop.org/article/10.1088/1367-2630/10/6/063011/pdf

[3] Justin Ma, Lawrence K. Saul, Stefan Savage, Geoffrey M. Voelker: *Beyond Blacklists: Learning to Detect Malicious Web Sites from Suspicious URLs*, (2009)
http://cseweb.ucsd.edu/ jtma/papers/beyondbl-kdd2009.pdf

[4] Dhia Mahjoub, David Rodriguez: *Beyond lexical and PDNS: using signals on graphs to uncover online threats at scale*,
https://www.virusbulletin.com/uploads/pdf/magazine/2017/VB2017-Mahjoub-Rodriguez.pdf

[5] Pratyusa K. Manadhata, Sandeep Yadav, Prasad Rao, and William Horne: *Detecting Malicious Domains via Graph Inference*,
http://www.covert.io/research-papers/security/Detecting malicious domains via graph inference.pdf

[6] Leyla Bilge, Engin Kirda, Christopher Kruegel, and Marco Balduzzi: *EXPOSURE: Finding Malicious Domains Using Passive DNS Analysis*, (7 February 2011)
https://sites.cs.ucsb.edu/ chris/research/doc/ndss11_exposure.pdf

[7] Tao Zhou, Jie Ren, Matus Medo, Yi-Cheng Zhang: *How to project a bipartite network?*, Physical Review E 76, 046115 (31 Jul 2007)
https://arxiv.org/pdf/0707.0540.pdf

[8] Juan Fernández-Gracia, Krzysztof Suchecki, José J. Ramasco, Maxi San Miguel, Víctor M. Eguíluz: *Is the Voter Model a model for voters?*, (June

2014)
https://arxiv.org/pdf/1309.1131.pdf

[9] Kevin P. Murphy *Machine Learning: A Probabilistic Perspective (Adaptive Computation and Machine Learning series)*. The MIT Press; 1 edition (August 24, 2012)

[10] Mark Newman. *Networks: An Introduction*. Oxford University Press; 1 edition (May 20, 2010).

[11] Barabási, A.-L., Pósfai, M. (2016). *Network science*. Cambridge: Cambridge University Press. ISBN: 9781107076266 1107076269

[12] Mark Felegyhazi, Christian Kreibich, Vern Paxson: *On the Potential of Proactive Domain Blacklisting*, (2010-04-27)
https://www.usenix.org/legacy/event/leet10/tech/full_papers/Felegyhazi.pdf

[13] M. Mobilia, A. Petersen, S. Redner: *On the Role of Zealotry in the Voter Model*, J. Stat. Mech. P08029, (2 Aug 2007)
https://arxiv.org/pdf/0706.2892.pdf

[14] Duen Horng Chau, Carey Nachenberg, Jeffrey Wilhelm, Adam Wright, Christos Faloutsos: *Polonium: Tera-Scale Graph Mining and Inference for Malware Detection,*
https://www.cc.gatech.edu/ dchau/polonium/polonium_sdm2011.pdf

[15] Claudio Castellano, Santo Fortunato, Vittorio Loreto: *Statistical physics of social dynamics*, Reviews of Modern Physics 81, 591-646 (2009)
https://arxiv.org/pdf/0710.3256.pdf

[16] Juan Fernández-Gracia: *Updating rules and the voter model*, (January 2011)
http://digital.csic.es/bitstream/10261/46143/1/tesinaMaster.pdf

[17] Hung Le, Quang Pham, Doyen Sahoo, Steven C.H. Hoi: *URLNet: Learning a URL Representation with Deep Learning for Malicious URL Detection*, (2 Mar 2018)
https://arxiv.org/pdf/1802.03162.pdf

[18] V. Sood, Tibor Antal and S. Redner: *Voter models on heterogeneous networks*, (2008)
https://www.maths.ed.ac.uk/ antal/Mypapers/voter08.pdf