MCCXXII

Università degli Studi di Padova

Dipartimento di Matematica "Tulio Levi-Civita"
Corso di Laurea in Matematica

Statistical tests for botnet detection in a network

Relatore:
Prof. Marco Formentin

Laureando:
Leonardo Solidoro

Matricola: 2072945

Anno Accademico 2024/2025
19/09/2025

Contents

Introduction

There are many real-world examples of networks, such as the Internet, wireless computer networks or social networks. Each can be described by its elements and their interconnections. In many cases, complex networks are modeled using random graphs, where the vertices of the graph represent the network's elements and the edges that form the connections are determined based on underlying probabilistic rules.

Networks might contain a small number of vertices that behave as malicious actors. In the examples mentioned, these could be infected servers or computers spreading malware over the Internet or fake user profiles sending spam and phishing messages within a social network. This set of anomalous vertices is referred to as the botnet. Detecting these unwanted or malicious elements is therefore of great interest.

This problem can be addressed with various approaches [4]. In this thesis, the problem of detecting a botnet in a network will be formulated from a statistical point of view, using only the structural information of the network.

The organization of the thesis is as follows.

In Chapter 1 we introduce some basic statistical concepts, such as the estimator of a statistical model's parameters and, in particular, a consistent estimator.

Furthermore, we define the statistical problem of hypothesis testing, which is the framework used to tackle the problem considered.

In Chapter 2 we formalize the problem of detecting a botnet in a network as a hypothesis testing problem. The analysis is based on a single observation of the network's structure, represented as a graph. The null hypothesis states that the network is free of a botnet and is modeled as a random geometric graph, which is the type of random graph that we adopt and it will be described in detail. In contrast, the alternative hypothesis is that in the random geometric graph there is a small subset of vertices, representing the botnet, which ignore the geometric structure and connect uniformly to every other vertex with a certain probability.

We propose two different tests, the isolated star test and the average distance test, for the purpose of detecting the presence of such a botnet and thus deciding between the hypothesis and the alternative based on the observed graph. The first test is based on the intuition that the botnet vertices form large isolated stars that are not present in random geometric graphs. The second test relies on the idea that the presence of a botnet significantly shortens the average graph distance.

Under appropriate assumptions on the model parameters, we will show that both tests can identify the correct hypothesis, with high probability, as the graph size approaches infinity. We also show that this result is optimal, which means that no test can reliably detect a botnet if these assumptions are not met.

Finally, we propose two consistent estimators for the model parameters required by our tests, since we do not have direct knowledge of their true values as we can only observe one realization of the random graph.

In Chapter 3 we accompany our theoretical results for the asymptotic regime with a simulation study to showcase the performance of our tests on graphs of finite size. We use synthetically generated data to compare them.
Our observations show that both tests always correctly identify graphs without a botnet. However, the isolated star test performs better in detecting the presence of a botnet than the average distance test. Furthermore, the estimators of the model parameters correctly identified the true values in our simulations.

The definitions presented in Chapter 1 are adapted from [5]. The results presented in Chapter 2 can be found in [1]. The simulations in Chapter 3 and the figures displayed are the products of original code, which is provided in the Appendix.

Chapter 1

Basic notions in statistics

In this chapter, we introduce and define some basic statistical concepts concerning the estimation of model parameters and the statistical problem of hypothesis testing, fundamental notions for the analysis of the model in the next chapters. The definitions presented in this chapter are adapted from [5].

1.1 Estimation in a parametric statistical model

Definition 1.1. A statistical model is a triple $(\chi, \mathcal{F}, \mathbb{P}_\theta : \theta \in \Theta)$ consisting of:

- χ is the sample space,
- $\mathcal{F}$ is a σ-algebra on χ,
- $\{\mathbb{P}_\theta : \theta \in \Theta\}$ is a class of probability measures on $(\chi, \mathcal{F})$.

We can describe a random phenomenon with an appropriate statistical model. The process of observing such a phenomenon is characterized by a random variable X with values in the sample space χ. The distribution $\mathbb{P}_\theta$ of the random variable is supposed partially unknown. The parameter θ, which labels the distribution, is only assumed to be in the parameter space Θ. A realization of X is called the measured or observed value as it is the specific value obtained by an observation.
Given a number of random observations, one intends to identify the probability measure that generated the observed data, this is called estimation.
We will be dealing with different probability measures, therefore we write $\mathbb{E}_\theta$ for the expectation with respect to the probability measure $\mathbb{P}_\theta$.
Statistical models can have the following additional properties.

Definition 1.2. A statistical model $(\chi, \mathcal{F}, \mathbb{P}_\theta : \theta \in \Theta)$ is called:

- a parametric model if $\Theta \in \mathbb{R}^d$ for some d.
- a standard model if either χ is a Borel subset of $\mathbb{R}^n$, $\mathcal{F}$ is the Borel σ-algebra restricted to χ and every $\mathbb{P}_\theta$ has a Lebesgue density or χ is discrete and $\mathcal{F} = \mathcal{P}(\chi)$; then every $\mathbb{P}_\theta$ has a discrete density.

We now give the definition of an estimator.

Definition 1.3. Let $(\chi, \mathcal{F}, \mathbb{P}_\theta : \theta \in \Theta)$ be a statistical model and $(\Sigma, \mathcal{S})$ an arbitrary event space

- A statistic is any measurable function $S : \chi \to \Sigma$

- Let $\tau : \Theta \to \Sigma$. A statistic $T : \chi \to \Sigma$ is called an estimator if it is used to estimate the value of $\tau(\theta)$

For every $x \in \chi$, we want to construct an estimator in such a way that $T(x)$ is close to the true $\tau(\theta)$. In this way, we can estimate the true value of θ, therefore determining the probability measure $\mathbb{P}_\theta$ that governs the random experiment and generated the observed data.
We will now define some common requirements on the estimators, the most suitable of which might depend on the situation.

Definition 1.4. Let $(\chi, \mathcal{F}, \mathbb{P}_\theta : \theta \in \Theta)$ be a statistical model and $\tau : \Theta \to \mathbb{R}$ a real characteristic. An estimator $T : \chi \to \mathbb{R}$ of $\tau(\theta)$ is called unbiased if

$$\mathbb{E}_\theta(T) = \tau(\theta) \text{ for all } \theta \in \Theta.$$

Otherwise the bias of T at θ is

$$B_T(\theta) = \mathbb{E}_\theta(T) - \tau(\theta).$$

An unbiased estimator avoids systematic errors by being on average close to the true value.
A commonly used performance measure is the mean squared error

$$\mathcal{E}_T(\theta) = \mathbb{E}_\theta\Big[(T - \tau(\theta))^2\Big] = Var_\theta(T) + B_T(\theta).$$

To keep this error as small as possible both bias and variance must be minimized simultaneously. One way to do so is by considering an unbiased estimator T of $\tau(\theta)$ that also has minimal variance between other unbiased estimators. Such an estimator is called a best estimator.

Definition 1.5. Let $(\chi, \mathcal{F}, \mathbb{P}_\theta : \theta \in \Theta)$ be a statistical model and $\tau : \Theta \to \mathbb{R}$ a real characteristic. For each $n \geq 1$, let $T_n : \chi \to \mathbb{R}$ be an estimator of $\tau(\theta)$ based on the first n observations. The sequence $(T_n)_{n\geq 1}$ of estimators of $\tau(\theta)$ is called consistent if

$$T_n \xrightarrow[n\to\infty]{\mathbb{P}_\theta} \tau(\theta) \text{ for all } \theta \in \Theta,$$

by definition of convergence in probability this is

$$\mathbb{P}_\theta(|T_n - \tau(\theta)| \leq \epsilon) \xrightarrow[n\to\infty]{} 1 \text{ for all } \epsilon > 0, \theta \in \Theta.$$

This is a performance criteria that concerns the long-term behavior of repeated observations, since we want the values of the estimator T_n to be typically close to the true value, for large n.
Considering the asymptotic behavior of a sequence of estimators T_n, we can also say that it is asymptotically unbiased if

$$\mathbb{E}_\theta(T) \xrightarrow[n\to\infty]{} \tau(\theta) \text{ for all } \theta \in \Theta.$$

1.2 Statistical hypothesis testing

We now give an introduction to the statistical problem of hypothesis testing. Its objective is to develop a clear decision rule for the rational behavior in random situations. First, a hypothesis is formulated about the random mechanism of the observations, then the observation results are used to decide between accepting or rejecting the hypothesis. A decision procedure for this problem is called a test. Accepting the hypothesis does not imply that it is true, but just that the observations support it.
We consider an appropriate statistical model $(\chi, \mathcal{F}, \mathbb{P}_\theta : \theta \in \Theta)$ in order to describe the situation. The parameter set Θ can be decomposed into two subsets Θ_0 and Θ_1:

- $\theta \in \Theta_0$ if for θ the hypothesis is true.
- $\theta \in \Theta_1$ if for θ the hypothesis is false.

We observe that $\Theta_0 \cup \Theta_1 = \Theta$ and $\Theta_0 \cap \Theta_1 = \emptyset$.
With this decomposition, we say that the null hypothesis is $H_0 : \theta \in \Theta_0$ and is tested against the alternative hypothesis $H_1 : \theta \in \Theta_1$. Usually, the null hypothesis is considered the expected normal case.

Definition 1.6. If Θ_0 contains just a single value of θ, a single distribution, then H_0 is a simple hypothesis. If Θ_0 contains more than one value of θ, then H_0 is a composite hypothesis. The analog can be said for the alternative hypothesis H_1.

Definition 1.7. Let $(\chi, \mathcal{F}, \mathbb{P}_\theta : \theta \in \Theta)$ be a statistical model and assume $\Theta = \Theta_0 \cup \Theta_1$ is a decomposition of Θ into the null and alternative hypothesis
We define a test of H_0 against H_1 as a function

$$\psi : \chi \to [0, 1]$$

where, if $x \in \chi$ is an observation, then

- $\psi(x) = 0$ when the null hypothesis H_0 is accepted
- $\psi(x) = 1$ when the null hypothesis is rejected and the alternative is supposed to be true, based on x
- $0 < \psi(x) < 1$ when there is not a clear-cut decision and the null hypothesis is rejected with probability $\psi(x)$

If $\psi(x) = 0$ or 1 for all $x \in \chi$, then ψ is called a non-randomized test procedure, otherwise it is called randomized.
In the first case ψ divides the sample space into two complementary regions. The critical region or the rejection region is defined as $C = \{x \in \chi : \psi(x) = 1\}$. Therefore, a non-randomized test ψ is simply the indicator function of the critical region C.
The probability of rejecting H_0 is

$$\mathbb{E}_\theta(\psi).$$

There are two types of error that can occur when performing a test: rejecting the null hypothesis when it is true (type-1 error) and accepting it when it is false (type-2 error). Ideally, we want to design a test that keeps the probabilities of these errors at a minimum. However, when the number of observations is given, both probabilities cannot be controlled simultaneously since they generally work against each other and there is a need to strike an appropriate balance between the two.

Definition 1.8. The function

$$G_\psi : \Theta \to [0, 1],\ G_\psi(\theta) = \mathbb{E}_\theta(\psi)$$

is called the power function of the test ψ.
For $\theta \in \Theta_1$, $G_\psi(\theta)$ is called the power of the test ψ against the alternative θ.

The power of the test is the probability of rejecting the null hypothesis when it is false. Therefore, the probability of a type-2 error, that is, the probability of accepting the null hypothesis when it is false is equal to $1 - G_\psi(\theta)$ for $\theta \in \Theta_1$.
The most popular method of striking a balance between the probabilities of the two types of error is to select a significance level, $0 < \alpha < 1$, and impose that the probability of the type-1 error should not exceed α, that is

$$G_\psi(\theta) \leq \alpha \text{ for all } \theta \in \Theta_0.$$

Subject to this condition, it is desired to minimize the probability of type-2 error or, equivalently, to maximize the power of the test

$$G_\psi(\theta) \text{ for all } \theta \in \Theta_1.$$

The choice of the level of significance α is arbitrary, but standard values are $\alpha = 0.01$ or 0.05.
Note that this condition introduces an asymmetry in the treatment of the null and alternative hypotheses. In most problems such an asymmetry can be natural, since one of the two errors might be less desirable in some sense. For this reason, one generally arranges the null and alternative hypotheses so that type-1 error is most avoided.
We now give a formal definition of what we discussed above.

Definition 1.9. The size $\alpha(\psi)$ of the test ψ is the worst-case probability of a type-1 error

$$\alpha(\psi) = \sup_{\theta \in \Theta_0} G_\theta(\psi).$$

We say that a test has a level of significance α, or that it is a level α test, if it satisfies

$$\sup_{\theta \in \Theta_0} G_\theta(\psi) \leq \alpha.$$

The previous requirements lead to the following definition.

Definition 1.10. A test ψ is called a most powerful (MP) test of level α if its size is at most α and for any other test ϕ of level α we have

$$G_\psi(\theta) \geq G_\phi(\theta) \text{ for all } \theta \in \Theta_1.$$

An accurate discussion of some classic examples of most powerful tests in a standard model can be found in [5]. However, these approaches are not applicable to our problem, since it is not a standard model.
Now we define the total probability of making an error, regardless of whether it is a type-1 or type-2 error.

Definition 1.11. We define the worst-case risk of a test ψ as

$$R(\psi) = \sup_{\theta \in \Theta_0} \mathbb{E}_\theta(\psi) + \sup_{\theta \in \Theta_1} 1 - \mathbb{E}_\theta(\psi)$$

This definition implicitly assumes that both types of error have equal cost or importance. More generally, this sum is a special case of a weighted sum of errors.
We consider the asymptotic behavior of a sequence of tests as the sample size tends to infinity.

Definition 1.12. For each sample size n, let ψ_n be a test.
We say that the sequence of tests $(\psi_n)_{n=1}^\infty$ is asymptotically powerful when

$$R(\psi_n) \xrightarrow[n \to \infty]{} 0.$$

Finding an asymptotically powerful sequence of tests essentially means minimizing the total probability of making an error. Therefore, such a sequence identifies the correct hypothesis in the regime $n \to \infty$.
This performance criterion will be used for our tests because we are interested in considering an arbitrarily large sample size.

Chapter 2

Botnet detection: model and results

2.1 Introduction

We look at the problem of detecting a botnet in a network. A network is often described in terms of a large number of vertices and their connection, a botnet is a small number of these vertices that connect differently than the rest. This is a potentially malicious anomaly in the network, thus it is of great interest detecting its presence.
A real-world example might be a social network, where every user is connected to others they interact with. In this case, the botnet represents fake users that send spam or phishing messages to everyone else.
We formalize this problem from a statistical point of view as a hypothesis testing problem where we observe a single instance of a graph which represents the structural information of the network analyzed.
The null hypothesis H_0 is that the graph is a realization of a random geometric graph and represents a network without the presence of a botnet. In contrast, the alternative hypothesis H_1 is that in the random geometric graph there is a small subset of vertices, representing the botnet, which ignore the geometric structure and connect uniformly to every other vertex with a certain probability.
We then propose two tests to decide whether we can accept H_0 or whether we have to reject it, based on the given graph.

2.2 Model formulation

In this section, we formalize the model discussed above. First, we describe the type of random graph that we will use to model the network: the random geometric graph.
We begin by giving some useful basic notation for graphs, which will also be used in the next sections.

- Let $G = (V, E)$ be an undirected simple graph with no self loops, where $V = \{1, ..., n\}$ is the set of vertices with $|V| = n$ and $E \subset \{\{u, v\} \in V \times V | u \neq v\}$ is the set of edges.
- Given $v \in V$, we use $N(v) = \{u \in V | \{u, v\} \in E\} \subset V$ to denote the subset of its neighbors.
- Given $v \in V$, we use $\delta(v) = \{e = \{u, v\} \in E | u \in V\} \subset E$ as the set of edges incident in v and $\deg_V(v) = |\delta(v)|$ as the degree of v.

- Given $u, v \in V$, we use $u \leftrightarrow v$ when $\{u, v\} \in E$ and we say that the vertices are connected.
- Given $u, v \in V$, we use $u \leftrightsquigarrow v$ if there exist a path of connected vertices between u and v.
- We say that the graph G is connected if $\forall u, v \in V$ we have that $u \leftrightsquigarrow v$.
- Given $u, v \in V$ with $u \leftrightsquigarrow v$, let $D_G(u, v)$ denote the graph distance between u and v, that is the length of the shortest path in the graph G that connects u and v.
- A subgraph of G is $H = (V', E')$ with $V' \subset V$ and $E' \subset E$ such that H is still a graph. We use $H = G_{V'}$ to indicate the subgraph induced by $V' \subset V$.

The random geometric graph is constructed considering n vertices, each is then given a random position on a d-dimensional unit torus and two vertices are connected if their distance on the torus is less than a given radius r. We will now explain in detail what this means.
Let $T^d := [0, 1]^d{}_{/\sim}$ be the d-dimensional unit torus, on T^d the Euclidean distance is the function:

$$D_T(x, y) = \sqrt{\sum_{j=1}^{d} \min(|x_j - y_j|, 1 - |x_j - y_j|)^2} \text{ , for } x, y \in T^d. \tag{2.1}$$

It is the minimum distance function accounting for the fact that the shortest path between two points could be warping around the identified edges of the torus rather than the regular Euclidean distance.
T^d is refereed to as the embedding space.
For all $v \in V$ vertex, let $X_v \in \mathbb{R}^d$ be a random vector uniformly distributed on T^d. This means that its components $X_v = (X_{v,1}, \ldots, X_{v,d})$ are random variables i.i.d. such that

$$X_{v,j} \sim Unif([0, 1])$$

for $j \in 1, \ldots, d$.
Two vertices $u, v \in V$ are connected, $u \leftrightarrow v$, when

$$D_T(X_u, X_v) \leq r.$$

The average edge probability is defined as

$$p = \mathbb{P}(u \leftrightarrow v) = \mathbb{P}(D_T(X_u, X_v) \leq r).$$

Note that, with this construction, the probability $0 < p < 1$ of u, v being connected is equal to the probability of a random point X_v landing on a d-dimensional ball of radius r, $B^d(r)$, inside T^d.
Then we have the explicit relation

$$p = \frac{V_{B^d(r)}}{V_{T^d}} = \frac{(\sqrt{\pi} r)^d}{\Gamma(d/2 + 1)} \tag{2.2}$$

with $V_{T^d} = 1$ by construction and $V_{B^d(r)} = \frac{(\sqrt{\pi} r)^d}{\Gamma(d/2+1)}$ by direct integration in spherical coordinates, where $\Gamma(\cdot)$ denotes the Gamma function.
From this relation we can also find the value of r which makes exactly p the probability of two vertices being connected with the distance rule given above, this is

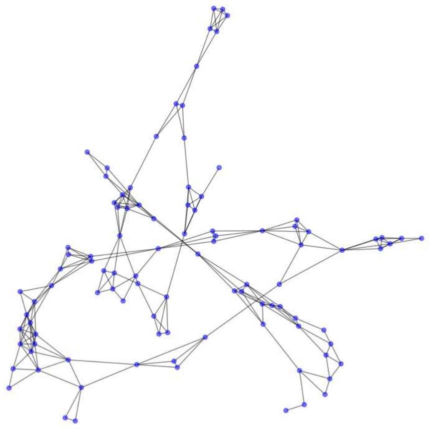


Figure 2.1: A realization of a random geometric graph $\mathbb{G}(n, d, p)$ with n = 100, d = 2 and p = 0.05

$$r = \frac{1}{\sqrt{\pi}} \left[p \, \Gamma(d/2 + 1)\right]^{\frac{1}{d}} . \tag{2.3}$$

Hence, a random geometric graph is completely described by n the number of vertices, d the dimension and p the average edge probability. We will use $\mathbb{G}(n, d, p)$ to denote it. Figure 2.1 shows an example of such a random graph.
The expected number of connections for every vertex in $G(n, d, p)$ is clearly $(n-1)p$. For large graphs, we can consider $n \simeq n - 1$, therefore, we will refer to np as the average degree.
Given a vertex $v \in V$, the distribution of its degree is binomial $deg_V(v) \sim Bin(n, p)$, this is

$$\mathbb{P}(deg_V(v) = k) = \binom{n}{k} p^k (1-p)^{n-k}$$

This model was chosen because it allows us to work with graphs that have a geometric structure. In fact, a random geometric graph can be visualized as a d-dimensional unit torus with n vertices inside, where each vertex is the center of a sphere of radius $r/2$ and two vertices are connected when their spheres overlap.
Generating a random geometric graph by checking if the distance in the embedding space between every pair of vertices is smaller than the given radius r can be done in $O(n^2)$. However, this is not feasible on very large graphs. Instead, we will use a KD-tree, a data structure from [9], to efficiently find all pairs whose distance is at most r.
Now we can define the null and alternative hypotheses considered in the hypothesis testing problem.

Definition 2.1. The observed graph G respects the null hypothesis, H_0, if it is a realization of a d-dimensional random geometric graph $\mathbb{G}(n, d, p)$ where n is the number of vertices and p the average edge probability.

The random geometric graph is the model considered for the network without the presence of the botnet.

Definition 2.2. The observed graph G respects the alternative hypothesis, H_1, if it is a realization of a d-dimensional random geometric graph, except for a small subset of

vertices, called the botnet $B \subset V$ with $|B| = k$, that ignore the geometric structure and instead connect to every other vertex independently with probability p.
In this case we say that the graph is a realization of $\mathbb{G}(n, d, p; k)$

In this model each pair of vertices $u, v \in V \setminus B$ is connected when $D_T(X_u, X_v) \leq r$, the same as in the previous model, while the remaining vertices in the botnet B are connected to every other vertex in V with probability p.

Remark 2.1. With this construction the average degree is the same under both hypothesis, that is np. This assumption rules out trivial scenarios where the botnet can be detected by calculating the edge degree. In fact, if a botnet vertex were to establish significantly more connections than other vertices, it would be easily identifiable. Furthermore, in practice, botnets are designed to imitate the behavior of regular nodes within the network.
The difference between the two hypotheses is that in the null hypothesis the vertices are only connected when their distance in the embedding space T^d is smaller than the connection radius r, as formally stated previously, while under the alternative hypothesis the vertices of the botnet can connect to every other vertex, even far ones, with probability p. We are going to exploit this with the two tests.

A visual comparison of the model under the null and alternative hypothesis is given in Figure 2.2. However, the locations of the vertices in the embedding space, as shown in the figure, are not available to us since we can only observe which vertices are connected in the graph.

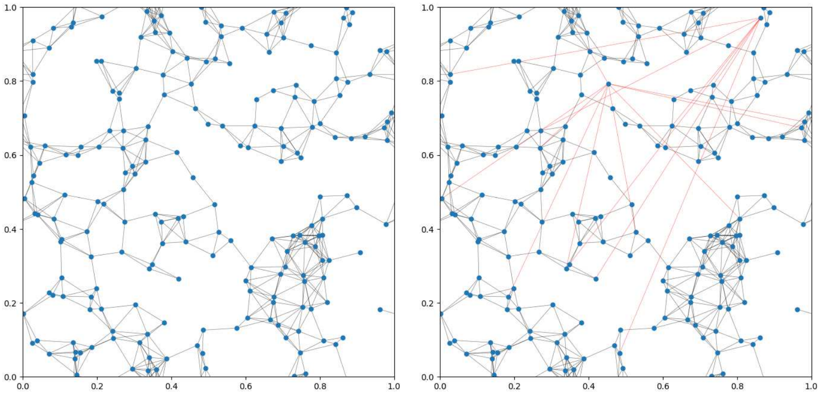


Figure 2.2: Example of the model under the null and alternative hypothesis in dimension $d = 2$, with $n = 200$ vertices, average degree $np = 6$ and $k = 2$ botnet vertices. The graphs are represented in the embedding space which in this case is the unit square with continuous boundaries, consistently with the distance function considered. The botnet is highlighted in red.

Remark 2.2. We assume that the dimension $d \geq 2$ remains fixed, while the edge probability p and the botnet size k depend on the graph size n.
We require the average degree to be sublinear in n: $np = o(n)$; otherwise the resulting graph would be too connected for a large n. This condition by definition is $p \to_{n\to\infty} 0$.
We also require $1 = O(np)$, otherwise the resulting graph will not be dense enough and most of the vertices will be isolated for a large n. By definition, it is the same condition as $\lim_{n\to\infty} \frac{1}{np} = l < \infty$.
In conclusion, putting together all the conditions on p we have $np \sim n^{1-\alpha}$ with $\alpha \in]0, 1]$.

Finally, we assume that $1 \leq k \leq o(n)$ to avoid having a botnet size that grows as much as n.
From now on, we will consider the asymptotic regime $n \to \infty$ to find the necessary condition for detecting a botnet.

We recall that a non-randomized test is a function

$$\psi : G \to \{0, 1\}$$

with $\psi(G) = 1$ when the null hypothesis H_0 is rejected and $\psi(G) = 0$ when it is accepted.

Definition 2.3. From Definition 1.11 we define the worst-case risk of a test in our specific setting as

$$R(\psi) = \mathbb{P}_0(\psi(G) \neq 0) + \max_{B \subset V, |B|=k} \mathbb{P}_B(\psi(G) \neq 1) \tag{2.4}$$

where $\mathbb{P}_0$ is the distribution of the random graph under the null hypothesis and $\mathbb{P}_B$ is the distribution of a graph, with botnet B, under the alternative hypothesis.

Let $(\psi_n)_{n=1}^{\infty}$ be a sequence of tests, we say it is asymptotically powerful when

$$R(\psi_n) \to 0.$$

Therefore, such a sequence identifies the underlying model correctly in the regime $n \to \infty$.
Before we introduce our two tests, we define the threshold in terms of the model parameters below which no test can be asymptotically powerful.
Intuitively, this happens when the expected number of edges connected to the botnet vertices is bounded, since in this case there is a positive probability that all the vertices of the botnet are isolated and therefore it would be impossible to distinguish the two hypotheses. This is formalized in the following theorem.

Theorem 2.1. *If $npk = O(1)$, then $R(\psi_n) > 0$ $\forall \psi_n$. This means that no test can be asymptotically powerful.*

Proof. We consider a similar version of the problem in which the vertices of the botnet $B \subset V$ are known, with p the average edge probability, $|B| = k$ and $|V| = n$. This means that the alternative hypothesis no longer has a randomly chosen botnet and the problem corresponds to a hypothesis test between two simple hypothesis.
We consider the risk in this setting

$$R^*(\psi) = \mathbb{P}_0(\psi(G) \neq 0) + \mathbb{P}_B(\psi(G) \neq 1).$$

This is clearly a lower bound for the worst-case risk $R(\psi)$ in Definition 2.3.
Considering the likelihood ratio

$$L(g) = \frac{\mathbb{P}_B(G = g)}{\mathbb{P}_0(G = g)},$$

the following result holds for every test

$$R^*(\psi) \geq \sup_{\tau > 0} \left\{ \frac{\tau}{\tau + 1} \mathbb{P}_0(L(g) \geq \tau) \right\} \tag{2.5}$$

A detailed proof of this can be found in [12, Proposition. 2.1].
Since $R(\psi) \geq R^*(\psi)$, to show that no test is asymptotically powerful it is sufficient to show that $\mathbb{P}_0(L(g) \geq \tau) > 0$, for some τ independent of n.
We define the event $A = \{\text{all vertices in B are isolated in the graph G}\}$.

We recall that $G_{V\setminus B}$ indicates the subgraph induced by the subset of vertices $V \setminus B$.
If we consider a graph g such that $\mathbb{P}_0(G = g|A) > 0$, this means that the graph could be sampled with positive probability from the null hypothesis when all vertices in B are isolated, it follows that

$$\begin{aligned}\mathbb{P}_0(G = g) &\le \mathbb{P}_0(G_{V\setminus B} = g_{V\setminus B})\\ &= \mathbb{P}_B(G_{V\setminus B} = g_{V\setminus B})\\ &= \frac{\mathbb{P}_B(G = g)}{(1-p)^{k(k-1)/2}(1-p)^{(n-k)k}},\end{aligned}$$

where the term at the denominator is the probability that in a generic graph, with distribution $\mathbb{P}_B$, the botnet is the one considered. This is equivalent to the probability that B vertices are isolated in a generic graph G, which happens when the vertices in B don't have edges between them and the remaining $n-k$ vertices are not connected to any of the k vertices.
We than obtain

$$\begin{aligned}L(g) = \frac{\mathbb{P}_B(G = g)}{\mathbb{P}_0(G = g)} &\ge (1-p)^{(n-k)k+k(k-1)/2}\\ &= e^{\log(1-p)[(n-k)k+k(k-1)/2]}\\ &= e^{-(p-o(p))[(n-k)k+k(k-1)/2]}\\ &= e^{-(1-o(1))npk},\end{aligned}$$

because we assumed that $p \to 0$ as $n \to \infty$ and $k = o(n)$.
This shows that $L(g)$ remains strictly positive under the assumption $npk = O(1)$. Therefore, if we choose $\tau > 0$ small enough we have that

$$\mathbb{P}_0(L(g) \ge \tau|A) = 1$$

for n large enough.
In this case we have that

$$\mathbb{P}_0(L(g) \ge \tau) \ge \mathbb{P}_0(L(g) \ge \tau|A)\mathbb{P}_0(A) = \mathbb{P}_0(A).$$

Now to estimate this probability we consider the vertices of the botnet $\{v_1, \ldots, v_k\} = B$ and the remaining $\{w_1, \ldots, w_{n-k}\}$. We reveal the vertices in B one at the time and we consider q_j the probability that v_j is not connected to any of the previous vertices given that all these previously revealed vertices are not connected. For $j \in \{1, \ldots, k\}$ we obtain

$$\begin{aligned}q_j &= \mathbb{P}_0(v_j \not\leftrightarrow v_i \forall i \in \{1, \ldots, j-1\}|v_i \not\leftrightarrow v_l \forall i < l\ \forall l \in \{1, \ldots, j-1\})\\ &\ge 1-(j-1)p.\end{aligned}$$

Finally, we observe that

$$
\begin{aligned}
&\mathbb{P}_0(L(g) \geq \tau) \\
&\geq \mathbb{P}_0(A) \\
&\geq \mathbb{P}_0(v_j \not\leftrightarrow v_i \forall i < j \ \forall j \in \{1, \ldots, k\}, v_j \not\leftrightarrow w_l \forall j \in \{1, \ldots, k\} \forall l \in \{1, \ldots, n-k\}) \\
&\geq (\prod_{j=1}^{k} q_j)(1-kp)^{n-k} \\
&= [\prod_{j=1}^{k}(1-(j-1)p)](1-kp)^{n-k} \\
&= e^{-(1+o(1))npk},
\end{aligned}
$$

which remains strictly positive as $n \to \infty$ under the assumption $npk = O(1)$. Using this result in (2.5) we have that

$$R(\psi) \geq R^*(\psi) > 0 \ \ \forall \psi.$$

Therefore, no test can be asymptotically powerful.

□

2.3 Tests for the detection problem

From now on, we will consider the asymptotic regime $npk \to \infty$.
In this section we present the two proposed tests: the isolated star test and the average distance test. We then show that above the considered threshold they are both asymptotically powerful.

2.3.1 Isolated star test

Given a vertex $v \in V$, we use $N(v) = \{u \in V | \{u, v\} \in E\} \subset V$ to denote the subset of its neighbors.

Definition 2.4. Let $S(v) \subseteq N(v)$ be the largest independent set in the subgraph of G induced by $N(v)$. We call $S(v)$ the isolated star of the vertex $v \in V$.

This means that every $u \in S(v)$ is connected by an edge to v ($\forall u \in S(v)$ we have $\{u, v\} \in E$), but any pair of vertices in $S(v)$ is not connected by an edge ($\forall u, w \in S(v)$ we have $\{u, w\} \notin E$).
For this test we will use the kissing number k_d, which is defined as the maximum number of non-overlapping spheres of the same radius that can be placed tangent to a central sphere in dimension d.
Now we can define the first test to decide whether we can accept the null hypothesis or we have to reject it, which means that we have detected the presence of a botnet in the observed graph G.

Definition 2.5. Let k_d be the kissing number in dimension d. The isolated star test rejects the null hypothesis for a given graph G, $\psi(G) = 1$, when

$$\max_{v \in V} |S(v)| > k_d. \tag{2.6}$$

Therefore, for this test we compute the size of the isolated star at every vertex of the given graph G and then detect the presence of a botnet when we find an isolated star larger than k_d.
Next we present the main result of this section, where we give conditions for this test to be asymptotically powerful.

Theorem 2.2. *If* $npk \to \infty$ *then the isolated star test from Definition 2.5 is asymptotically powerful.*

Proof. We observe that the isolated star test has type-1 error equal to zero, that is $\mathbb{P}_0(\psi(G) \neq 0) = 0$, since it is impossible for a graph G sampled under H_0 to have an isolated star larger than the kissing number k_d.
This is because of the underlying geometry of our model, where every vertex $v \in V$ is assigned a random vector $X_v \in \mathbb{R}^d$ uniformly distributed on T^d the d-dimensional unit torus and two vertices $u, v \in V$ are connected if $D_T(X_u, X_v) \leq r$. Hence, this is equivalent to a model where every vertex is the center of a sphere of radius $r/2$ on a d-dimensional torus and two vertices are connected when their spheres overlap. Since the kissing number is the maximum number of non-overlapping spheres of the same radius that can be placed tangent to a central sphere in dimension d, under these hypotheses, there can be no isolated star bigger than k_d in G.
To show that this test is asymptotically powerful we now have to show that, under H_1, the probability of correctly rejecting the null hypothesis H_0, that is $\mathbb{P}_B(\psi(G) = 1)$, tends to one. From the definition of the test, this is equivalent to showing that the probability of having an isolated star larger than k_d tends to one under the alternative hypothesis.
Let $\deg_{V \setminus B}(v)$ be the number of non-botnet neighbors of a vertex $v \in B$.
First we show that if $\deg_{V \setminus B}(v) \geq k_d + 1$ then v will form an isolated star of size $|S(v)| \geq k_d + 1$ with high probability.
Given $v \in B$, we define the event

$$D(v) = \{\deg_{V \setminus B}(v) \geq k_d + 1\}$$

Conditionally on this event, let $\{v_1, \dots, v_{k_d+1}\} \subseteq V \setminus B$ be a subset of $k_d + 1$ non-botnet neighbors of v. We reveal these vertices one at a time and consider q_j the probability that v_j is not connected to any of the previous vertices given that all these previously revealed vertices are not connected. For $j \in \{1, \dots, k_d + 1\}$, we obtain the following

$$\begin{aligned} q_j &= \mathbb{P}_B(v_j \nleftrightarrow v_i, \forall i \in \{1, \dots, j-1\} | D(v), v_i \nleftrightarrow v_l, \forall i < l \ \forall l \in \{1, \dots, j-1\}) \\ &\geq 1 - (j-1)p, \end{aligned}$$

where we observe that conditioning on $D(v)$ does not affect the distribution of the other vertices, because $v \in B$ is a botnet vertex and does not respect the geometric properties of the graph.
Then we have

$$\begin{aligned} \mathbb{P}_B(|S(v)| \geq k_d + 1 | D(v)) &\geq \mathbb{P}_B(v_j \nleftrightarrow v_i \forall i < j, j \in \{1, \dots, k_d + 1\} | D(v)) \\ &= \prod_{j=1}^{k_d+1} q_j \\ &\geq \prod_{j=1}^{k_d+1} (1 - (j-1)p) \\ &\geq (1 - k_d p)^{k_d} \to 1 \end{aligned}$$

as $p \to 0$ in our assumptions and k_d is constant.
Hence, any vertex of the botnet $v \in B$ with $\deg_{V \setminus B}(v) \geq k_d + 1$ will form an isolated star of size $|S(v)| \geq k_d + 1$ with probability tending to one.
For the second part of the proof we have to show that, with high probability, there exists a botnet vertex $v \in B$ that has $\deg_{V \setminus B}(v) \geq k_d + 1$.
We observe that $\deg_{V \setminus B}(v)$ are i.i.d. random variables distributed as $Bin(n-k, p)$. Now, because of our assumptions on the asymptotic regime, we have that $npk \to \infty$, $k = o(n)$, $p \to 0$ and $1 = O(np)$. It follows that the expected non-botnet degree is either $(n-k)p \to \infty$ or $(n-k)p = O(1)$.
When $(n-k)p \to \infty$: every vertex $v \in B$ will eventually have $\deg_{V \setminus B}(v) \geq k_d + 1$ with high probability.
For the other case, we obtain and bound the error in approximating $\deg_{V \setminus B}(v)$ by the Poisson distribution using the Stein-Chen method [2, pp. 31], it follows that

$$\begin{aligned} || \deg_{V \setminus B}(v) - Poi((n-k)p) ||_{TV} &\leq 2 \min\left(1, \frac{1}{(n-k)p}\right) \Big[(n-k)p - Var(Bin(n-k,p)) \Big] \\ &\leq 2p \to 0. \end{aligned} \tag{2.7}$$

We also note that $(n-k)p \nrightarrow 0$ since we assumed $1 = O(np)$.
Therefore, if $(n-k)p = O(1)$, by (2.7) there is a positive probability that $\deg_{V \setminus B}(v) \geq k_d + 1$ independently for each $v \in B$. Since in this case $|B| = k \to \infty$, to satisfy the assumption $npk \to \infty$, there exists at least one vertex v in the botnet with non-botnet degree greater than the kissing number, with high probability. This is

$$\mathbb{P}_B(D(v)) \to 1.$$

Finally, combining this with the first part, where we showed that if $\deg_{V \setminus B}(v) \geq k_d + 1$ then v will form an isolated star of size $|S(v)| \geq k_d + 1$ with high probability, we can conclude

$$\mathbb{P}_B(|S(v)| \geq k_d + 1) \to 1.$$

Therefore, this test is asymptotically powerful. ☐

2.3.2 Average distance test

We will define the second test to decide whether we can accept or reject the null hypothesis based on the observed graph G.
For this test, we require that p is large enough to ensure that the graph is connected with high probability.

Definition 2.6. Given $u, v \in V$ with $u \leftrightsquigarrow v$ two connected vertices, let $D_G(u,v)$ be the graph distance between u and v, that is, the length of the shortest path in the graph G that connects u and v. We define the average graph distance as

$$D_G^{avg}(G) = \frac{\sum_{1 \leq u < v \leq n} \mathbb{1}_{\{u \leftrightsquigarrow v\}} D_G(u,v)}{\sum_{1 \leq u < v \leq n} \mathbb{1}_{\{u \leftrightsquigarrow v\}}}$$

We give a lower bound for the average graph distance under the null hypothesis, where the observed graph is a realization of a random geometric graph.

To do this, we first give a lower bound to the average Euclidean distance between two random vectors uniformly distributed on the torus, $X_1, X_2 \in [0,1]^d$, that is

$$\begin{aligned}
\mathbb{E}_0[D_T(X_1,X_2)] &= \int_{[0,1]^d} \sqrt{\sum_{j=1}^{d} \min\left(\left|x_j - \frac{1}{2}\right|, 1 - \left|x_j - \frac{1}{2}\right|\right)^2}\ dx_1 \dots dx_d \\
&= \int_{[0,1]^d} \sqrt{\sum_{j=1}^{d} \left|x_j - \frac{1}{2}\right|^2}\ dx_1 \dots dx_d \\
&\geq \int_{[0,1]^d} \max_{1\leq j\leq d} \left|x_j - \frac{1}{2}\right|\ dx_1 \dots dx_d \\
&= \mathbb{E}_0\left[\max_{1\leq j\leq d}\left|x_j - \frac{1}{2}\right|\right]
\end{aligned}$$

by symmetry this is simply the expectation of the maximum of d independent uniform random variables $Y_1, \dots, Y_d$ on $[0, \frac{1}{2}]$ and if we denote $M = \max_{1\leq j\leq d}(Y_j)$ than we have

$$\begin{aligned}
\mathbb{E}_0[D_T(X_1,X_2)] &\geq \mathbb{E}(M) \\
&\geq \int_0^{\frac{1}{2}} y\ f_M(y)dy \\
&= \int_0^{\frac{1}{2}} y\ (2^d d\ y^{d-1})dy = \frac{d}{2(d+1)}.
\end{aligned} \tag{2.8}$$

Under the null hypothesis, due to the geometric structure of the graph, a vertex can be represented as a random vector X_u uniformly distributed on the embedding space $[0,1]^d$ and two vertices u, v are connected only when the distance of their corresponding vectors X_v, X_u is less than the connection radius r.

Since we assumed that the graph is connected with high probability, we have

$$D_G(u,v) \geq \frac{D_T(X_v, X_u)}{r}.$$

This means that vertices that are separated by a large Euclidean distance are also separated by a large graph distance, under H_0.

We can consider the following lower bound on the average graph distance, which holds with high probability

$$D_G^{avg}(G) \geq \frac{1}{\binom{n}{2}} \sum_{1\leq u<v\leq n} D_T(X_v, X_u)/r. \tag{2.9}$$

Observe that the right-hand side of this can be seen as a U-statistic. In general this is defined as

$$U_n = \frac{1}{\binom{n}{b}} \sum_{(i_1,\dots,i_b)\in I_{n,b}} h(X_{i_1}, \dots, X_{i_b}),$$

where $X_{i_1}, \dots, X_{i_b}$ are random variables i.i.d. in a general space, $I_{n,b}$ is the set of combinations of b-tuples consisting of b distinct elements from $\{1, \dots, n\}$, h is called the kernel of the U-statistic and is assumed to be symmetric.

If $b = 1$, then U_n is an average of i.i.d. random variables. In our case $b = 2$ and $h(X_u, X_v) = D_T(X_u, X_v)$.
Using Chebyshev's inequality, for every $\epsilon > 0$, we obtain

$$P_0\left(\Big|\sum_{1\leq u<v\leq n} D_T(X_v, X_u)\binom{n}{2}^{-1} - \mathbb{E}_0[D_T(X_1, X_2)]\Big| > \epsilon\right)$$

$$\leq \frac{Var_0(\sum_{1\leq u<v\leq n} D_T(X_v, X_u)\binom{n}{2}^{-1})}{\epsilon^2}$$

$$\leq \frac{1}{\epsilon^2}\frac{2}{n}Var_0(D_T(X_1, X_2)) \to 0,$$

where for the second inequality we used a result on the variance of the U-statistic from [7, Theorem. 5.2].
Hence, Chebyshev's inequality ensures that $\sum_{1\leq u<v\leq n} D_T(X_v, X_u)\binom{n}{2}^{-1}$ is concentrated around $\mathbb{E}_0[D_T(X_1, X_2)]$ with probability tending to one.
Therefore using this result in (2.9) and (2.8), we obtain that the following lower bound holds, with high probability, for any $\epsilon > 0$

$$D_G^{avg}(G) \geq (1-\epsilon)\frac{\mathbb{E}_0[D_T(X_1, X_2)]}{r} \geq (1-\epsilon)\frac{d}{2(d+1)r}.$$

We will later show that, under the alternative hypothesis, the average graph distance is significantly smaller, since botnet vertices can create shortcuts between distant vertices making many paths much shorter. Hence, for this test we can compute $D_G^{avg}(G)$ for the given graph and detect the presence of a botnet when it is smaller than the considered threshold.

Definition 2.7. Fix $\epsilon > 0$. The average distance test rejects the null hypothesis for a given graph G, $\psi(G) = 1$, when

$$D_G^{avg}(G) < (1-\epsilon)\frac{d}{2(d+1)r} \tag{2.10}$$

This brings us to the main result of this section, where we give conditions for this test to be asymptotically powerful.

Theorem 2.3. *If $npk \to \infty$ and p is large enough to ensure that the subgraph induced by all non-botnet vertices is connected with high probability, then the average distance test from Definition 2.7 is asymptotically powerful.*

Proof. Under the null hypothesis, we have shown that the lower bound

$$D_G^{avg}(G) \geq (1-\epsilon)\frac{d}{2(d+1)r}$$

holds with high probability for any $\epsilon > 0$. Therefore, the average distance test has vanishing type-1 error, that is

$$\mathbb{P}_0(\psi(G) \neq 0) = \mathbb{P}_0\left(D_G^{avg}(G) < (1-\epsilon)\frac{d}{2(d+1)r}\right) \to 0.$$

In order to show that this test is asymptotically powerful we now have to show that the type-2 error also vanishes, that is $\mathbb{P}_B(\psi(G) \neq 1) \to 0$. To do this we are going to show

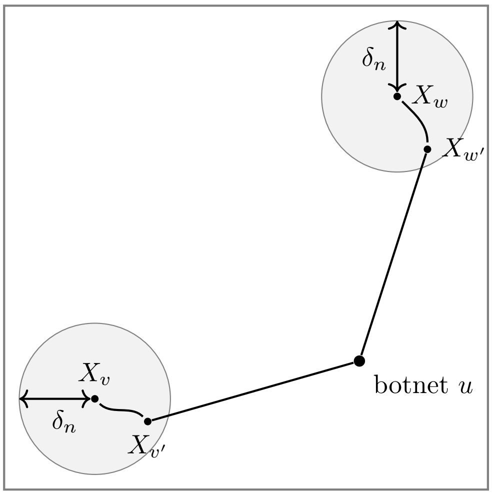


Figure 2.3: Example of botnet vertex $u \in B$ creating a shortcut between non-botnet vertices $v, w \in V \backslash B$.

that, under the alternative hypothesis, there is a botnet vertex that creates a shortcut between most pairs of non-botnet vertices, making their distance smaller than $o(1)/r$ with high probability. We will then show that the average graph distance is smaller than the threshold in Definition 2.7, with probability tending to one.

It is known from [11] that the assumption of connectedness in a random geometric graph implies $np \geq \Omega(\log(n))$. Therefore, this implies that $np \to \infty$ and thus the hypothesis $npk \to \infty$ is always satisfied for any $k \geq 1$.

Define the event $C = \{G_{V\backslash B} \text{ is connected}\}$, where $G_{V\backslash B}$ is the subgraph induced by $V\backslash B$. We have $P_B(C) \to 1$ by assumption.

For a vertex $v \in V\backslash B$, let $B_{\delta_n}(X_v)$ denote the ball of radius

$$\delta_n = (V_d \log(np))^{-1/d}$$

and with center in X_v a random vector uniformly distributed on T^d associated with v, as defined in our model, where $V_d = \frac{\pi^{d/2}}{\Gamma(d/2\ +1)}$ is the volume of a d-dimensional unit ball. Also, let

$$A_v = \{v' \in V\backslash B \mid X_{v'} \in B_{\delta_n}(X_v)\} \subseteq V\backslash B$$

denote the set of non-botnet vertices with their respective location inside $B_{\delta_n}(X_v)$.

We have that

$$\begin{aligned}
\mathbb{E}_B[|A_v|] &= \sum_{v' \in V\backslash B} \mathbb{P}_B(v' \in A_v) \\
&= \sum_{v' \in V\backslash B} \mathbb{P}_B(D_T(X_{v'}, X_v) < \delta_n) \\
&= (n-k)V_d \delta_n^d \\
&= \frac{n-k}{\log(np)} \\
&= (1+o(1))\frac{n}{\log(np)},
\end{aligned}$$

where we substituted $\delta_n = (V_d \log(np))^{-1/d}$ and we used the assumption that $k = o(n)$. Then, for any $\epsilon > 0$, we obtain

$$\begin{aligned}\mathbb{P}_B\left(|A_v| \geq (1-\epsilon)\frac{n}{\log(np)}\right) &= 1 - \mathbb{P}_B\left(|A_v| < (1-\epsilon)\frac{n}{\log(np)}\right)\\ &\geq 1 - \mathbb{P}_B\left(|A_v| < (1-\epsilon/2)\mathbb{E}[|A_v|]\right)\\ &\geq 1 - e^{-\frac{\epsilon^2}{8}\frac{n}{log(np)}} \to 1,\end{aligned}$$

where in the last inequality we used the relative Chernoff bound [10, Theorem. 4.5]. Now, we consider the probability that there exists a vertex $v' \in A_v$ such that it connects to a botnet vertex $u \in B$. This is

$$\begin{aligned}&\mathbb{P}_B(\exists v' \in A_v : v \leftrightarrow u)\\ &\geq \mathbb{P}_B\left(\exists v' \in A_v : v \leftrightarrow u \,\middle|\, |A_v| \geq (1-\epsilon)\frac{n}{\log(np)}\right)\mathbb{P}_B\left(|A_v| \geq (1-\epsilon)\frac{n}{\log(np)}\right)\\ &\geq (1+o(1))\left(1-(1-p)^{(1-\epsilon)\frac{n}{\log(np)}}\right) \qquad (2.11)\\ &\geq (1+o(1))\left(1-e^{(1-\epsilon)\frac{n}{\log(np)}\log(1-p)}\right)\\ &\geq (1+o(1))\left(1-e^{-(1-\epsilon)\frac{np}{\log(np)}}\right) \to 1,\end{aligned}$$

where we used $np/\log(np) \to \infty$.
To continue, we need to estimate the graph distance with the torus distance. The following theorem holds (a detailed proof of this can be found in [3, Theorem. 8]):

Theorem. *There exists a constant* K*, independent of* n*, such that for any pair of vertices in the same connected component* $v, v' \in V$ *with* $D_T(X_v, X_{v'}) \gg \frac{log(n)}{n\ r^{d-1}}$ *we have*

$$D_G(v, v') \leq K\frac{D_T(X_v, X_{v'})}{r} \qquad (2.12)$$

with high probability.

Therefore, given the event $C = \{G_{V\setminus B} \text{ is connected}\}$, the theorem above guarantees that the shortest path between v and every $v' \in A_v$ is of length $D_G(v, v') \leq O(\delta_n)/r$, with high probability.
Now, for a given $v \in V\setminus B$ and an arbitrary $u \in B$, we have that

$$\begin{aligned}&\mathbb{P}_B(D_G(v,u) \leq 1 + O(\delta_n)/r)\\ &\geq \mathbb{P}_B(C \cap \{D_G(v,u) \leq 1 + O(\delta_n)/r\})\\ &\geq \mathbb{P}_B(\exists v' \in A_v : v' \leftrightarrow u, D_G(v,v') \leq O(\delta_n)/r \mid C)\mathbb{P}_B(C) \to 1\end{aligned}$$

where we used (2.11), (2.12) and the assumption that $\mathbb{P}_B(C) \to 1$.

Similarly to the situation described in Figure 2.3, we can apply the previous result twice for an arbitrary pair of vertices $v, w \in V\backslash B$ and $u \in B$, we obtain

$$\mathbb{P}_B(D_G(v,w) \leq 2 + 2\ O(\delta_n)/r)$$

$$\geq \mathbb{P}_B(D_G(v,u) \leq 1 + O(\delta_n)/r,\ D_G(w,u) \leq 1 + O(\delta_n)/r)) \to 1.$$

We observe that

$$\frac{2 + 2\ O(\delta_n)/r}{1/r} = 2r + 2\ O(\delta_n) \to 0$$

where the convergence to 0 follows from $\delta_n = (V_d \log(np))^{-1/d} \to 0$ under our assumptions for this theorem and $r \to 0$, because of the relation (2.3) and the assumption that $p \to 0$. Therefore, we have that

$$2 + 2\ O(\delta_n)/r = o(1/r)$$

We can use this in our previous result to conclude that

$$\mathbb{P}_B(D_G(v,w) \leq o(1)/r) \geq \mathbb{P}_B(D_G(v,w) \leq 2 + 2\ O(\delta_n)/r) \to 1.$$

This can be strengthened to also include the botnet vertices, in fact observe that every botnet vertex connects to several non-botnet vertices with high probability (as we showed in the second part of the proof of Theorem 2.2). Therefore, for an arbitrary pair $v, w \in V$ we have

$$\mathbb{P}_B(D_G(v,w) \leq o(1)/r) \to 1. \tag{2.13}$$

This is the main result of the proof.
We consider the diameter of G, that is,

$$diam(G_V) = \max_{v,w \in V} D_G(v,w)$$

and we can now show that it is at most $O(1)/r$ with high probability.
We first consider the diameter of $G_{V\backslash B}$ and we have

$$\mathbb{P}_B(diam(G_{V\backslash B}) \leq O(1)/r) \geq \mathbb{P}_B\left(C \cap \{diam(G_{V\backslash B}) \leq O(1)/r\}\right) \to 1,$$

where the convergence to 1 follows from using (2.12)
Similarly to what we did above we can extend this to the diameter of G, since we have shown that every botnet vertex connects to at least one non-botnet vertex with high probability. That is,

$$\mathbb{P}_B\left(diam(G_V) \leq O(1)/r\right) \to 1. \tag{2.14}$$

Finally, for any $a > 0$, from (2.14) we have

$$\mathbb{P}_B\left(D_G^{avg}(G) \geq \frac{a}{r}\right) = \mathbb{P}_B\left(\nVdash_{\{diam(G_V) \leq O(1)/r\}} D_G^{avg}(G) \geq \frac{a}{r}\right) - o(1)$$

$$\leq \frac{r}{a}\mathbb{E}_B\left(\nVdash_{\{diam(G_V) \leq O(1)/r\}} D_G^{avg}(G)\right) - o(1),$$

where we used Markov's inequality.
Now we can use the dominated convergence theorem, since $D_G^{avg}(G) \leq diam(G_V)$, and (2.13) in order to show that

$$\lim_{n\to\infty} r\,\mathbb{E}_B\left(\nVdash_{\{diam(G_V) \leq O(1)/r\}} D_G^{avg}(G)\right) = \mathbb{E}_B\left(\nVdash_{\{diam(G_V) \leq O(1)/r\}} \lim_{n\to\infty} r D_G^{avg}(G)\right) = 0.$$

Putting this all together we have that

$$\mathbb{P}_B \left(D_G^{avg}(G) \geq \frac{a}{r} \right) \to 0.$$

In particular, choosing $a = (1-\epsilon)\frac{d}{2(d+1)}$, we obtain

$$\mathbb{P}_B \left(D_G^{avg}(G) < (1-\epsilon)\frac{d}{2(d+1)r} \right) = 1 - \mathbb{P}_B \left(D_G^{avg}(G) \geq (1-\epsilon)\frac{d}{2(d+1)r} \right) \to 1.$$

This proves that the average distance test is asymptotically powerful. □

2.4 Estimation of model parameters

In the problem considered, we are given a single observation of a graph and we want to use the two proposed tests to determine whether we can accept the null hypothesis or we have to reject it. However, computing the threshold for the isolated star test requires knowledge of the dimension d of the embedding space, while the threshold for the average distance test requires having the dimension d and the connection radius r.

We recall that the embedding space of the random geometric graph, as defined in Section 1, is not available to us, therefore we do not have direct knowledge of these two model parameters.

In this section, we show how to estimate the dimension d and the connection radius r of the given graph under both the null and alternative hypothesis.

Definition 2.8. Let the vertices $v, w \in V$ be both directly connected to a vertex $u \in V$. We define the clustering coefficient C_d as the probability that v and w are also directly connected. That is

$$C_d = \mathbb{P}_0(v \leftrightarrow w \mid u \leftrightarrow v, u \leftrightarrow w).$$

We use the clustering coefficient to estimate the dimension d.

Under the null hypothesis, the clustering coefficient can be computed analytically using [6, see (15)]. We obtain

$$C_d = \mathbb{P}\left(Beta\left(\frac{d+1}{2}, \frac{1}{2} \right) \leq \frac{3}{4} \right) + \mathbb{P}\left(Beta\left(\frac{d+1}{2}, \frac{d+1}{2} \right) \leq \frac{1}{4} \right), \tag{2.15}$$

where $Beta(\cdot,\cdot)$ denotes a random variable with a beta distribution.

This shows that it is purely a geometric quantity depending only on the dimension d.

An intuitive way to estimate the clustering coefficient for a given graph is to use the following estimator

$$\hat{C_d} = \frac{\sum_{1 \leq u,v,w \leq n} \nVdash_{\{v \leftrightarrow w, u \leftrightarrow v, u \leftrightarrow w\}}}{\sum_{1 \leq u,v,w \leq n} \nVdash_{\{u \leftrightarrow v, u \leftrightarrow w\}}}, \tag{2.16}$$

for distinct $u, v, w \in V$.

Therefore, to estimate d we can first estimate the clustering coefficient with $\hat{C_d}$ using (2.16) and then invert (2.15) to obtain an estimate $\hat{d}$ of the dimension.

We will now show an important result on this method of estimating the dimension.

Lemma 1. *Using the clustering coefficient to estimate the dimension with $\hat{d}$ is consistent under both the null hypothesis and the alternative hypothesis, that is $\hat{d} \xrightarrow{\mathbb{P}_0} d$ and $\hat{d} \xrightarrow{\mathbb{P}_B} d$.*

Proof. We begin by showing that $\hat{C}_d \xrightarrow{\mathbb{P}_o} C_d$, therefore it follows that $\hat{d} \xrightarrow{\mathbb{P}_0} d$ by the continuous mapping theorem since the relation in (2.15) is continuous.
From (2.16) we have that

$$\hat{C}_d(G) = \frac{n^{-3}\sum_{1\le u,v,w\le n} \nVdash_{\{v\leftrightarrow w, u\leftrightarrow v, u\leftrightarrow w\}}/p^2}{n^{-3}\sum_{1\le u,v,w\le n} \nVdash_{\{u\leftrightarrow v, u\leftrightarrow w\}}/p^2}, \tag{2.17}$$

we will then show that the numerator converges in probability to C_d and the denominator converges in probability to 1.
Let

$$X = n^{-3} \sum_{1\le u,v,w\le n} \nVdash_{\{v\leftrightarrow w, u\leftrightarrow v, u\leftrightarrow w\}}/p^2,$$

then X is exactly the numerator in (2.17) and it has

$$\begin{aligned}
\mathbb{E}_0[X] &= n^{-3} \sum_{1\le u,v,w\le n} \mathbb{E}_0[\nVdash_{\{v\leftrightarrow w, u\leftrightarrow v, u\leftrightarrow w\}}]/p^2 \\
&= n^{-3} \sum_{1\le u,v,w\le n} \mathbb{P}_o(v\leftrightarrow w, u\leftrightarrow v, u\leftrightarrow w)/p^2 \\
&= n^{-3} \sum_{1\le u,v,w\le n} \frac{\mathbb{P}_o(u\leftrightarrow v, u\leftrightarrow w)}{p^2}\mathbb{P}_o(v\leftrightarrow w\ | u\leftrightarrow v, u\leftrightarrow w) \\
&= (1+o(1))C_d.
\end{aligned}$$

Moreover, we have that

$$\begin{aligned}
\mathbb{E}_0[X^2] =& n^{-6} \sum_{1\le u,v,w,u',v',w'\le n} \mathbb{E}_0[\nVdash_{\{v\leftrightarrow w, u\leftrightarrow v, u\leftrightarrow w\}}\nVdash_{\{v'\leftrightarrow w', u'\leftrightarrow v', u'\leftrightarrow w'\}}]/p^4 \\
=& n^{-6} \sum_{\substack{1\le u,v,w,u',v',w'\le n \\ \{u,v,w\}\cap\{u',v',w'\}=\emptyset}} \mathbb{E}_0[\nVdash_{\{v\leftrightarrow w, u\leftrightarrow v, u\leftrightarrow w\}}\nVdash_{\{v'\leftrightarrow w', u'\leftrightarrow v', u'\leftrightarrow w'\}}]/p^4 \\
&+ 3n^{-6} \sum_{\substack{1\le u,v,w,v',w'\le n \\ \{u,v,w\}\cap\{v',w'\}=\emptyset}} \mathbb{E}_0[\nVdash_{\{v\leftrightarrow w, u\leftrightarrow v, u\leftrightarrow w\}}\nVdash_{\{v'\leftrightarrow w', u\leftrightarrow v', u\leftrightarrow w'\}}]/p^4 \\
&+ 3n^{-6} \sum_{\substack{1\le u,v,w,w'\le n \\ \{u,v,w\}\cap\{w'\}=\emptyset}} \mathbb{E}_0[\nVdash_{\{v\leftrightarrow w, u\leftrightarrow v, u\leftrightarrow w\}}\nVdash_{\{v\leftrightarrow w', u\leftrightarrow v, u\leftrightarrow w'\}}]/p^4 \\
&+ n^{-6} \sum_{1\le u,v,w\le n} \mathbb{E}_0[\nVdash^2_{\{v\leftrightarrow w, u\leftrightarrow v, u\leftrightarrow w\}}]/p^4 \\
=& (1+o(1))\left[C_d^2 + 3n^{-1}C_d^2 + 3n^{-2}\frac{C_d^2}{p} + n^{-3}\frac{C_d^2}{p^2}\right] \\
=& (1+o(1))C_d^2,
\end{aligned}$$

where the final step follows from the assumption made in our model that $1 = O(np)$.

Therefore, we have

$$Var_0(X) = \mathbb{E}_0[X^2] - \mathbb{E}_0[X]^2 = o(1).$$

We can use this result in Chebyshev's inequality, given an $\epsilon > 0$, it follows that

$$\mathbb{P}_0\Big(|X - C_d| > \epsilon\Big) = \mathbb{P}_0\Big(|X - \mathbb{E}_0[X]| > \epsilon\Big) < \frac{Var_0(X)}{\epsilon^2} \to 0,$$

by definition this is $X \xrightarrow{\mathbb{P}_0} C_d$.
We have shown that the numerator in (2.17) converges in probability to C_d. We also have that the denominator converges in probability to 1, with largely similar computations. Hence, we have $\hat{C}_d \xrightarrow{\mathbb{P}_0} C_d$.
Finally, it follows from the continuous mapping theorem that $\hat{d} \xrightarrow{\mathbb{P}_0} d$ and we can conclude that this estimator is consistent under the null hypothesis.
Under the alternative hypothesis, the proof is largely similar. Since we assumed that $k = o(n)$ the botnet size is small, it can be seen that the first and second moments of X converge to the same values. Therefore, using the same argument as above, we have $X \xrightarrow{\mathbb{P}_B} C_d$. Hence, we have $\hat{C}_d \xrightarrow{\mathbb{P}_B} C_d$. Finally, by the continuous mapping theorem we can conclude that the estimator $\hat{d}$ is also consistent under the alternative hypothesis. $\square$

To estimate the connection radius r we first estimate the edge probability $p = \mathbb{P}_0(u \leftrightarrow v)$, where $u, v \in V$ are two generic vertices.
We recall that in Section 2.2 we derived the explicit relation (2.3), which is

$$r = \frac{1}{\sqrt{\pi}} \left[p \; \Gamma(d/2 + 1)\right]^{\frac{1}{d}},$$

by exploiting the geometrical properties of the random geometric graph.
Therefore, we can use the estimate of p and the one of d given previously to obtain an estimate of r using this relation.
An intuitive way to estimate the edge probability for a given graph is to use the following estimator, with $u, v \in V$

$$\hat{p} = \binom{n}{2}^{-1} \sum_{1 \le u < v \le n} \mathbb{1}_{\{u \leftrightarrow v\}} \tag{2.18}$$

We will now show that this method gives a consistent estimator of p.

Lemma 2. *Using $\hat{p}$, defined above, to estimate p is consistent under both the null and alternative hypothesis, that is $\frac{\hat{p}}{p} \xrightarrow{\mathbb{P}_0} 1$ and $\frac{\hat{p}}{p} \xrightarrow{\mathbb{P}_B} 1$.*

Proof. We begin by showing that $\frac{\hat{p}}{p} \xrightarrow{\mathbb{P}_0} 1$.
Using the estimator $\hat{p}$ as given in (2.18) we have

$$\begin{aligned}
\mathbb{E}_0[\hat{p}/p] &= \binom{n}{2}^{-1} \sum_{1 \le u < v \le n} \mathbb{E}_0\left[\mathbb{1}_{\{u \leftrightarrow v\}}\right]/p \\
&= \binom{n}{2}^{-1} \sum_{1 \le u < v \le n} \mathbb{P}_0\left(u \leftrightarrow v\right)/p \\
&= \binom{n}{2}^{-1} \sum_{1 \le u < v \le n} 1 \; = \binom{n}{2}^{-1} \binom{n}{2} = 1.
\end{aligned}$$

With similar computations, we can also show that the estimator $\hat{p}$ is unbiased.
Moreover, we have

$$\begin{aligned}
\mathbb{E}_0[(\hat{p}/p)^2] =&\binom{n}{2}^{-2} \sum_{\substack{1\leq u<v\leq n\\ 1\leq u'<v'\leq n}} \mathbb{E}_0\left[\mathbb{1}_{\{u\leftrightarrow v\}}\mathbb{1}_{\{u'\leftrightarrow v'\}}\right]/p^2\\
=&\binom{n}{2}^{-2} \sum_{\substack{1\leq u<v\leq n\\ 1\leq u'<v'\leq n\\ \{u,v\}\cap\{u',v'\}=\emptyset}} \mathbb{E}_0\left[\mathbb{1}_{\{u\leftrightarrow v\}}\right]\mathbb{E}_0\left[\mathbb{1}_{\{u'\leftrightarrow v'\}}\right]/p^2\\
&+\binom{n}{2}^{-2} \sum_{1\leq u<v\leq n} \mathbb{E}_0\left[\mathbb{1}^2_{\{u\leftrightarrow v\}}\right]/p^2\\
=&\binom{n}{2}^{-2}\left[\binom{n}{2}^2-\binom{n}{2}\right]+\binom{n}{2}^{-2}\left[\binom{n}{2}\frac{1}{p}\right]\\
=&1-\binom{n}{2}^{-1}+\binom{n}{2}^{-1}\frac{1}{p}=1+o(1)
\end{aligned}$$

where this last equality follows from the assumption $1 = O(np)$.
Therefore, we have

$$Var_0(\hat{p}/p) = \mathbb{E}_0[(\hat{p}/p)^2] - \mathbb{E}_0[\hat{p}/p]^2 = o(1).$$

We can use this result in Chebyshev's inequality, given an $\epsilon > 0$,

$$\mathbb{P}_0\Big(|\hat{p}/p-1|>\epsilon\Big)=\mathbb{P}_0\Big(|\hat{p}/p-\mathbb{E}_0[\hat{p}/p]|>\epsilon\Big)<\frac{Var_0(\hat{p}/p)}{\epsilon^2}\to 0,$$

and by definition this is $\frac{\hat{p}}{p}\xrightarrow{\mathbb{P}_0}1$.
Under the measure $\mathbb{P}_B$ in our model, for distinct $u, v \in V$, we have

$$\mathbb{P}_B(u\leftrightarrow v) = p = \mathbb{P}_0(u\leftrightarrow v).$$

Hence, performing the same computations as above, we can conclude that $\frac{\hat{p}}{p}\xrightarrow{\mathbb{P}_B}1$. □

Finally, to estimate the connection radius r we can estimate the edge probability using $\hat{p}$ from (2.18) and then use our estimate of d, given previously, to obtain the estimate $\hat{r}$ from the relation in (2.3).
Since the radius r is given as a continuous function of p in (2.3) and we have shown that $\hat{p}$ is consistent, by the continuous mapping theorem we have that our estimate $\hat{r}$ is also consistent under the null and alternative hypothesis. That is $\frac{\hat{r}}{r}\xrightarrow{\mathbb{P}_0}1$ and $\frac{\hat{r}}{r}\xrightarrow{\mathbb{P}_B}1$.

Chapter 3

Numerical Simulations

In the previous chapter, we have shown that the two proposed tests are asymptotically powerful when $npk \to \infty$, which means that the sum of worst-case errors approaches zero as the number of vertices $n \to \infty$. However, asymptotic results can be poor approximations to the actual finite setting. Furthermore, convergence to a limit as $n \to \infty$ does not guarantee that the approximation improves with increasing n. Therefore, we accompany the theoretical results with a simulation study to compare the performance of the two tests in the finite case.

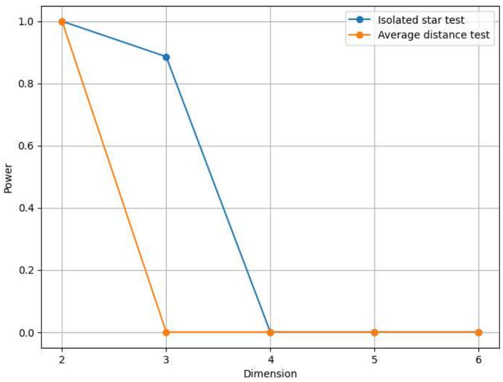
First, we generate a random geometric graph $\mathbb{G}(n, d, p)$, as described in Section 2.2, with the chosen parameters. Then, for graphs under the alternative hypothesis, we add the k botnet vertices. From this graph, we estimate the necessary model parameters with the consistent estimators described in Section 2.4 and use them to compute the rejection thresholds for the two tests. For the isolated star test, the rejection threshold is the kissing number k_d, defined in Section 2.3.1, this is explicitly known only in very few dimensions, however, there exists good upper bounds. These are numerically difficult to calculate and having them as precise as possible is of great interest for our results. Therefore, for dimensions $d \leq 24$, we use the best known upper bounds from [8]. Finally, we evaluate the two test statistics, defined in Section 2.3.1 and Section 2.3.2, and decide whether or not we can reject the null hypothesis. This process is repeated for every sample in the simulation.

We have found that both the isolated star test and the average distance test have zero type-1 error, which means that in our simulations under the null hypothesis they correctly identified every graph as without a botnet. This was expected, given their construction and the first part of the proofs of Theorem 2.2 and Theorem 2.3, respectively. Therefore, our primary interest is the power of the tests, which measures the detection rate when a botnet is present.

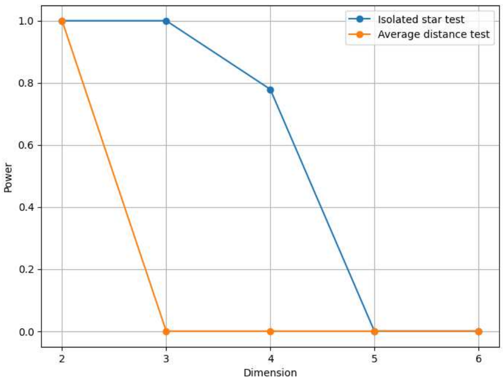
The results of the simulation study are reported in Figure 3.1. These showcase how performance varies according to the probability of connection p and the embedding dimension d, while the number of vertices n and the botnet size k remain fixed. We use the average degree np instead of p to characterize the graph.

We observe that both tests perform quite well, even on relatively small graphs, provided that the dimension d is small. However, the isolated star test outperforms the average distance test, especially when the average degree np is higher.

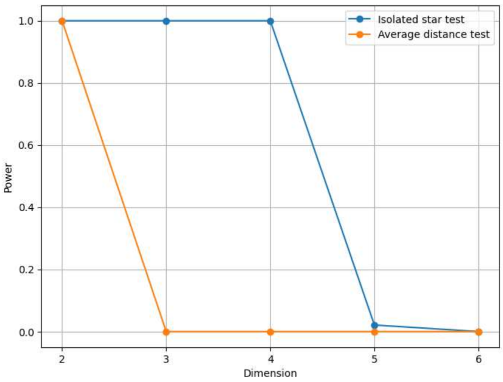
The isolated star test rejects the null hypothesis when the graph contains an isolated star larger than the kissing number k_d. The reason for the observed performance of this test might be that, as shown in Theorem 2.2, a botnet vertex will form an isolated star at least as large as its non-botnet degree with high probability. Therefore, when the average degree np is high, the graph is more likely to contain a large isolated star. Furthermore, the rejection threshold k_d is lower when the dimension d is small, but quickly becomes too



(a) Average degree $np = 10$



(b) Average degree $np = 20$



(c) Average degree $np = 30$

Figure 3.1: The power of the isolated star test and the average distance test as a function of the dimension d on different average degrees np. The parameters used to generate the graphs are: number of vertices $n = 10000$ and botnet size $k = 10$. We generated 1000 samples for each dimension in each simulation.

high for our relatively small graph as the dimension increases. For example, numerical simulations suggest that, for dimension $d = 4$ and average degree $np = 10$, the isolated star under the alternative hypothesis is typically smaller than 16, however the kissing number is $k_4 = 24$. Hence, this test performs best when the dimension d is small and the average degree np is large.

The performance of the average distance test is also related to the dimension d and the average degree np. Recall that in Theorem 2.3 we have exploited the fact that a botnet vertex can create shortcuts between distant vertices in the embedding space, decreasing the average graph distance under the alternative hypothesis. However, as the dimension increases the connection radius in (2.3) also increases, which causes the average graph distance among non-botnet vertices to decrease. For this reason, the shortcuts created by botnet vertices have a less pronounced effect in higher dimensions, making them less detectable. Furthermore, we observed that in our settings the rejection threshold used for this test, from Definition 2.7, becomes far too small as the dimension increases, contributing to the poor performance observed. Conversely, when np is large there is a higher probability that botnet vertices create shortcuts between distant vertices in the embedding space, which should make detection easier. However, in our simulations we did not see this effect playing a major role in the performance of this test. Hence, with this test we see the best performance when the dimension d is small.

Recall that in Theorem 2.3 we made the technical assumption that the subgraph of non-botnet vertices is connected with high probability. However, in simulations, this test performed the same when considering an average degree $np = 10$, where we observed many graphs that had a large connected component but were not fully connected, as when considering a higher average degree, which ensures that the random geometric graph is connected with high probability. Therefore, this leads us to the conjecture that Theorem 2.3 also holds under the milder condition that a giant connected component exists.

We also investigated the influence of the botnet size on the tests' performance. To do this, we repeated the first simulation with a much larger botnet size, $k = 100$. The results are reported in Figure 3.2.

We observed that the power of the isolated star test was slightly higher, presumably because a larger botnet increases the probability that at least one botnet vertex will form a large isolated star. However, if we were to consider an even larger botnet size in our relatively small graph, this could start to introduce some errors in the estimators of the model parameters and would not lead to good performance.

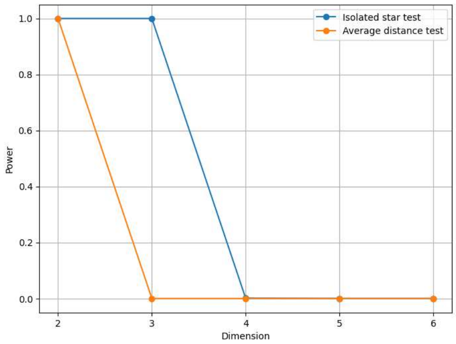


Figure 3.2: The power of the isolated star test and the average distance test as a function of the dimension d. The parameters used to generate the graphs are: number of vertices $n = 10000$, average degree $np = 10$ and botnet size $k = 100$. We generated 1000 samples.

Finally, we found that the estimator for the dimension d was accurate in all the cases considered. We only saw some errors starting from dimension $d = 12$, with $n = 10000$ vertices. An extremely large botnet size k can also cause errors in this estimator. Furthermore, the errors between the estimated connection radius and the true value of r were negligible (on the order of 10^{-4} when d, used in this estimator, was estimated correctly) and did not impact our results. Therefore, using the estimated model parameters instead of the true values in our simulation did not introduce any errors and yielded the same performance.

Chapter 4

Conclusions

In this thesis, the problem of detecting a botnet in a network is formalized as a hypothesis testing problem, using only the structural information of the network represented as a random graph, in order to propose and analyze two statistical tests: the isolated star test and the average distance test.

Exploiting the underlying geometry of our model, based on the random geometric graph, we have shown in Theorems 2.2 and 2.3 that both tests are asymptotically powerful under appropriate assumptions on the asymptotic regime of the model parameters. This means that the sum of the worst-case probability of type-1 and type-2 errors goes to zero as the graph size goes to infinity. In particular, we considered two test statistics: the size of the largest isolated star and the average graph distance. We proposed a rejection threshold for each and showed that these thresholds cause both error probabilities to converge to zero. Furthermore, in Theorem 2.1 we have shown that these results are optimal, as no test can be asymptotically powerful if the assumptions on the model parameters are not met.

With the numerical simulations, we examined the performance of the tests in the finite sample case to complement our theoretical results for the asymptotic regime. We used the consistent estimators, given in Section 2.4, for the model parameters required by our tests. We observed that both tests performed well, even with a relatively small number of edges, when the dimension of the embedding space was small and the average degree was high. In fact, both had zero type-1 error with the parameters we considered. The isolated star test had higher power than the average distance test, particularly when considering a higher average degree. However, the power of both tests deteriorates as the dimension increases.

Appendix A

Python Codes

A.1 Simulations

We give all the functions used in the simulation.

```python
import numpy as np
import networkx as nx
import matplotlib.pyplot as plt
import random
import scipy.special as sp
import scipy.spatial as sc
import scipy.stats as stats


def get_r(p,d):
    """
    Compute the connection radius based on the probability p and embedding dimension d,
    by using equation (2.3).

    Args:
        - p: the probability of connection.
        - d: the embedding dimension.

    Returns:
        - the connection radius.
    """

    r = (p * sp.gamma(d/2 + 1))**(1/d) / np.pi**(1/2)

    return r


def get_graph(n,d,r):
    """
    Generates n random points in d-dimensional unit cube using numpy.
    Creates a graph of size n, then connects two vertices if their corrisponding points
    have Euclidean distance on the torus less than or equal to the threshold r.

    Args:
        - n: the number of vertices.
        - d: the dimension.
        - r: the connection radius.

    Returns:
        - a networkx graph object.
        - the list of points in the embedding space used to generate the graph.
    """

    points = np.random.rand(n, d)
    graph = nx.Graph()
    for i in range(n):
        graph.add_node(i)

    #Add edges using an efficient KD-Tree. The `boxsize` parameter enables
    #distance calculations with periodic boundary conditions.
    kdtree = sc.cKDTree(points, boxsize=[1.0] * d)

    edge_pairs = kdtree.query_pairs(r, output_type='ndarray')

    graph.add_edges_from(edge_pairs)

    return graph, points


def rewiring(graph, k, p):
    """
    Selects k random vertices, removes their incident edges, and connects
    them to every other vertex with probability p.

    Args:
        - graph: a networkx graph.
        - k: the number of vertices in the botnet.
        - p: the probability of connecting a selected vertex to another vertex.

    Returns:
        - a modified networkx graph and the list of selected vertices.
    """
    bot = []
    if k > 0:
      bot = random.sample(list(graph.nodes()), k)

    for node in bot:
      for neighbor in list(graph.neighbors(node)):
        graph.remove_edge(node, neighbor)

    for i in bot:
      for j in graph.nodes():
        if i != j:
          if random.random() < p:
            graph.add_edge(i, j)

    return graph, bot


    def build_look_up(d_min=1,d_max=100):
```

```python
    """
    Builds a lookup table of values of the clustering coefficient for integer dimensions.

    Args:
        - min_d: the minimum integer dimension for the lookup table.
        - max_d: the maximum integer dimension for the lookup table.

    Returns:
        - an np array of integer dimensions and the lookup table.
    """

    def calc_Cd(d):
        """
        Calculates the theoretical clustering coefficient C_d for a given dimension d.
        This function implements Equation (2.15):
        C_d = P(Beta((d+1)/2, 1/2) <= 3/4) + P(Beta((d+1)/2, (d+1)/2) <= 1/4)

        Args:
            - d: the dimension.

        Returns:
            - the theoretical clustering coefficient C_d.
        """

        # Calculate P(Beta((d+1)/2, 1/2) <= 3/4)
        term1 = stats.beta.cdf(0.75, a=(d+1)/2, b=1/2)

        # Calculate P(Beta((d+1)/2, (d+1)/2) <= 1/4)
        term2 = stats.beta.cdf(0.25, a=(d+1)/2, b=(d+1)/2)

        return term1 + term2

    d_values = np.arange(d_min, d_max + 1, dtype=int)

    Cd_values = np.array([calc_Cd(d) for d in d_values])

    return d_values, Cd_values


def estimate_d(graph):
    """
    Estimates the global clustering coefficient (transitivity) of a graph
    as defined in equation (2.16).
    Estimates the integer dimension 'd' by finding the value in the lookup table
    that is closest to the estimated clustering coefficient.

    Args:
        - graph: a networkx graph object.

    Returns:
        - the estimated dimensiont.
    """

    # networkx.transitivity(G) directly computes the ratio of triangles
    # to connected triples, which corresponds to the formula in equation (2.16).
    Cd = nx.transitivity(graph)

    min_index = np.argmin(np.abs(Cd_values - Cd))
    d_hat = d_values[min_index]

    return d_hat


def estimate_r(graph, d, n):
    """
    Estimates the edge probability p of a graph as described in equation (2.18)
    with this value we can estimate the connection radius r
    by using equation (2.3).

    Args:
        - graph: a networkx graph object.
        - d: the dimension of the embedding space.
        - n: the number of vertices.

    Returns:
        - the estimated connection radius.
    """

    n_edges = graph.number_of_edges()

    p_hat = n_edges / (n * (n - 1) / 2)

    r_hat = (p_hat * sp.gamma(d/2 + 1))**(1/d) / np.pi**(1/2)

    return r_hat


def Kd_upper_bound(d):
    """
    Returns the upper bound for the kissing number in dimension d.

    Args:
        - d: the dimension.

    Returns:
        - the upper bound as an integer or float
    """
    upper_bounds_table = {
        1: 2,
        2: 6,
        3: 12,
        4: 24,
        5: 44,
        6: 77,
        7: 134,
        8: 240,
        9: 363,
        10: 553,
        11: 868,
        12: 1355,
        13: 2064,
        14: 3174,
        15: 4853,
        16: 7320,
        17: 10978,
        18: 16406,
        19: 24417,
        20: 36195,
        21: 53524,
        22: 80810,
        23: 122351,
        24: 196560
    }

    if d in upper_bounds_table:
        return upper_bounds_table[d]


def calc_max_Isolated_Star(graph):
    """
    Calculates the cardinality of the isolated star for a every vertex in the graph.
    (size of the largest independent set of the subgraph
    induced by the neighbors of the vertex)
    Returns the maximum cardinality.

    Args:
        - graph: a networkx graph object.

    Returns:
        - the cardinality of the biggest isolated star in the graph G.
    """
    Isolated_Star_card = []
    for vertex in graph.nodes():
        neighbors = list(graph.neighbors(vertex))
        if not neighbors:
            Isolated_Star_card.append(0)
        subgraph = graph.subgraph(neighbors)

        # NetworkX's max_independent_set uses a greedy algorithm,
        # which is not guaranteed to find the true maximum independent set
        # but is a reasonable approximation and is computationally feasible.
        Isolated_Star = nx.approximation.maximum_independent_set(subgraph)

        Isolated_Star_card.append(len(Isolated_Star))

    max_Isolated_Star = np.max(Isolated_Star_card)

    return max_Isolated_Star


def calc_D_avg(graph):
    """
    If the graph is connected, calculates the average shortest path length of the graph.
    If the graph is not connected, calculates the average shortest path length
    in every connected component and returns the maximum.

    Args:
        - graph: a networkx graph.

    Returns:
        - the average shortest path length.
    """

    D_avg = []
    for i, component_nodes in enumerate(nx.connected_components(graph)):
        component = graph.subgraph(component_nodes)

        avg = nx.average_shortest_path_length(component)

        if np.isnan(avg):
            D_avg.append(0)
        else:
            D_avg.append(avg)

    max_D_avg = np.max(D_avg)

    return max_D_avg
```

```
total_d = 5
total_iter = 1000

start_d = 2
start_iter = 1

n = 10000
p = 0.001
k = 10

numerr_d_hat = [0] * (total_d + 1 - start_d)
numerr_d_bot_hat = [0] * (total_d + 1 - start_d)
numReject_IsolatedStar = [0] * (total_d + 1 - start_d)
numReject_IsolatedStar_bot = [0] * (total_d + 1 - start_d)
numReject_D_avg = [0] * (total_d + 1 - start_d)
numReject_D_avg_bot = [0] * (total_d + 1 - start_d)
numReject_IsolatedStar_hat = [0] * (total_d + 1 - start_d)
numReject_IsolatedStar_bot_hat = [0] * (total_d + 1 - start_d)
numReject_D_avg_hat = [0] * (total_d + 1 - start_d)
numReject_D_avg_bot_hat = [0] * (total_d + 1 - start_d)

d_values, Cd_values = build_look_up()


for d in range(start_d, total_d+1):

    r=get_r(p,d)

    for iter in range(start_iter, total_iter+1):

        graph=get_graph(n,d,r)[0]
        graph_bot=graph.copy()
        graph_bot=rewiring(graph_bot, k, p)[0]


        d_hat=estimate_d(graph)
        if d_hat != d:
            numerr_d_hat[d - start_d] += 1

        r_hat=estimate_r(graph, d_hat, n)

        d_bot_hat=estimate_d(graph_bot)
        if d_bot_hat != d:
            numerr_d_bot_hat[d - start_d] += 1

        r_bot_hat=estimate_r(graph_bot, d_bot_hat, n)


        Kd=Kd_upper_bound(d)

        max_Isolated_Star = calc_max_Isolated_Star(graph)

        if max_Isolated_Star > Kd:
            numReject_IsolatedStar[d - start_d] += 1

        max_Isolated_Star_bot = calc_max_Isolated_Star(graph_bot)

        if max_Isolated_Star_bot > Kd:
            numReject_IsolatedStar_bot[d - start_d] += 1

        max_D_avg = calc_D_avg(graph)

        if max_D_avg < d/(2*(d+1)*r):
            numReject_D_avg[d - start_d] += 1

        max_D_avg_bot = calc_D_avg(graph_bot)

        if max_D_avg_bot < d/(2*(d+1)*r):
            numReject_D_avg_bot[d - start_d] += 1


        Kd_hat=Kd_upper_bound(d_hat)

        if max_Isolated_Star > Kd_hat:
            numReject_IsolatedStar_hat[d - start_d] += 1

        Kd_bot_hat=Kd_upper_bound(d_bot_hat)

        if max_Isolated_Star_bot > Kd_bot_hat:
            numReject_IsolatedStar_bot_hat[d - start_d] += 1

        if max_D_avg < d_hat/(2*(d_hat+1)*r_hat):
            numReject_D_avg_hat[d - start_d] += 1

        if max_D_avg_bot < d_bot_hat/(2*(d_bot_hat+1)*r_bot_hat):
            numReject_D_avg_bot_hat[d - start_d] += 1

    start_iter = 1
```

This function is used to generate Figure 3.1 and Figure 3.2.

```
def plot_power (numReject_IsolatedStar_bot, numReject_D_avg_bot, start_d, total_d, total_iter):
    power_isolated_star = [n / total_iter for n in numReject_IsolatedStar_bot]
    power_d_avg = [n / total_iter for n in numReject_D_avg_bot]

    x_labels = [str(d) for d in range(start_d, total_d + 1)]


    plt.figure(figsize=(8, 6))

    plt.plot(x_labels, power_isolated_star, marker='o',
             linestyle='-', label='Isolated star test')
    plt.plot(x_labels, power_d_avg, marker='o',
             linestyle='-', label='Average distance test')

    plt.xlabel('Dimension')
    plt.ylabel('Power')
    plt.legend()
    plt.grid(True)
    plt.show()
```

A.2 Visualize the graphs

This function is used to generate Figure 2.1.

```
def visualize_graph(graph_bot, bot):
    """
    Visualizes the given graph using a spring layout provided by `networkx`.
    Highlights in red the vertices in the list bot and the edges they make.

    Args:
        - graph_bot: a networkx graph.
        - bot: a list of vertices in the botnet.
    """
    plt.figure(figsize=(8, 8))

    pos = nx.spring_layout(graph_bot)

    # Create a list of colors for vertices, red if in bot, black otherwise
    node_colors = ['red' if node in bot else 'blue' for node in graph_bot.nodes()]
    # Create a list of colors for edges, red if either vertex is in bot, black otherwise
    edge_colors = ["red" if u in bot or v in bot else "black" for u, v in graph_bot.edges()]

    nx.draw(graph_bot, pos, with_labels=False, node_size=25,
            node_color=node_colors, edge_color=edge_colors, alpha=0.5)

```

```python
    plt.title('Random Geometric Graph with Force-Directed Layout (Botnet in Red)')
    plt.show()
```

This function is used to generate Figure 2.2.

```python
def visualize_embedding_graph(graph_bot, points, bot):
    """
    Visualizes the given random geometric graph in the embedding space,
    with periodic boundary conditions.
    Highlights in red the edges made by the vertices in the list bot.

    Args:
        - graph_bot: a networkx graph.
        - points: positions of the points in the embedding space.
        - bot: a list of vertices in the botnet.
    """
    plt.figure(figsize=(8, 8))
    for i in range(n):
      for j in range(i + 1, n):
        if graph_bot.has_edge(i, j):
          x1, x2 = points[i][0], points[j][0]
          y1, y2 = points[i][1], points[j][1]

          if i in bot or j in bot:
            plt.plot([x1, x2], [y1, y2], 'r-', linewidth=0.5, alpha=0.5)
          else:
            # Check if wrapping occurs in either dimension
            wrap_x = np.abs(x1 - x2) > 0.5
            wrap_y = np.abs(y1 - y2) > 0.5

            if not wrap_x and not wrap_y:
              # No wrapping, draw a single line
              plt.plot([x1, x2], [y1, y2], 'k-', linewidth=0.5, alpha=0.5)
            elif wrap_x and not wrap_y:
              # Wrapping in x-dimension
              if x1 > x2: #Ensure x1 < x2
                x1, x2 = x2, x1
                y1, y2 = y2, y1

              plt.plot([x1, x2-1], [y1, y2], 'k-', linewidth=0.5, alpha=0.5)
              plt.plot([x2, 1+x1], [y2, y1], 'k-', linewidth=0.5, alpha=0.5)

            elif not wrap_x and wrap_y:
              # Wrapping in y-dimension
              if y1 > y2: #Ensure y1 < y2
                x1, x2 = x2, x1
                y1, y2 = y2, y1

              plt.plot([x1, x2], [y1, y2-1], 'k-', linewidth=0.5, alpha=0.5)
              plt.plot([x2, x1], [y2, y1+1], 'k-', linewidth=0.5, alpha=0.5)

            elif wrap_x and wrap_y:
              # Wrapping in both x and y dimensions
              if x1 > x2:
                x1, x2 = x2, x1
                y1, y2 = y2, y1

              if y1 > y2:
                plt.plot([x1, x2-1], [y1, y2+1], 'k-', linewidth=0.5, alpha=0.5)
                plt.plot([x2, x1+1], [y2, y1-1], 'k-', linewidth=0.5, alpha=0.5)
              else:
                plt.plot([x1, x2-1], [y1, y2-1], 'k-', linewidth=0.5, alpha=0.5)
                plt.plot([x2, x1+1], [y2, y1+1], 'k-', linewidth=0.5, alpha=0.5)

    plt.plot(points[:, 0], points[:, 1], 'o', markersize=5)
    plt.title('Random Geometric Graph in Embedding Space (with periodic boundary)')
    plt.xlim(0, 1)
    plt.ylim(0, 1)
    plt.show()
```

Bibliography

[1] G. Bet, K. Bogerd, R. M. Castro, and R. van der Hofstad. Detecting a botnet in a network. *Mathematical Statistics and Learning*, 3(3):315–343, 2021.

[2] F. Den Hollander. Probability theory: The coupling method. *Lecture notes available online (http://websites. math. leidenuniv. nl/probability/lecturenotes/CouplingLectures. pdf)*, 3, 2012.

[3] R. B. Ellis, J. L. Martin, and C. Yan. Random geometric graph diameter in the unit ball. *Algorithmica*, 47(4):421–438, 2007.

[4] M. Feily, A. Shahrestani, and S. Ramadass. A survey of botnet and botnet detection. In *2009 third international conference on emerging security information, systems and technologies*, pages 268–273. IEEE, 2009.

[5] H. O. Georgii. *Stochastics: introduction to probability and statistics*. Walter de Gruyter, 2012.

[6] J. M. Hammersley. The distribution of distance in a hypersphere. *The Annals of Mathematical Statistics*, 21(3):447–452, 1950.

[7] W. Hoeffding. A class of statistics with asymptotically normal distribution. *The Annals of Mathematical Statistics*, 19(3):293–325, 1948.

[8] N. Leijenhorst and D. de Laat. Solving clustered low-rank semidefinite programs arising from polynomial optimization. *Mathematical Programming Computation*, 16(3):503–534, 2024.

[9] S. Maneewongvatana and D. M. Mount. Analysis of approximate nearest neighbor searching with clustered point sets. *arXiv preprint cs/9901013*, 1999.

[10] M. Mitzenmacher and E. Upfal. *Probability and computing: Randomization and probabilistic techniques in algorithms and data analysis*. Cambridge university press, 2017.

[11] M. Penrose. *Random geometric graphs*. OUP Oxford, 2003.

[12] A.B. Tsybakov. *Introduction to Nonparametric Estimation*. Springer Series in Statistics. Springer New York, 2008.