NIST Technical Note 2035

A General Methodology for Deriving Network Propagation Models of Computer Worms

Shuvo Bardhan
Douglas Montgomery
James Filliben
Alan Heckert

This publication is available free of charge from:
https://doi.org/10.6028/NIST.TN.2035

National Institute of
Standards and Technology
U.S. Department of Commerce

NIST Technical Note 2035

A General Methodology for Deriving Network Propagation Models of Computer Worms

Shuvo Bardhan
Douglas Montgomery
Advanced Networks Technology Division
Information Technology Laboratory

James Filliben
Alan Heckert
Statistical Engineering Division
Information Technology Laboratory

This publication is available free of charge from:
https://doi.org/10.6028/NIST.TN.2035

February 2019

U.S. Department of Commerce
Wilbur L. Ross, Jr., Secretary

National Institute of Standards and Technology
Walter Copan, NIST Director and Undersecretary of Commerce for Standards and Technology

National Institute of Standards and Technology Technical Note 2035
Natl. Inst. Stand. Technol. Tech. Note 2035, 45 pages (February 2019)
CODEN: NTNOEF

This publication is available free of charge from:
https://doi.org/10.6028/NIST.TN.2035

Abstract

Externally-launched computer worms which maliciously propagate within networks are one of the most serious and dangerous security threats facing the commercial, political, military, and research communities today. With an eye to the ultimate goal of detection and prevention of such worms, the purpose of this paper is twofold: to develop predictive models for the number of infected hosts per iteration and the number of iterations to saturation, and to present a systematic methodology (simulator construction + data generation + 2 sequential fitting steps) for the construction of such models. This methodology will have application across a variety of worm-modeling scenarios. These models will have 3 core factors known to affect worm propagation : size of the network space, proportion of the space with susceptible hosts, and rate at which an infected host scans for other vulnerable hosts; three additonal factors will then be added for exceptionally large networks. Further, this paper presents a worm propagation sensitivity analysis which provides valuable insight into the most important factors (and interactions) affecting worm propagation speed. For demonstration purposes (and with no loss of generality), we apply this methodology to a 3-factor class B ($= 2^{16} - 1$ IP Addresses) network and derive a high quality predictive model (error $< 4\%$).

Key words

Computer Network, Network Propagation Models, Propagation Methodology, Sensitivity Analysis, Experiment Design, Predictive Models, Worm Modeling.

Table of Contents

List of Tables

List of Figures

1. Introduction

Computer worms are one of the most dangerous forms of cyber attack on the internet [1][2]. Once a computer worm infects a host, the worm propagates to infect other hosts and the infected hosts are collectively being used for (i) Distributed Denial of Service (DDoS) attack [3], (ii) Phishing [4], and (iii) Exfiltration. In 2001, Code-Red (computer worm) infected 359,000 hosts all across the globe and launched a DDoS attack, causing an estimated damage of $2.6 billion [5].

The purpose of this paper is twofold: to develop predictive models for the number of infected hosts per iteration and the number of iterations to saturation, and to present a systematic methodology (simulator construction + data generation + 2 sequential fitting steps) for the construction of such models. This methodology will apply to both global propagation worms and to local / internal over a relevant IP (Internet Protocol) v4 (Version 4) address subspace (we use, for example, the IPv4 Class B Address subspace of size $\Omega = 2^{16}$ as shown in Figure 1)[6, 8]. In our paper, we take a data-driven approach consisting of 6 steps :

1. Define a population space over which we want our model to be valid.
2. Determine factors and range of factors settings that span the population space.
3. Determine a representative subset of the population that may be sampled.
4. Construct a generic worm simulator—valid over the population space.
5. Generate simulator data over a subset of relevant representative conditions in the population space.
6. Carry out a specific 2-step modeling sequence: local, then global, to create a final general model valid over the specified population space.

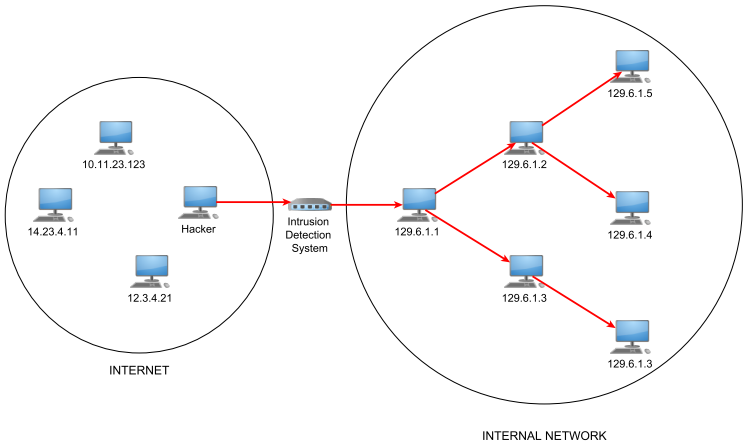

Fig. 1. Generalized Network Propagation of Computer Worms (Internal Network IP Address Space Size = $\Omega = 2^{16}$)

Over time, there have been several worm propagation models presented in the literature [1]; though serving as advancements, these models were theoretical in nature, did not present fitted model parameter values, nor estimates of model error. The novelty of our approach (and how it differs from the literature) is that it is "data-driven", has elements of both formal experiment design and statistical analysis intrinsic to the methodology, does present relevant fitted model parameters, and does yield a fully-fitted model with small predictive error ($< 4\%$).

The organization of the paper is as follows:

1. *Introduction*: this section discusses the relevance of our work and the paper itself.
2. *Related Work*: elaborates on the related worm propagation models in literature and the motivation.
3. *Experiment Design*: deals with the scope of our experiment, factors and levels and the experiment design matrix (8 design points) we have decided for our experiment.
4. *Data Generation / Collection*: deals with description of the generic algorithm and the generated 8 design points for our experiment.
5. *Data Analysis*: deals with the sensitivity analysis and optimization study, local and global modeling using the 8 design points data.
6. *Conclusion*: concisely discusses the impact and importance of our methodology and model.

2. Scan-Based Models

The modeling of computer worm propagation can be broadly divided into several types, the two most noteworthy being Scan-Based Models, which make use of infection probabilities, population subsets and host scanning rates; and Topology-Based Models, which make use of and exploits additional network topology information (if known). Our work focuses on Scan-Based Models, which have two further components, based on domain-scope [1]:

1. Homogeneous models in which the contamination is global (e.g., world-wide-web);
2. Localized models in which the contamination is confined within a subnetwork (e.g., NIST).

A seminal paper on Scan-Based models was Chen[7] in 2007, which presents solutions for both the Homogeneous and Localized cases. The Chen Scan-Based Homogeneous (internet wide) Model is referred to as the AAWP (Analytical Active Worm Propagation) Model. In their paper at any discrete time tick t, the number of hosts is denoted by m_t, the number of infected hosts is denoted by n_t and the number of initial infected hosts is denoted by h_0 (i.e. at time $t = 0$, n_0=h_0). If the Scanning Rate of the worm is s, then the *AAWP* number of infected hosts at a given time tick t is given by

$$n_{t+1} = n_t + (m_t - n_t)\left[1 - \left(1 - \frac{1}{2^{32}}\right)^{sn_t}\right] \tag{1}$$

If the notions of Death Rate (d) and Patching Rate (P) are introduced, then the *AAWP* number of infected hosts at each time tick t is given by

$$n_{t+1} = n_t + (m_t - n_t)\left[1 - \left(1 - \frac{1}{2^{32}}\right)^{sn_t}\right] - (d+p)n_t \tag{2}$$

Chen's Scan-Based Localized Model is referred to as the LAAWP (Local AAWP) model, and is a special case of Localized Models known as Discrete Time Models. For simplicity, the authors omit Death Rate and Patching Rate in this model. For their solution, they take into account three probabilities (i) p_0–scans a random address, (ii) p_1–scans an address with the same first octet and (iii) p_2–scans an address with the same first two octets, where $\sum_{i=0}^{2} p_i = 1$. They applied their model to three subnets:

- subnet 1 ($\Omega = 2^8 - 1$), where the first octet is fixed;
- subnet 2 ($\Omega = 2^8 - 1$) having same first octet like subnet 1, but with a smaller hit list;
- subnet 3 ($\Omega = 2^{16} - 2^8$), having a larger population.

For the *LAAWP* Model, the average number of infected hosts and the average number of scans hitting subnet i, are represented by b_i and k_i respectively, where $i = 0, 1, 2$.

$$\begin{aligned}
k_1 &= p_2 s b_1 + p_1 s[b_1 + (2^8 - 1)b_2]/2^8 \\
&\quad + p_0 s[b_1 + (2^8 - 1)b_2 + (2^{16} - 2^8)b_3]/2^{16} \\
k_2 &= p_2 s b_1 + p_1 s[b_1 + (2^8 - 1)b_2]/2^8 \\
&\quad + p_0 s[b_1 + (2^8 - 1)b_2 + (2^{16} - 2^8)b_3]/2^{16} \\
k_3 &= p_2 s b_3 + p_1 s b_3 \\
&\quad + p_0 s[b_1 + (2^8 - 1)b_2 + (2^{16} - 2^8)b_3]/2^{16}
\end{aligned}$$

Using the *LAAWP* Model, the number of infected hosts b_i is derived to be–

$$b_{i+1} = b_i + (\frac{N}{2^{16}} - -b_i)n_i\left[1 - \left(1 - \frac{1}{2^{16}}\right)^{k_i}\right] \tag{3}$$

3. Motivation

Our model is different than Chen's model in many ways. Our model provides an alternate approach to derive worm propagation equations from synthetic data. We provide a generic algorithm that could be used for IP scanning and email scanning. We provide realism by experimenting with different seeds in the Pseudo-Random Number Generator (PRNG). Idealy, the comparison and ranking of IDS should be accurate for both benign traffic and malicious traffic. Chen's model would allow accurate IDS comparison, but for scanning worms only. Our generic algorithm on the contrary could be extended to allow accurate IDS comparison for scanning worms, flash worms, email worms, router worms, and botnets.

4. Experiment Design

In this section, we present the experiment design of our paper. We initially discuss the scope, which presents the various factors associated with the propagation of computer worms in a network. The factors which we have chosen for our experiment illustrated in the next section 3.1. Finally, we conclude this section with the Experiment Design Matrix.

Worm Behavior in the real world depends on many factor, some depending on the host, some depending on the environment, and some depending on worm specs itself. Among host factors one would include the number of scannable hosts in an enterprise, the number of infectible hosts, and the number of initially infected hosts. Environment factos would include network topology and death and patching rates. Worm factors would include scanning rates. Following Chen's lead, our study will focus on 6 of these 7 factors, reserving the network topology for the subject of another paper.

¡In reality, one can come up with several factors which affect worm propagation apart from the ones we have taken. We have purposely chosen these factors to illustrate our methodology–keeping chen's model as a reference. Our methodology serves as an improvement by not only ecompassing the existing types of worms, but also encompassing future worms.

From these factors we will develop an improved model (and model-building methodology) which will have a significantly broader and more robust range of applications coverting not just (Chen's) scanning worms but also router worms, email worms and botnets.¿

4.1 Factors

In [7], Chen et al. have presented a general model on the propagation of computer worms. That model includes the following $k = 6$ factors:

1. *Population Size* (Number of Hosts Scannable).
2. *Hit List* (Number of Hosts Infectible).
3. *Scanning Rate* (Number of probes/sec scannable by an infected host).
4. *Number of Initial Infected Hosts*.
5. *Death Rate* (Number of infected hosts/second getting "disconnected"/"dead").
6. *Patching Rate* (Number of infected hosts/second getting "recovered").

The factors and settings in our experiment have been derived from Chen's model (see column 1 of Table 1). For our convenience, we have chosen to rename Chen's "Hit List" factor as "Susceptible Proportion" (see Factor $X2$). This represents the proportion of the Population Size (IP Address Space) [8] of a network which could be infected by the computer worm.

From these factors we will develop an improved model (and model-building methodologies) which will have a significantly broader and more robust range of applications, covering not just Chen's Scanning worms, but also flash email worms, router worms and even botnets.

4.2 Scope

Every experiment has a specified (conceptual) range (That is , "scope") of factor values (population) over which a derived model is deemed valid. To be useful, these factor ranges must be based in reality and practice. The population/scope of the 6 factors in our experiment is shown in column 2 of Table 1:

1. For $X1$ (Population Size), the population Ω of admissible values is 2^{24} for Class A blocks, 2^{16} for Class B blocks, and 2^8 for Class C blocks. This is in line with practical domain sizes encountered in a moderately-sized enterprise (100,000 hosts).
2. For $X2$ (Susceptible Proportion), the probability range is (obviously) 0 to 1.
3. For $X3$ (Scanning Rate), the chosen range is 10 to 100 probes per second.
4. For $X4$ (Number of Initial Infected Hosts), the value ranges from 1 to Np.
5. For $X5$ (Death Rate), the "reasonable" population is 0 to 0.001 (e.g. if at a point in time, a total of 7000 hosts become infected then 7 hosts would be "disconnected", "dead" or "eliminated without patching").
6. For $X6$ (Patching Rate), the chosen rate is 0 to 0.0005 (e.g. if at a point in time Np = 20,000 then 10 hosts would become "recovered" or "invulnerable").

Table 1. Scope–Factors and Settings

Factors	**Applicable Settings**	**Chosen Settings**
$X1$–*Population Size (N)*	Class A Blocks ($\Omega = 2^{24}$) Class B Blocks ($\Omega = 2^{16}$) Class C Blocks ($\Omega = 2^8$)	{64000 , 128000 }
$X2$–*Susceptible Proportion (p)*	[0 , 1]	{0.25 , 0.75}
$X3$–*Scanning Rate (r)*	[10 , 100]	{10 , 50}
$X4$–*Number of Initial Infected Hosts (n)*	[1 , $< Np$]	{ 1 }
$X5$–*Death Rate (d)*	$[0, 0.001]$	{ 0 }
$X6$–*Patching Rate (P)*	$[0, 0.0005]$	{ 0 }

4.3 Settings

Beyond the population and scope of the factor is the additional issue as to what settings (levels) the factor should take on during the course of the conducted experiment. Such settings are dictated by the total number of runs affordable in the experiment in concert with what settings constitute a "representative" subsampling from the larger population. For our initial experiment, the chosen settings for the 6 factors are given in columnn 3 of Table 1:

1. For $X1$ (Population Size), we have chosen 2 settings (N=64000 and N=128000 hosts).
2. For $X2$ (Susceptible Proportion), we have chosen 2 rates: p=0.25 and p=0.75.
3. For $X3$ (Scanning Rate), the 2 values selected were 10 and 50 probes/second.
4. For the 3 Factors $X4$ (Number of Initial Hosts Infected), $X5$ (Death Rate), and $X6$ (Patching Rate), the following single settings were chosen (1, 0, and 0, respectively) to reflect the fact that all 3 of these factors were fixed in our initial experiment.

For justification of the settings of Factors 3 through 6, see Chen [7].

4.4 Initial Modelling

As a first pass we have chosen to focus our modeling effort on the propagation of computer worms on internal networks, and so the number k) of factors under investigation will subsequently be reduced. In particular, the underlying assumptions in our experiment are that once a computer worm penetrates an internal network it is undetectable while it propagates. In such case factors $X5$ (Death Rate) and $X6$ (Patching Rate) do not apply in our model and hence our system reduces from $k = 6$ to $k = 4$ factors. These factors will be reincorporated later in the manuscript when we generalize our model from internal to world wide.

Further, we have found factor $X4$ (the Number of Initial Infected Hosts) to be redundant in our experiment design since the expanded model for Y = number of infected hosts:

$$Y = g(t, X1, X2, X3, X4) \tag{4}$$

will be found to be identical to the more parsimonious model:

$$Y = f(t + X4, X1, X2, X3) \tag{5}$$

Thus our system under study will initially consist of the iteration factor t plus $k = 3$ domain factors. Other factors (known or unknown) could of course come into play by potentially having an effect, but the purpose of this paper is to not only derive meaningful worm behavior results, but also to demonstrate an underlying model-building methodology which would encompass all such factors–however many in number. In this context, we have chosen this smaller number ($k = 3$) of factors at first pass with the thought that extension to a larger number of factors should be evident.

4.5 Design Matrix

From the previous section, we have k=3 (domain-definition) factors. The appropriate experiment design is dictated by the desired scope on the one hand, and time/cost considerations on the other. In light of how a researcher might use our approach in practice and in the interest of specificity, we (arbitarily) set the data generation / collection component of our experiment to be 10 observations or less for our initial experiment.

In order to accommodate estimation of factor effects and interactions, we could thus run an ideal default 2^3 full factorial design. This design has k = number of factors = 3 , and n = number of runs = 8. The design is given tabularly (in Table 2) and graphically (in Figure 2):

Table 2. Experiment Design Matrix

Run #	Factors in Original Units			Factors in Coded Units		
	Population Size N	**Susceptible Proportion** p	**Scanning Rate** r	**X1**	**X2**	**X3**
1	64000	0.25	10	-1	-1	-1
2	128000	0.25	10	+1	-1	-1
3	64000	0.75	10	-1	+1	-1
4	128000	0.75	10	+1	+1	-1
5	64000	0.25	50	-1	-1	+1
6	128000	0.25	50	+1	-1	+1
7	64000	0.75	50	-1	+1	+1
8	128000	0.75	50	+1	+1	+1

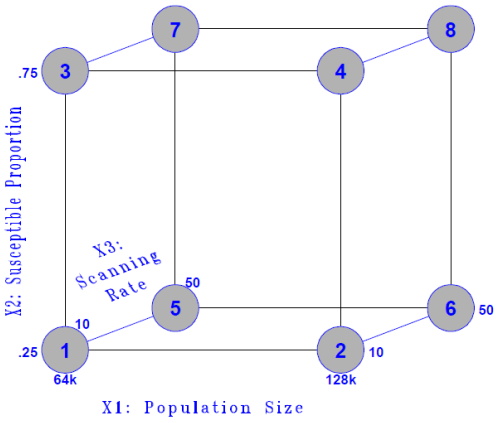

Fig. 2. Graphical Representation of Runs 1–8 (Given in Table 2)

In Figure 2, we number the 8 vertices 1 to 8 which corresponds to the "Yates" [10] order given in Table 2. We emphasize that other designs involving more levels and with more replications could have been used, but we chose this design because, though simple, it still allows us to apply the full range of analysis operations that will result in a good-fitting final model. Note also, that though this design has but 3 factors and 2 levels, it will prove to be an excellent first step in providing insight into–

1. Important Factors: the relative importance of the 3 Factors–$X1$, $X2$ and $X3$ (and their interactions).
2. Optimal Settings: the best (and worst) Settings of those 3 Factors.
3. Predictive Model: for a given iteration t and for a given ($X1$, $X2$, $X3$), a curve relating the number of individual infected-hosts curve as a function of (t,$X1$,$X2$,$X3$).

5. Data Generation/Collection

The purpose of this section is to describe the algorithmic components that were incorporated to generate (for each of the 8 points specified in the above experiment design matrix) the data set by which our analysis approach could be applied in generating the desired final fitted model.For each ($X1$,$X2$,$X3$), the output from this section will be a time series consisting of the response $Y(t)$ (= number of infected hosts) as a function of iteration t. This will continue until saturation is achieved; i.e., until all hosts have become infected.

5.1 Algorithm: Worm Propagation

Here we present our algorithm which describes the steps involved in the propagation of a computer worm in a generalized network. This algorithm is general in the sense that it accommodates not only the above 3 factors ($X1$ to $X3$), but also makes provision for the 3 additional factors: $X4$: Number of Initial Infected Hosts, $X5$: Death rate, and $X6$: Patching rate (which we fixed for purposes of demonstrating our approach).

The output of the algorithm is the list of infected IP address's at the end of each iteration of scanning done by the infected hosts. Table 3 describes the lists, important variables and functions used in Algorithm 1.

Algorithm 1 has 3 loops–one outer while loop and two inner loops. The outer-while loop deals with the termination of the algorithm. The inner while-loop creates random IP addresses to be scanned (by the infected hosts). The inner nested for-loop deals with the identification of newly infected hosts by the computer worm. Over time, the number of infected hosts i will equal the number of susceptible hosts s and the algorithm will terminate.

Table 3. Generator Algorithm–Parameters and Pseudo Code

Type	Name	Description
Lists	List_IP	List of IP Address's in IP Space.
	List_SP	List of Susceptible IP Address's.
	List_IIP	List of Infected IP Address's.
	Rand_IP	List of Random IP Address's.
	List_INF	List of the Number of Infected Hosts (IP Address's) in each iteration.
Variables	N	Number of hosts in a network ($X1$).
	p	Proportion of hosts susceptible to the computer worm ($X2$).
	r	Scan Rate of the worm ($X3$).
	n	Number of Initial Infected Hosts ($X4$).
	d	Death Rate of the worm ($X5$).
	P	Patching Rate of the worm ($X6$).
	i	Number of newly infected hosts (per iteration).
Functions	Random (L , n)	Function which returns n random IP addresses from a list of IP Address's (L) in the form of a List.

Algorithm 1 Generalized Network Worm Propagation

Input: List_IP , N , p , r, n, d, P

Output: List_INF

1: Rand_IP $\leftarrow \phi$
2: List_INF $\leftarrow \phi$
3: $i \leftarrow n$
4: List_SP = Random (List_IP , $|\,(p \times N)\,|$)
5: **while** $i \; < \; |\,(p \times N)\,|$ **do**
6: $\quad k \leftarrow 0$
7: $\quad$ **while** $k < (i$ - $(d + P) \times i)$ **do**
8: $\quad\quad$ Rand_IP $\uplus$ { Random (List_IP , p) }
9: $\quad\quad k \leftarrow k + 1$
10: $\quad$ **for** ip_i in Random_IP **do**
11: $\quad\quad$ **for** ip_j in List_SP **do**
12: $\quad\quad\quad$ **if** $ip_i == ip_j$ *and* $ip_i \notin$ List_IIP **then**
13: $\quad\quad\quad\quad$ List_IIP $\uplus$ { ip_i }
14: $\quad\quad\quad\quad i \leftarrow i + 1$
15: $\quad$ List_INF $\uplus$ { i }
16: $\quad$ Rand_IP $\leftarrow \phi$
17: **return** List_INF

5.2 Data

Application of Algorithm 1 to the 8 design points in Table 2 yielded 2 responses ($Y(t)$ = cumulative number of infected hosts through iteration t and Y' = minimum number of iterations-to-saturation) for each iteration (column 5 of Table 4). This was continued until full saturation (all hosts in the population become infected). The number of iterations-to-saturation will of course vary depending on the specified ($X1$,$X2$,$X3$) settings. Table 4 below shows the synthetically generated response for the 8 design points.

Table 4. Response Yield

Run Id	Factors			Response Y	Response Y'
	X1 *Pop. Size*	**X2** *Susc. Prop.*	**X3** *Scan. Rate*	(*Cumulative number of Infected Hosts per iteration*)	(*Min. number of iterations to Saturation*)
1	-1(64K)	-1(.25)	-1(10)	1,3,4,44,158,1890,5556,11684,15279,15935, 15997,16000.	12
2	+1(128K)	-1(.25)	-1(10)	1,3,10,32,110,375,1306,4264,12107,24240, 30842,31880,31988,31999,32000.	15
3	-1(64K)	+1(.75)	-1(10)	1,7,64,554,4468,26323,47667,48000.	8
4	+1(128K)	+1(.75)	-1(10)	1,6,50,427,3571,26065,86856,95987,96000.	9
5	-1(64K)	-1(.25)	+1(50)	1,12,162,2093,13306,16000.	6
6	+1(128K)	-1(.25)	+1(50)	1,10,120,1610,15749,31956,32000.	7
7	-1(64K)	+1(.75)	+1(50)	1,36,1374,32031,48000.	5
8	+1(128K)	+1(.75)	+1(50)	1,36,1353,40084,96000.	5

Note the first design run ($X1$,$X2$,$X3$) = (-1,-1,-1) = (64000,0.25,10). The generator started at iteration t=0 with 1 host infected, and then took 11 more iterations until saturation (= all of the $64000 \times 0.25 = 16000$ hosts being infected). For this first case, these 1+11 values are sufficient to carry out the modeling of

$$Y(t) = \text{number of infected hosts} = f_1(t,-1,-1,-1).$$

Similarly the second design run ($X1$,$X2$,$X3$) = (+1,-1,-1) = (128000,0.25,10) yields 1+13 values; this is sufficient to carry out the modelling of

$$Y(t) = f_2(t,+1,-1,-1), \tag{6}$$

and so on for all 8 design points. These 8 functions f_1, f_2, $\ldots$ f_8 will thus serve as a basis for deriving a universal function

$$Y = g(t,X1,X2,X3) \tag{7}$$

to encompass all of the domain definition factor values.

Figure 3 is a graphical display of the data in Table 2.

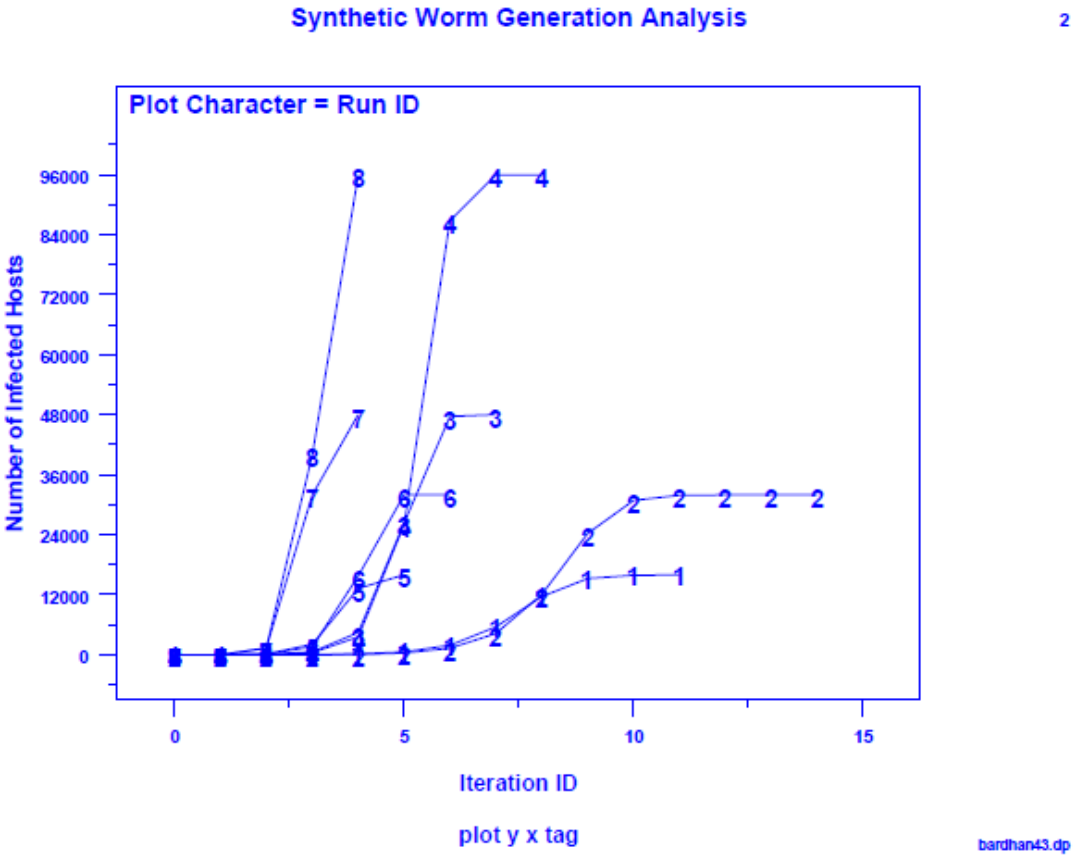

Fig. 3. Multi-trace plot of the infected number of hosts as a function of the Iteration ID t for each of the 8 domain-definition specifications in the design matrix.

In Figure 3, it is seen that

1. All traces are monotonically increasing (with a sigmoid shape).
2. Some traces for e.g., 8 is steep and where saturation is achieved quickly.
3. Other traces for e.g., 1 are more elongated and where saturation takes a much longer time.
4. The number of iterations-to-saturation is modest (between 4 and 14).

An alternate graphical representation of the generated data is the following (Figure 4) cube-plot:

The left and right cube faces correspond to $X1$ (Population Size) being small ($64k$) and large ($128K$) respectively. Note that each of the 8 nodes of the cube shows the cumulative number of infected hosts as a function of iteration t. Similarly, the bottom and top faces are $X2$ (Susceptible Proportion: .25 and .75) and the front and back faces are $X3$ (Scanning Rate: 10 and 50).

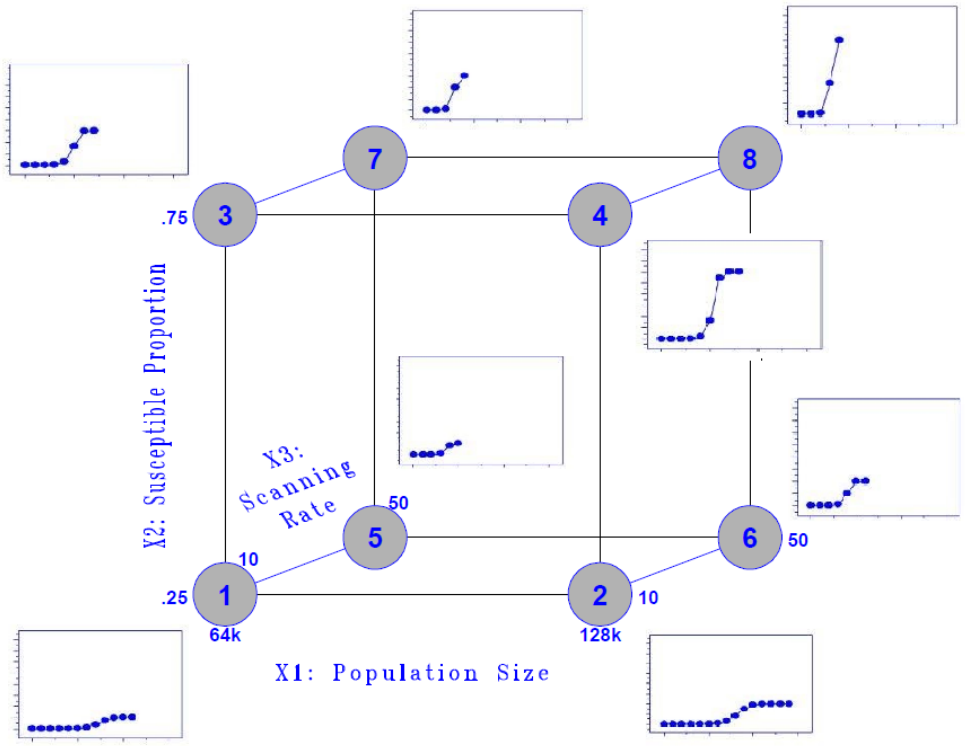

Fig. 4. Cube-trace plot of the infected number of hosts as a function of the Iteration ID t for each function of the 8 domain-definition specification in the design matrix

6. Data Analysis

6.1 Sensitivity and Optimization Study

Sensitivity analysis is the process of determining those factors (and interactions) that most effect a particular response of interest. This determination is a necessary and insightful first step to achieving the ultimate objective of this manuscript as a whole–to produce a global predictive function f:

$$Y = f(t, X1, X2, X3) \tag{8}$$

where

Y = *Cumulative number of infected hosts*,
t = *Iteration ID*,
$X1$ = *Population Size*,
$X2$ = *Susceptible Proportion*, and

$X3$ = *Scanning Rate*.

Our approach is to first carry out a classical sensitivity analysis on the $k = 3$ Factors (and interactions) and then incorporate iteration information later.

This sensitivity analysis is preliminary (but essential) to the actual modeling process. Sensitivity analysis provides the information (and insight) as to what factors are most (and least) critical. Sensitivity analysis will not translate that importance determination into a quantitative weighting–that task is relegated to the formal model-fitting, which is the phase 2 component in our methodology.

With respect to which of the $k = 3$ factors most affect systems responses, we present Figures 5, 6 and 7, which embed factor setting information within the plot via character and color. The plot characters in Figure 5 are the two Population Sizes (blue = $64K$ and red = $128K$); the plot characters in Figure 6 denote the two Susceptible Proportions (blue=25% and red=75%); and the plot characters in Figure 7 show the two Scanning Rates (blue=10 and red=50).

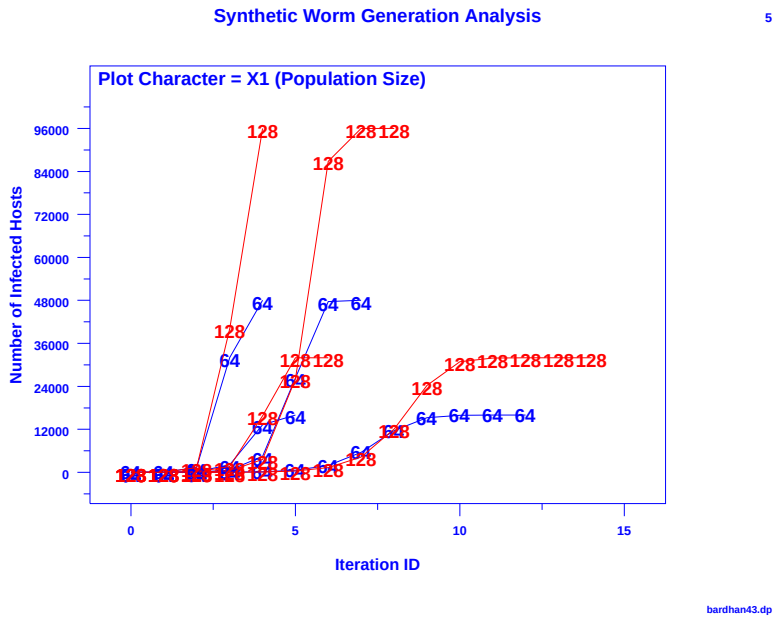

Fig. 5. Plot of the Cumulative Number of Infected Hosts versus Iteration ID with Plot Character representing $X1$ = Population Size

The three plots on the raw data give us as an initial view of the effect of various factors on the response. The most striking conclusion from the three plots comes from Figures 6 and 7. In Figure 7, e.g., it is seen that the preponderance of the response traces on the right half of the plot (= the larger iteration half) are blue (= low Scanning Rate of 10) which is consistent with the expected behavior that low (= 10) Scanning Rates will take longer to

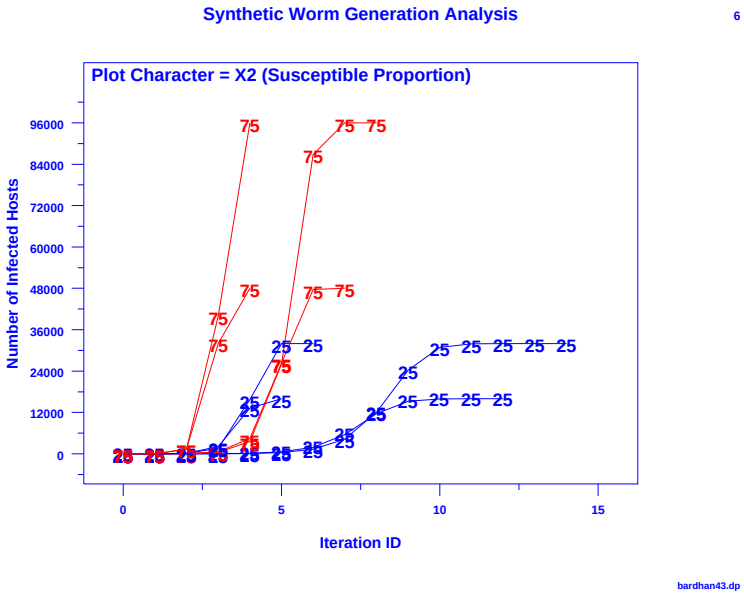

Fig. 6. Plot of the Cumulative Number of Infected Hosts versus Iteration ID with Plot Character representing $X2$ = Susceptible Proportion

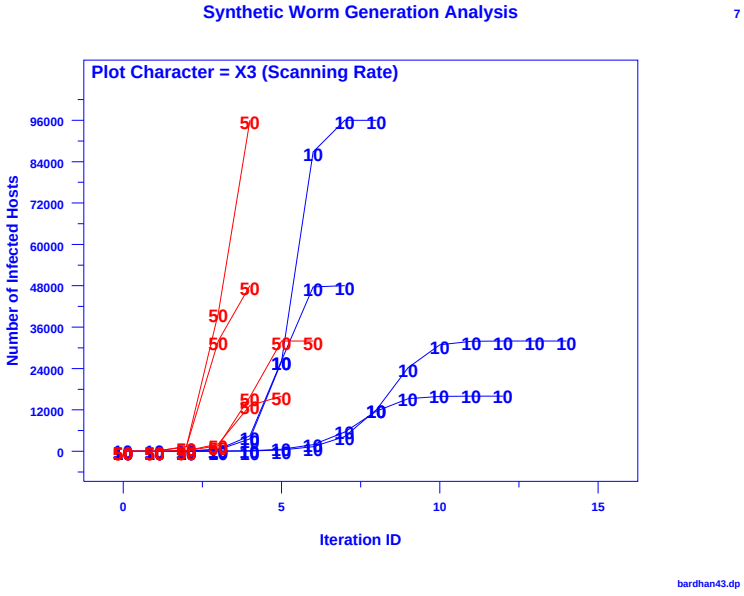

Fig. 7. Plot of the Cumulative Number of Infected Hosts versus Iteration ID with Plot Character representing $X3$ = Scanning Rate

saturate, while high (= 50) Scanning Rates will saturate faster (= shorter iterations); the appearance of Figure 7 affirms that $X3$ (=Scanning Rate) is an important (and, as it turns out the most important) of the 3 factors regarding its influence on Y = cumulative number of the infected hosts (and hence also Y' = the number of iterations-to-saturation).

Figure 8 is a cube-plot in which each node shows Y' = number of iterations-to-saturation. The Y' = 12 shown in the lower left corner of the cube indicates that for this (-1,-1,-1) = (64K,0.25,10) condition it took 12 iterations to reach saturation (as confirmed in row 1 of Table 4).

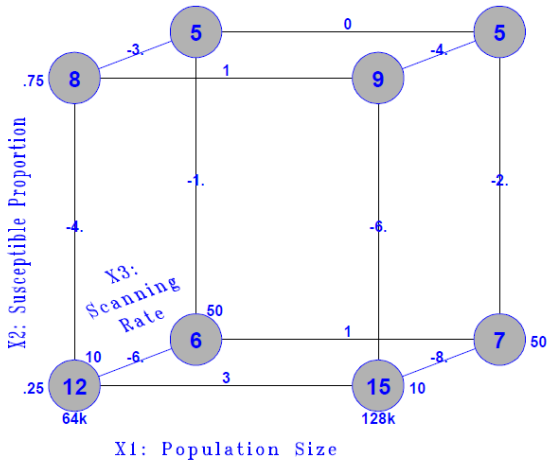

Fig. 8. Cube plot for Y' = Number of iterations-to-saturation

It is clear as one precedes from the low Scanning Rate (= front of cube) to the high Scanning Rate (back of cube), the iterations-to-saturation decrease in all 4 cases (11 to 5, 14 to 6, 7 to 4, and 8 to 4), thus

1. $X3$ = Scanning Rate is an important factor;
2. the more severe setting is on the back plane ($X3$ = 50 = Faster Scanning Rate);
3. both of the above two conclusions are robustly true over all 4 ($X1$: Population Size, $X2$: Susceptible Proportion) combinations.

Note also that the biggest Scanning Rate effect (=15-7=8) occurs at the (Population Size = + = 128K, Susceptible Proportion = - = 0.25) combination.

Figure 9 is a main effects plot–the most important tool for ascertaining the relative importance of the 3 factors under study: The horizontal axis gives the 3 factors under study along with the 2 levels for each factor. The vertical axis is the mean number of iterations-to-saturation for each of the 2 levels of the 3 factors under study.

It is noteworthy that for a given factor, the (simple) vertical distance (=difference) between the means for the 2 levels on a main effects plot is identical to the least squares estimate of the factor effect. Steep lines imply important factors; shallow lines indicate less-important factors. The annotation under the plot lines for each factor are estimated (least squares factor effect), the relative (to the grand mean) factor effect, and the cumulative distribution function of the ANOVA F distribution for testing factor statistical significance (values in excess of 95% imply significance).

From the main effects plot, we conclude–

1. In this case, factor $X3$ (Scanning Rate) is seen to be the most important factor with an estimated effect of 5.25 that is, Y' = the number of iterations-to-saturation is reduced on the average by 5.25 iterations as $X3$ proceeds from a front plane Scanning Rate value of 10 to a back plane Scanning Rate value of 50. Note that this least square estimate of 5.25 for the $X3$ effect is identical to the average of the 4 local differences: 12-6 = 6, 15-7 = 8, 8-5 = 3, 9-5=4. Relatively speaking, this average difference of 5.25 is quite large (compared to the global average of all the 8 values (=7.375)); The relative effect is (5.25/7.375) $\times$ 100 = 71% as noted on the plot. Finally, this 5.25 effect size turns out to be statisticaly significant (via the usual one-way ANOVA) and hence is highlighted in red.

2. Factor $X2$ (Susceptible Proportion) is the next most important factor. On the average, as $X2$ proceeds from a bottom plane Susceptible Proportion value of .25 to a top plane value of .75, the Y' = number of iterations-to-saturation decreases by 3.25 (44%). While large, this result is not statistically significant.

3. Finally factor X1 (Population Size) is the least important factor on the average, as Y' goes from left plane to Population Size of 64K to right plane Population Size of 128K, the number of iterations-to-saturation increases by (only) 1.25 iterations (17%) and is not statistically significant.

6.2 Local Modeling

The next step in the modeling process is to create a set of 8 narrow-domain (=$f_i(t)$) functions f_1 to f_8 (that is, each function f_i is valid only for a single ($X1$,$X2$,$X3$) combination and thus only for a single vertex in the design-matrix cube). It will be seen that each of these 8 functions produces a high-accuracy fit (as a function of t) at its particular design point and local environments.

For guidance as to the particular admissible functional forms for each member of this set of 8, we make note the similar shape of all 8 sigmoid curves in Figures 5, 6 and 7, namely they are:

1. Monotonically increasing, with
2. a well defined lower plateau, and
3. a well defined upper plateau.

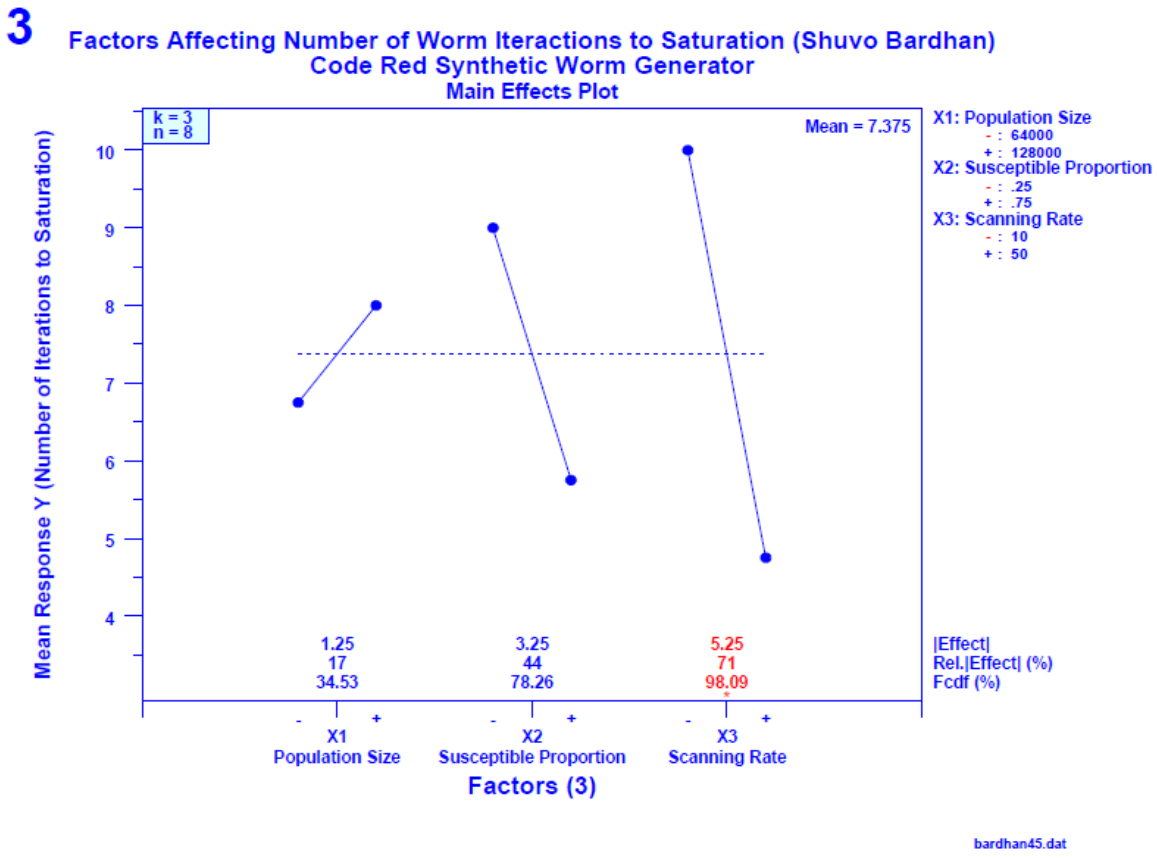

Fig. 9. Main Effects Plot

6.2.1 Choice of Local Model

Mathematically, a large class of functions meet the monotonicity and plateau criteria. The most obvious set of admissible functions is the set of probability-theory cumulative distribution functions (cdf's). Further, since such cdfs exist for a variety of distributions (e.g., uniform, logistic, Cauchy, exponential, lognormal, gamma, Weibull) and distributional shapes (symmetric and skewed), there is a rich set of possible functions that may serve as a good basis for our local modeling problem.

One particular choice is the classic logistic cdf (cumulative distribution function) $L(t)$:

$$y = L(t, \hat{\mu}, \hat{\sigma}) = \frac{1}{1 + e^{\frac{t-\hat{\mu}}{\hat{\sigma}}}} \tag{9}$$

As desired, this function is a dual-plateau, monotonically increasing function; it has vertical limits (0,1).

6.2.2 Local Model Fitting

In fitting this kernel logistic function to each of the 8 available domain-definition data sets for the response $Y(t)$ = Cumulative number of infected hosts through iteration t (see Table 4 and Figure 4), it is clear that the location and scale parameters (μ and σ, respectively) will vary from case to case, but it will be seen that is of no consequence and will be accommodated in a later step. Also, since the classic logistic model ranges vertically from 0 to 1, but the observed data (see Figure 4) ranges vertically from [0,pN] where N = the factor X_1(Population Size setting), and p = the factor $X2$ (Susceptible Proportion setting), then the following modified logistic function was fit for each case of the 8 cases:

$$\begin{aligned} f_i(t) &= f_i\,(t, \hat{\mu}_i, \hat{\sigma}_i, X1_i, X2_i) = f_i(t, \hat{\mu}_i, \hat{\sigma}_i, N_i, p_i) \\ &= 1 + (N_i p_i - 1) \times \frac{1}{1 + e^{\frac{-(t-\hat{\mu}_i)}{\hat{\sigma}_i}}} \qquad i = 1, 2, \ldots, 8. \end{aligned} \tag{10}$$

where $N_i = X1$ and $p_i = X2$ are fixed for each run out of the 8.

This 8 logistic fits are shown in Figure 10. In Figure 10, the circles are the raw data; the solid lines are the predicted values from the least square fit of the logistic models. Note the uniform excellence of the 8 logistic fits as seen visually from Figure 10 and as quantified (see plot legend) by the small Residual Standard Deviations. The Residual SD is formally defined as–

$$s_{res} = \sqrt{\frac{\sum(deviations\ of\ raw\ and\ model\ -\ based\ predicted\ values)^2}{n-p}} \tag{11}$$

where n = the number of observations (here = 8), and where p = the number of fitted parameters per fit (here = 2; μ_i and σ_i).

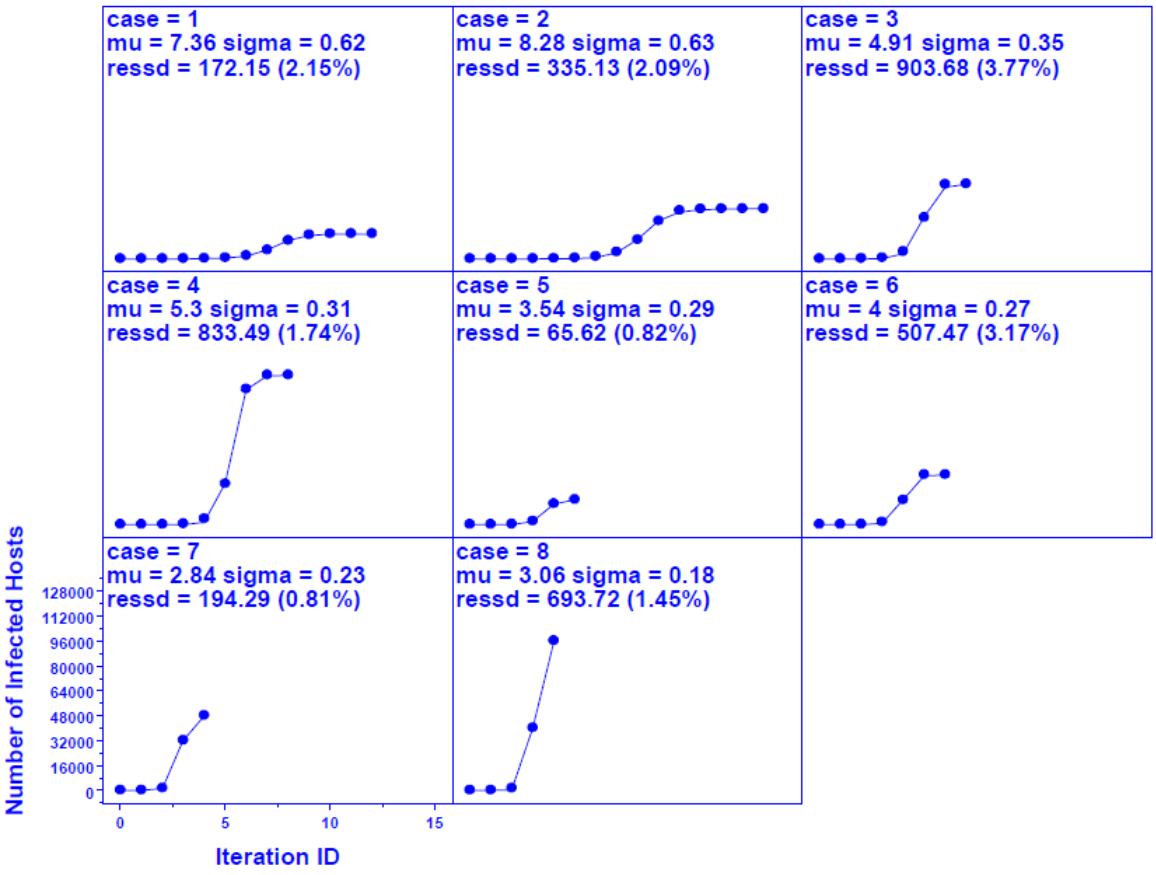

Fig. 10. Data and fitted logistic model for $Y(t)$ = Cumulative number of infected hosts for each of the eight ($X1$,$X2$,$X3$) cases

A perfect fit in which (for all the iterations) the logistic predicted values equaled the observed data would yield $s_{res} = 0$. In all 8 cases, s_{res} is relatively small. For example, s_{res} for Case 1 = 172.65 which (when compared to the mean response) yields a relative s_{res} of (only) 2.16% via the formula–

$$rel(s_{res}) = 100 \times \frac{residual\ standard\ deviation}{mean\ response} \tag{12}$$

In none of the 8 cases does the relative s_{res} exceed 4%.

Finally, the logistic fit for each case yields its own best fit values for location and scale parameters μ_i and σ_i. The s_{res}, μ_i and σ_i values are all given in the legend within each plot. All of the quantitative information for these 8 local fits can be summarized in Table 5.

Table 5. Logistic Location and Scale estimates for the 8 data sets

Run Id	X1	X2	X3	N	p	r	$\hat{\mu}_i$	$\hat{\sigma}_i$	ResSD	Rel(resSD)%
1	*-1*	*-1*	*-1*	64000	0.25	10	7.3639	.6225	172.1523	2.15
2	*+1*	*-1*	*-1*	128000	0.25	10	8.2761	.6279	335.1314	2.09
3	*-1*	*+1*	*-1*	64000	0.75	10	4.9136	.3528	903.6795	3.77
4	*+1*	*+1*	*-1*	128000	0.75	10	5.3048	.3142	833.4942	1.74
5	*-1*	*-1*	*+1*	64000	0.25	50	3.5421	.2863	65.6193	0.82
6	*+1*	*-1*	*+1*	128000	0.25	50	4.0032	.2749	507.4677	3.17
7	*-1*	*+1*	*+1*	64000	0.75	50	2.8373	.2332	194.2945	0.81
8	*+1*	*+1*	*+1*	128000	0.75	50	3.0605	.1825	693.7248	1.45

The first column is the design Run Id (1 to 8); columns 2,3 and 4 are the coded factor settings (-1 and +1); columns 5,6 and 7 are the uncoded (=original) factor settings; columns 8 and 9 are the least square estimates for μ_i(location) and σ_i(scale) respectively; columns 10 and 11 are the residual standard deviations and the relative residual standard deviations.

6.2.3 Local Model Validation

We note that from Table 5 that the logistic model provides an excellent fit of the response Y (= cumulative number of infected hosts at iteration t) for each (and every) one of the 8 domain-definition design cases, and for each iteration within each case. Considering that the response ranges from (0 to 64000) and (0 to 128000), a residual standard deviation s_{res} value less than 1000 would be generally deemed as excellent. In our case, from column 10 of Table 5, it is seen that of the residual standard deviations s_{res} are all less than 1000. Further, the 8 relative residual standard deviation $rel(s_{res})$ are all seen to be universally small–each being less than 4%, and 4 out of 8 cases being less than 2%. We conclude, therefore, that each of the 8 localized fitted models–for 8 fixed (Population Size, Susceptible Proportion, Scanning Rate) combinations–predicts the number of infected hosts with a relatively high degree of accuracy.

6.3 Global Modeling

6.3.1 Choice of Global Model

Global Model for $Y(t)$: Our choice for a global model will be an extension of the choice for the local model. Since each of the 8 local data traces are dual-plateau and monotonically increasing, and since the logistic model $L(t,\mu,\sigma)$ in (10 and 11) serves as a good fit for each with only the location and scale parameter estimates differing from one data-domain ($X1$,$X2$,$X3$) condition to the next, then our choice for the global model will be this same general logistic model $L(t,\mu,\sigma)$ but with derived sub-models functionally relating μ and σ to each of the 3 data domain variables. In short, our global model from (10) for the response $Y(t)$ = the cumulative number of infected host is thus:

$$\begin{aligned} Y(t) &= f(t,X1,X2,X3) \\ &= 1+(Np-1)L(t,\mu,\sigma) \\ &= 1+(X1X2-1)L(t,\mu,\sigma) \\ &= 1+(X1X2-1)L(t,\mu(X1,X2,X3),\sigma(X1,X2,X3)) \end{aligned} \tag{13}$$

Global Model for Y': From this global model for the primary response trace $Y(t) = f(t,X1,X2,X3)$) we could also infer a global model for the second response (Y' = number of iterations-to-saturation). Such an approach for Y' would be both valid and reasonable. Alternatively, a simpler global model for Y' may be obtained by starting with eight instances of data (12,15,8,9,6,7,5,5–as given in the last column of Table 4 and the 2^3 cube plot of Figure 8) and then using a direct model consisting of

$$\begin{aligned} Y'(X1,X2,X3) = b_0 + 0.5[&b_1X1 + b_2X2 + b_3X3 + \\ &b_{12}X1X2 + b_{13}X1X3 + b_{23}X2X3 + \\ &b_{123}X1X2X3] \end{aligned} \tag{14}$$

This flexible 8-parameter empirical model has the important property that it will in fact fit any observed 8 (= 2^3) cube data points exactly, and so a perfect-fit (zero-error) model will always result. In particular, it is seen directly from the bottom of Figure 8 that the fitted global model for Y' = number of iterations-to-saturation is

$$\begin{aligned} Y'(X1,X2,X3) = 7.5 + 0.5[&(1)X1 + (-3.5)X2 + (-5.5)X3 + \\ &(-0.5)X1X2 + (-0.5)X1X3 + (2)X2X3 + \\ &(0)X1X2X3] \end{aligned} \tag{15}$$

This Y' perfect-fit model at the 8 data points is an excellent starting point. The fitted model reaffirms the relative importance of the $X3$ (= Scan Rate) Factor (with |effect| = 5.5), and the $X2$ (= Susceptible Proportion) Factor (with |effect| = 3.5), and adds previously-unknown information about a modest $X2X3$ (= Susceptible Proportion $\times$ Scan Rate) interaction (with |effect| = 2).

6.3.2 Global Model Fitting

Global Model for Y(t): For the initial response $Y(t)$ = cumulative number of infected hosts (at time t), we are thus left with the task of fitting the global model as presented in equation (13). This seeks to synthesize and expand the local models from being not just a function of the iteration t, but also a function of the 3 individual factors: Population Size, Susceptible Proportion and Scanning Rate. Our proposed methodology is to analytically consider the 8 location estimates and the 8 scale estimates from the 8 local fits for $Y(t)$ in the same fashion that we just considered the 8 iterations-to-saturation data for Y', namely, as 8 values on a 2^3 cube, and then fit these 8 values via a 2^3 perfect-fit empirical model. Specifically, we will model the location and scale estimates $\hat{\mu}$ and $\hat{\sigma}$ as functions of the domain-definition factors $X1$,$X2$ and $X3$:

- $\hat{\mu} = g_1(X1, X2, X3) = g_1$ (Population Size, Susceptible Proportion, Scanning Rate)
- $\hat{\sigma} = g_2(X1, X2, X3) = g_2$ (Population Size, Susceptible Proportion, Scanning Rate)

and then compute least squares estimates for needed parameters in these 2 models.

To this end we first form a 2^3 factorial design cube representation of $\hat{\mu}$ and $\hat{\sigma}$ (= column 8 and 9 of Table 5) as a function of the three underlying factors. This is shown in Figure 11 for $\hat{\mu}$:

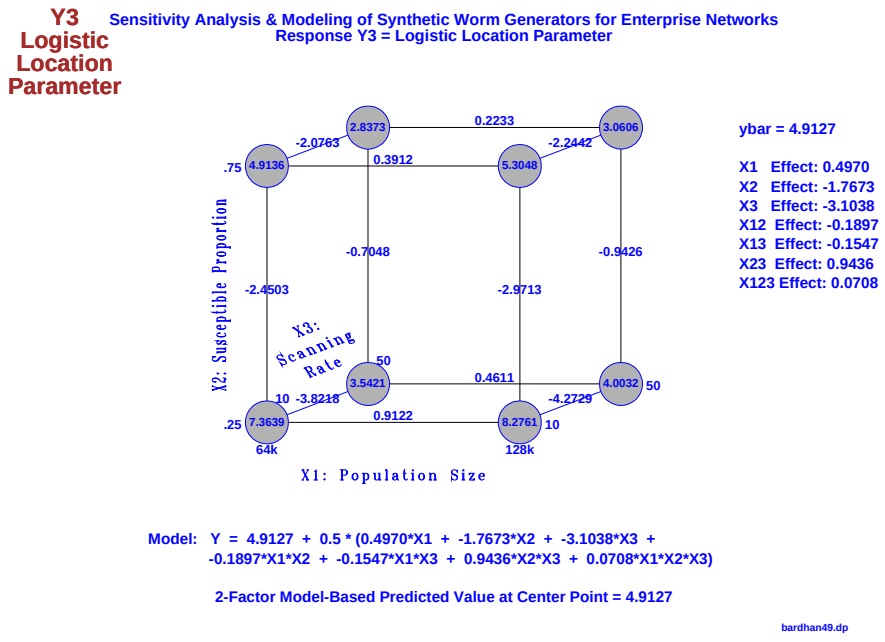

Fig. 11. The least square location estimates $\hat{\mu}$ as a function of the Population Size, Susceptible Proportion and Scanning Rate

From classic 2^3 full factorial sensitivity analysis methodology, we carry out least squares estimation of factor and interaction effects and form the ranked list of factors affecting the values of $\hat{\mu}$:

Table 6. Ranked list of factors affecting location estimate $\hat{\mu}$

Factor or Interaction	Effect Estimate
$X3$	-3.10375
$X2$	-1.76720
$X23$	0.94350
$X1$	0.49695
$X12$	-0.18970
$X13$	-0.15475
$X123$	0.07080

Note the domination by factors $X3$ (Scanning Rate) and $X2$ (Susceptible Proportion) and the relative unimportance of factor $X1$ (Population Size). This ranking of $X3$ followed by $X2$ is not unexpected since it is identical to that seen in Main Effects Plot (Figure 9) for the response Y' = Number of iterations-to-saturation.

More importantly, however, our approach again draws on the previously-discussed fact that for the modeling of 2^3 full factorial designs, an empirical additive model exists involving the mean, the 3 main effects, the 3 2-term interactions, and the 1 estimated 3-term interaction, and this model has the remarkable quality that it provides an exact fit to any 8 data points are. Thus (from the bottom of Figure 11) the perfect-fit empirical model for $\hat{\mu}$ = g1 ($X1$,$X2$,$X3$) for the 8 $\hat{\mu}$ values at hand is

$$\begin{aligned}\hat{\mu} &= g1(X1,X2,X3)\\ &= 4.912575 + 0.5[(0.49695)X1 \ + \ (-1.7672)X2 \ +\\ &\quad (-3.10375)X3 \ + \ (-0.1897)X1X2 \ + \ (-0.15475)X1X3 \ +\\ &\quad (0.9435)X2X3 \ + \ (0.0708)X1X2X3]\end{aligned} \tag{16}$$

The effect estimates of Table 6 of course carry into the prediction equation itself as coefficients. This fitted model perfectly predicts the 8 nodal $\hat{\mu}_i$ values with the net effect that no error is added to the already-seen (Table 6) small relative errors (less than 4%) for each of the 8 local models.

In a similar fashion, we repeat the above procedure for the 8 $\hat{\sigma}$ values presented in column 9 of Table 6. The cube plot is shown in Figure 12. Note that the cube plot (happens) to yield effect estimates $\hat{\sigma}$ (Table 7) with the same ordering as the $\hat{\mu}$ values of Table 6 and the Y' (= number of iterations-to-saturation) values of Figure 8. The desired prediction equation for $\hat{\sigma}$ is shown in equation 17.

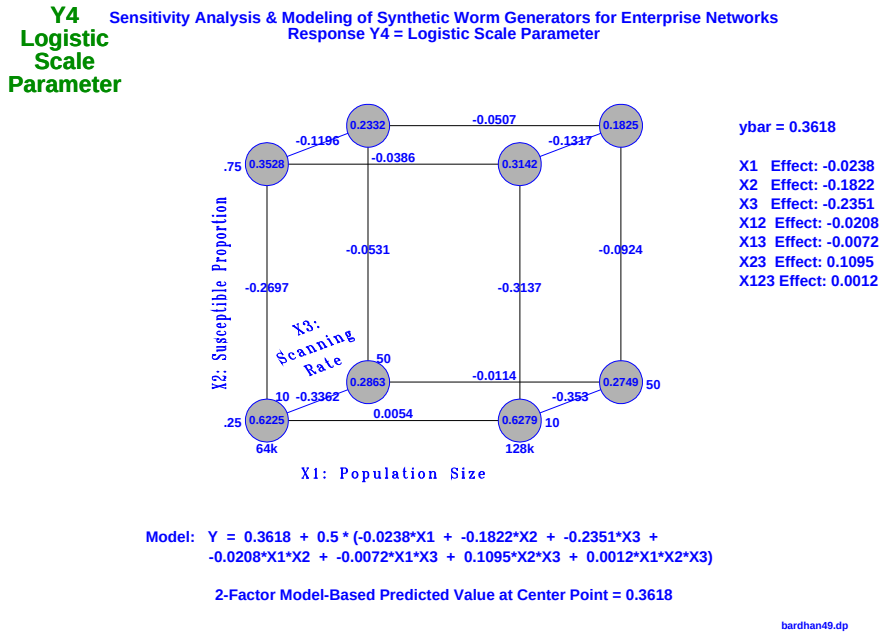

Fig. 12. The least square location estimates $\hat{\sigma}$ as a function of the Population Size, Susceptible Proportion and Scanning Rate

$$\begin{aligned}\hat{\sigma} &= g2\,(X1, X2, X3) \\ &= 0.3617875 + 0.5\,[(-0.023825)X1 + (-0.182225)X2 + \\ &\quad (-0.235125)X3 + (-0.020825)X1X2 + (-0.007225)X1X3 + \\ &\quad (0.109475)X2X3 + (0.001175)X1X2X3]\end{aligned} \tag{17}$$

Table 7. Ranked list of factors affecting scale estimate $\hat{\sigma}$

Factor or Interaction	Effect Estimate
$X3$	-0.23513
$X2$	-0.18223
$X23$	0.10948
$X1$	-0.02383
$X12$	-0.02083
$X13$	-0.00723
$X123$	0.00118

Synthesizing all in the previous, our final general prediction equation relating $Y(t)$ = number of infected hosts to (iteration t , Population Size $X1$, Susceptible Proportion $X2$, Scanning Rate $X3$) becomes

$$\begin{aligned} Y(t) = f(t,X1,X2,X3) &= (Np-1)\left(\frac{1}{1+e^{\frac{t-\hat{\mu}}{\hat{\sigma}}}}\right) \\ &= (X1X2--1)\; L(t,\hat{\mu},\hat{\sigma}) \\ &= (X1X2--1)\left(\frac{1}{1+e^{\frac{t-\hat{\mu}}{\hat{\sigma}}}}\right) \\ &= (X1X2-1)\left(\frac{1}{1+e^{\frac{t-g1(X1,X2,X3)}{g2(X1,X2,X3)}}}\right) \end{aligned} \tag{18}$$

where $L(t,\hat{\mu},\hat{\sigma})$ is given by (9), $\hat{\mu} = g_1(X1,X2,X3)$ is given by (16) and $\hat{\sigma} = g_2(X1,X2,X3)$ is given by (17).

Since the $\hat{\mu}$ and $\hat{\sigma}$ sub-models are perfect-fits and hence add nothing to the error, then the total prediction error across the observed iteration values t and the 8 domain definition design points ($X1$,$X2$,$X3$) of this global model is identical to that of Table 5, namely less than 4% for all 8 data cases.

6.3.3 Global Model Validation

We now address the questions as to how well the models predict at test points other than the 8 2^3 cube "training points" utilized in the model construction. There are two test scenarios:

1. Extrapolatary, in which a test point is chosen outside the 2^3 training cube conditions;
2. Interpolatory, in which a test point is chosen within the 2^3 training cube.

Extrapolation: Extrapolation in general is always a more challenging test problem and has varying degrees of success–depending on the problem and the model. Providing bounds on extrapolation error is beyond the scope of the paper (and will not be considered further).

Interpolation: Contrary to extrapolation, one would expect a model to (at least) do well for internal test points. By nature of the fitting process and the model perfect-fit property at the nodal points, we would expect test points in the immediate vicinity of the 8 nodal points to predict extremely well. In that light, it may be argued that the most challenging internal test point is the value maximally far away from all of the 8 cube points, namely the center point. In the original, 2^3 design, the real values $X1$: (64000 and 128000), $X2$: (.25 and .75), and $X3$: (10 and 50) were coded each as -1 and +1. The center point will thus be ($X1$,$X2$,$X3$) = (96000,.50,30) and will be coded as (0,0,0).

It is good experiment design practice (to assist in testing statistical significance) to include replication somewhere in the design, and so we shall incorporate them at the center

point. To provide tighter critical values, we recommend a minimum of $n_i = 3$ replicates. Such replicates were run and values for $Y(t)$ = number of infected hosts at at iteration t were generated. The time t was extended for as long as necessary until saturation was achieved. Our expanded version of the data in Table 4 thus becomes Table 8 below:

Table 8. Expanded Design and Data ($k = 3$ factors, $n = 8 + 3$ runs)

Run #	Factors			Response Y	Response Y'
	X1 *Pop. Size*	**X2** *Susc. Prop.*	**X3** *Scan. Rate*	(*Cumulative Number of Infected Hosts in each iteration*)	(*Minimum number of iterations-to-saturation*)
1	-1(64K)	-1(.25)	-1(10)	1,3,44,158,1890,5556,11684,15279,15935, 15997,16000	12
2	+1(128K)	-1(.25)	-1(10)	1,3,10,32,110,375,1306,4264,12107,24240, 30842,31880,31988,31999,32000.	15
3	-1(64K)	+1(.75)	-1(10)	1,7,64,554,4468,26323,47667,48000	8
4	+1(128K)	+1(.75)	-1(10)	1,6,50,427,3571,26065,86856,95987,96000	9
5	-1(64K)	-1(.25)	+1(50)	1,12,162,2093,13306,16000	6
6	+1(128K)	-1(.75)	+1(50)	1,10,120,1610,15749,31956,32000	7
7	-1(64K)	+1(.25)	+1(50)	1,36,1374,32031,48000	5
8	+1(128K)	+1(.25)	+1(50)	1,36,1353,40084,96000	5
9	0(96K)	0(.50)	0(30)	1,16,246,3792,34501,48000	6
10	0(96K)	0(0.5)	0(30)	1,15,249,3769,34407,48000	6
11	0(96K)	0(0.5)	0(30)	1,16,235,3678,34004,48000	6

For cross-validation purposes, for the response $Y(t)$ = number of infected hosts, we shall utilize our global equation to generate a predicted value curve at the center point. Similarly, for the response Y' = number of iterations-to-saturation, we shall utilize our global equation (18) to generate a predicted value at the center point. Comparing the predicted curve with the triplicated center point data traces (and the predicted iterations number to the observed saturation numbers) allows us to qualify the quality of the fit at this high-leverage interpolatory test point.

Case 1: Center-point Prediction for $Y(t)$ = Cumulative Number of Infected Hosts Applying the node-based prediction equation at the center point (0,0,0) and making use of the fact that $\hat{\mu}(0,0,0) = 4.91270$ is simply the mean value (= 4.91270) in (16), and $\hat{\sigma}$(0,0,0) = 0.361787 is the mean value in (17), the center point equation for $Y(t)$ thus simplifies to:

$$\begin{aligned} Y(t) = f(t,X1,X2,X3) = f(t,0,0,0) &= 1 + (Np-1)L(t,\hat{\mu},\hat{\sigma}) \\ &= 1 + (X1X2-1)\left\{\frac{1}{1+e^{\frac{-(t-\hat{\mu})}{\hat{\sigma}}}}\right\} \\ &= 1 + ((96000)(0.5)-1)\left\{\frac{1}{1+e^{\frac{-(t-4.91270)}{0.361787}}}\right\} \end{aligned} \tag{19}$$

To determine the prediction error at the center point, let us examine in detail one of the three center-point rows (9, 10 and 11) of Table 9. We shall here use row 9 of Table 8.

Figure 13 graphically presents the data for row 9, and Table 9 presents the error calculations.

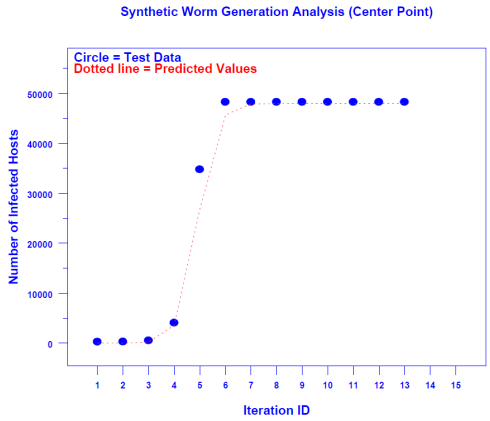

Fig. 13. Center-Point–Test Data vs. Predicted Values

Table 9. Global-Model center-point predicted values and residuals for $Y(t)$

Iteration t	**Test Data** Y	**Predicted Values** Y_{pred}	**Residuals** $Y - Y_{pred}$	**Relative Residuals** (in %)
1	1	1.9646	-0.9646	-96.460
2	16	16.2984	-0.2984	-1.865
3	246	242.5504	+3.4496	+1.402
4	3792	3566.2831	+225.7169	+5.952
5	34501	26882.0975	+7618.9025	+22.083
6	48000	45735.1991	+2264.8009	+4.719
7	48000	47850.6337	+149.3663	+0.311
8	48000	47990.5571	+9.4429	+0.197
9	48000	47999.4047	+0.5953	+0.001
10	48000	47999.9625	+0.0375	+0.000
11	48000	47999.9976	+0.0024	+0.000
12	48000	47999.9999	+0.0001	+0.000
13	48000	48000.0000	0.0000	0.000

This global model for $Y(t)$ does well for the early iterations: 1.9646 vs. 1 for $t = 1$ iteration, 16.2984 vs. 16 for $t = 2$, and 242.5504 vs. 242 for $t = 3$. It does poorest in transition: 3566.2831 vs. 3792 (= 6.0% error) at $t = 4$ iterations, 26882.0975 vs. 34501 (= 22.1% error) at $t = 5$. The model does well in approaching saturation: 45735.1991 vs. 48000 (= 4.7% error) at $t = 6$ iterations, 47850.6337 vs. 48000 (= 0.3% error) at $t = 7$, and error $\leq 0.2\%$ for $t \geq 8$, with final exact convergence at $t = 13$. Based on the observed 6

iterations-to-saturation, the $Y(t)$ fit as a whole has a relative residual standard deviation of 9.66%.

Regarding Y', the global model's predicted number of infected hosts tends to underestimate the observed number of infected hosts, and so the model's predicted number of iterations-to-saturation = will necessarily be longer than the observed value of 6 iterations-to-saturation. In fact the global model yields an exact 48000 (accurate to 4 decimal places) number of iterations-to-saturation at 13 iterations, but more realistically 10 iterations suffice (rounded to the closest integer of 48000), and 7 iterations is in practice adequate with an error of about 150 hosts relative to 48000 (= 0.3% error).

In summary although the global model's $Y(t)$ prediction at the center point is poorer (9.66%) than at the 8 nodal points (0.81% to 3.77%), this residual standard deviation error rate is still quite good, and the generality and extensibility of the approach is a statistical virtue in that it provides accurate predictions of the number of infected hosts across a much broader range of worm-infection scenarios. Further, the purpose of the manuscript was to introduce and describe a general methodology that addresses and opens modeling that would not otherwise be available. Having achieved that, we note that minor modifications of the proposed method can (and will) significantly improve the prediction accuracy for interpolation.

In particular, for simplicity we have described the entire methodology using the data (= number of infected hosts) in its raw form. To address this interpolation issue of poorer center-point prediction accuracy, we point out that one may profitably redo the entire analysis in transformed units. To be precise (though beyond the scope of this paper) it may be shown that repeating the entire analysis in optimaly-transformed units, such as square-roots or logs or inverses. will have the desired effect of eliminating the large $X2X3$ interaction, and thus make the response surface planar as opposed to hyperbolic. Such planarity would certainly have the net effect of reducing the center-point predictive error of 9.77%, which results from the un-transformed raw units analysis. The bottom line is that the observed poorer performance is due not to the general methodology being proposed, but due to the units in which the data analysis is being carried out. Transformed or not, the methodology remains the same. For brevity, we shall illustrate below the benefits of transforming the data by making use of the simpler secondary response Y' = number of iterations-to-saturation.

Case 2: Center-point Prediction for Y' = Number of iterations-to-saturation As discussed at the end of section 6.3.1, to assess center point prediction for the derived model: Y' = number of iterations-to-saturation, a more direct alternate approach may be used than relying on the explicit $Y(t)$ = number of infected hosts model. For Y', the observed number of iterations-to-saturation may be recorded as a response unto itself and analyzed unto itself. This was done already in Figure 8 and a derived direct, non-logistic, exact-fit, empirical model may be utilized based on classic analysis of 2^3 factorial experiments. Such a model was derived at the end of section 6.3.1 and is given by equation 15.

Evaluating equation (15) at the center point (0,0,0) causes all terms to vanish and yields the center point iterations-to-saturation as simply the mean value, namely,

$$Y'(X1,X2,X3) = Y'(0,0,0) = 7.5 \tag{20}$$

Hence based on observed number of iterations to saturation in the training set's 8 nodal points, the predicted number of iterations-to-saturation is 7.5. The true value (from the rightmost column of the last row of Table 8) is 6. The error of prediction is thus $7.5 - 6 = 1.5$ iterations. Though close, this is larger than desirable.

The cause of the prediction error is readily seen in Figure 14, which is a contour plot of the two most important factors $X2$ and $X3$.

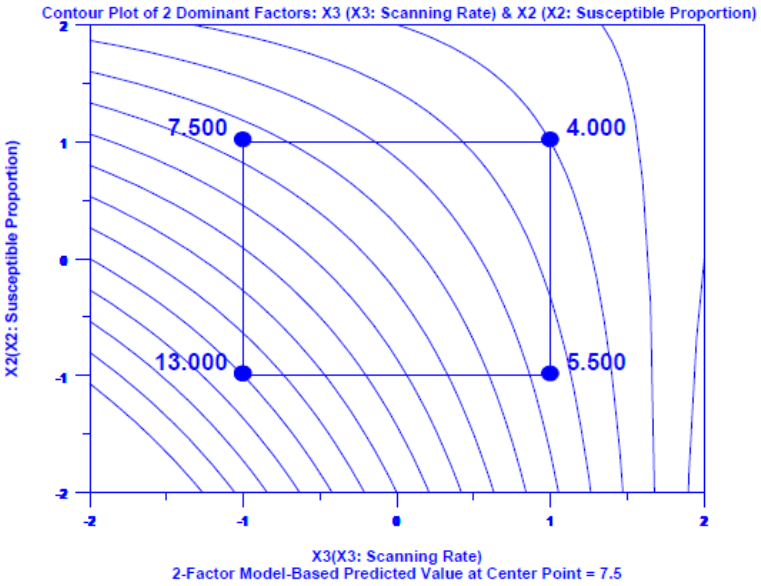

Fig. 14. Raw Data Y': Contour plot of two most important factors: $X3$ and $X2$. Note the curvature–thus indicating the existence of an X2X3 interaction term, which in turn yields more complicated (and less-precise) center-point predictions

Note the pronounced curvature of the contour line, which is a manifestation of an $X2X3$ interaction. Note also that these contour lines were derived from an analysis of the four corner points. With such a strong interaction, it is near-impossible to reliably infer the center-point (or other interpolated nearby values) predictions based on corner-point data/behavior.

On the other hand, with the corrective action recommended above for the $Y(t)$ global model, this 1.5-unit discrepancy may (even with the same general approach) be reduced by an appropriate transformation of the data. Though again beyond the scope of this paper, it may be shown that for this Y' data, a re-execution of the recommended methodology with

the transformed data:

$$T(Y') = \frac{1}{\sqrt{Y'}} \tag{21}$$

yields a center point prediction of 6.57, while the transformation.

$$T(Y') = \frac{1}{Y} \tag{22}$$

yields an even better center point prediction of 6.2 iterations (which rounds to 6 iterations)–agreeing with the 6 iterations-to-saturation observed in the test data.

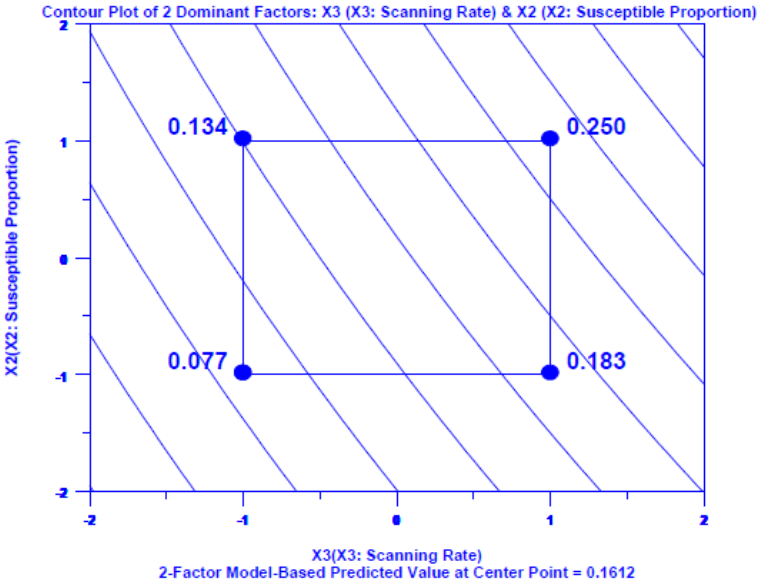

Fig. 15. Transformed Data $Y' = 1/Y'$: Contour plot of two most important factors: $X3$ and $X2$. Note the relative lack of curvature–thus indicating a reduction in the effect of the $X2X3$ cross product terms, which in turn yields more stable (and more accurate) center point predictions

The net effect of the transformation is to remove/reduce the interaction/cross-product term in the model. The result is that the response surface becomes linear rather than having cross-product-induced curvature. Such linearity typically results in a simpler response surface and more accurate predictions–for both interpolation and extrapolation. To illustrate this effect of the transformation, note Figure 15, which is a contour plot of the same two most important factors, but with transformed units $T(Y') = \frac{1}{Y'}$.

6.4 Global Models with 4+ factors

This section deals with the problem of how to make use of the developed three-factor model methodology and how to extend that to additional factors. In particular, how can the two level cube based methodology be extended to include Table 1's additional factors $X4$, $X5$ and $X6$:

6.4.1 4-Term Global Model for $Y(t)$: Including $X4$ (Number of Initial Infected Hosts)

The global model for $Y(t)$ for three factors, as given in equation 15, is

$$Y(t) = f(t,X1,X2,X3) = 1 + (X1X2–1)L(t,\hat{\mu}(X1,X2,X3),\hat{\sigma}(X1,X2,X3)) \tag{23}$$

where $\hat{\mu}$ is given in equation 16 and $\hat{\sigma}$ is given in equation 17.

This model assumes that $Y(1) = 1$, that is, the number of initial infected hosts is unity. In the event that the number of initial infected hosts is not one but rather is a variable $X4$ unto itself, then the range of the response $Y(t)$ will no longer be [1,Np], but rather [$X4$, Np] = [$X4$, $X1X2$]. Given that, and given the independent two-stage nature of the methodology's local and then global fitting, then the eight local fits $f_i(t)$= $f_i(t,X1,X2,X3,X4)$ will still involve the logistic model, but with slightly altered $X1,X2,X3,X4$.

In particular, the 8 local logistic fits will now have the form–

$$f_i(t,X1,X2,X3,X4) = X4 + (X1X2–X4)L(t,\hat{\mu}_i,\hat{\sigma}_i) \tag{24}$$

and the synthesized global fit for $Y(t)$ also has similar form–

$$f(t,X1,X2,X3,X4) = X4 + (X1X2–X4)L(t,\hat{\mu}(X1,X2,X3),\hat{\sigma}(X1,X2,X3)) \tag{25}$$

where $\hat{\mu}$ and $\hat{\sigma}$ are given in equations 16 and 17, respectively.

6.4.2 6-Term Global Model: Including $X4$, $X5$ and $X6$ (Number of Initial Infected Hosts, Death Rate, and Patching Rate)

The methodology for the six-factor case will be the same as the original three-factor case, but will have two sampling options—one involving $n = 32$ runs, and the other with $n = 16$ runs. Either is admissible; pros and cons are discussed, and the final choice is dictated by the maximum number n of runs affordable based on time and cost constraints.

This $k = 6$ factor general worm propagation model may thus be fit in a highly efficient fashion: "costing" only $n = 16$ runs. If affordable, the ($k = 6$, $n = 32$) 2^5 full factorial design is recommended, but if not affordable, the ($k = 6$, $n = 16$) 2^{5-1} orthogonal fractional factorial design is an excellent and highly-affordable alternate recommendation for the worm-propagation modeling case.

2^5 Full Factorial Design (k = 5, n = 32): Since $X4$ may be simply folded into the three factor model, then the full six factor generalized model will be folded into a five factor model. The default experiment design for the five-factor model is a 2^5 full factorial, and so the global model will emanate from the 2^5 = 32 localized models f_i:

$$f_i(t, X1, X2, X4) = X4 + (X1X2–X4)L(t, \mu_i, \sigma_i) \qquad i = 1, 2, ..., 32 \qquad (26)$$

where each localized logistic model L is a function of five factors: $(X1, X2, X3, X5, X6)$. Note how the term $X4$ appears in a partitioned fashion—up front, but not involved in any of the logistic sub-functions.

For the five factors $(X1, X2, X3, X5, X6)$, the data for the logistic fits would by default come from 2^5 = 32 number-of-infected-hosts data traces collected over the 32 fixed settings from a 2^5 full factorial experiment design. These 32 logistic fits yield high-precision predicted values at each iteration t within the 32 localized data traces. Further, these 32 local logistic fits would yield 32 ($\hat{\mu}_i$, $\hat{\sigma}_i$) pairs of estimated values for the logistic parameter values μ_i and σ_i.

As before, these 32 estimated $\hat{\mu}_i$ and $\hat{\sigma}_i$ values may themselves be envisioned as nodal values on a five-dimensional hypercube, and hence μ and σ are functionally related to $X1$, $X2$, $X3$, $X5$ and $X6$:

$$\begin{aligned} \hat{\mu}_i &= \hat{\mu}_i\ (X1, X2, X3, X5, X6), and \\ \hat{\sigma}_i &= \hat{\sigma}_i\ (X1, X2, X3, X5, X6) \end{aligned} \qquad (27)$$

Many five-factor functions may be utilized to model these relationships, but of particular note is the five-factor 32-term model consisting of

1. a constant
2. 5 main effects, 10 2-term interactions, 10 3-term interactions
3. 5 4-term interactions, and 1 5-term interaction.

This model has the property that the resulting least squares fit of the 32 coefficients of the model match perfectly (zero error) the 32 input values μ_i and σ_i. The resulting ($n = 32$) global model for $Y(t)$ = number of infected hosts thus becomes

$$\begin{aligned} Y(t) &= f(t, X1, X2, X3, X4, X5, X6) \\ &= X4 + (X1X2 - X4)\ L(t, \hat{\mu}_i(X1, X2, X3, X5, X6), \hat{\sigma}_i(X1, X2, X3, X5, X6)) \end{aligned} \qquad (28)$$

This $k = 6$ factor general worm propagation model may thus be fit in a highly efficient fashion: "costing" only $n = 32$ runs. If this is affordable, then this ($k = 6$, $n = 32$) modeling procedure is recommended. It has the property that it has high-precision predicted values at the 32 nodal points.

2^{5-1} Orthogonal Fractional Factorial Design (k = 5, n = 16): If $n = 32$ runs is too expensive, then an efficient lower cost alternative is readily available. In particular, an excellent alternative to the 2^5 full factorial design is a 2^{5-1} orthogonal fractional factorial design which still examines $k = 5$ factors, but requires only $n = 2^{5-1}$ = 16 runs. The canonical design matrix is shown in Table 10.

Table 10. 2^{5-1} Orthogonal Fractional Factorial Design ($k = 5$ factors, $n = 16$ runs)

X1	X2	X3	X4	X5
-1	-1	-1	-1	+1
+1	-1	-1	-1	-1
-1	+1	-1	-1	-1
+1	+1	-1	-1	+1
-1	-1	+1	-1	-1
+1	-1	+1	-1	+1
-1	+1	+1	-1	+1
+1	+1	+1	-1	-1
-1	-1	-1	+1	-1
+1	-1	-1	+1	+1
-1	+1	-1	+1	+1
+1	+1	-1	+1	-1
-1	-1	+1	+1	+1
+1	-1	+1	+1	-1
-1	+1	+1	+1	-1
+1	+1	+1	+1	+1

This 16-run design is balanced (even column has half -1 and half +1) and is orthogonal (every pair of columns has a quarter (-1,-1), a quarter (-1,+1), a quarter (+1,-1), and a quarter(-1,+1)). The design has excellent statistical estimation properties (minimal bias and uncertainty for effect and interaction estimates). Note that the selected 16 points are a judicious subset of a 32-run 2^5 full factor design–this 2^{5-1} design will result in 16 sampled points and hence 16 unsampled points (from the 32-run full factorial).

Note that these 16 runs are only eight runs more expensive than the ($k = 3$, $n = 8$) full factorial design that we originally utilized for the examination of the $k = 3$ factors $X1$, $X2$ and $X3$. Thus for twice the effort ($n = 8$ to 16 runs), we have developed a modeling methodology encompassing twice the number of factors in worm-space: $k = 3$ to 6 factors.

In particular, the above 2^{5-1} experiment design will be utilized and so only 2^{5-1} = 16 localized models f_i will be needed.

$$f_i(t, X1, X2, X4) = X4 + (X1X2 - -X4)\, L(t, \hat{\mu}_i, \hat{\sigma}_i) \qquad i = 1, 2, \ldots, 16. \tag{29}$$

The training data for the logistic fits would come from 16 number-of-infected-hosts data traces collected over 16 fixed settings from the specified 2^{5-1} fractional factorial experiment design. These 16 logistic fits yield high-precision predicted values at each iteration t within the 16 localized data traces. The orthogonal design assures that 16 training points are selected in a balanced and comprehensive fashion over the 5-space, thus these 16 points serve as high leverage data points to compensate for the 16 data points not sampled (relative to the full (= 32 run) factorial design). These 16 points have enough balance to help assure high precision, but also have enough coverage to assure minimal bias.

In addition to the 16 prediction traces that result, the 16 local logistic fits would yield 16 ($\hat{\mu}_i$, $\hat{\sigma}_i$) pairs of estimated values for the logistic parameter values μ_i and σ_i. As before, these estimated $\hat{\mu}_i$ and $\hat{\sigma}_i$ values may themselves be envisioned as nodal values on a 5-dimensional hypercube, and hence are functionally related to $X1, X2, X3, X5$, and $X6$:

$$\begin{aligned} \hat{\mu} &= \hat{\mu}(X1, X2, X3, X5, X6), \textit{and} \\ \hat{\sigma} &= \hat{\sigma}(X1, X2, X3, X5, X6) \end{aligned} \tag{30}$$

To model these μ and σ, functional relationships, the five-factor 32-term model consisting of

- a constant +
- 5 main effects, 10 2-term interactions, 10 3-term interactions +
- 5 4-term interactions, and 1 5-term interaction,

is still powerfully relevant, but with 16 values as input, the fitting task of estimating 32 values (coefficients) as output seems—on the face of it—impossible. In practice, however, the fitting is eminently possible with the imposition of a reasonable assumption namely, that all of the dominant causalty is from main effects and two-term interactions, and the three-term (and higher) interactions are relatively unimportant (i.e., near zero). It is our experience that such an assumption is appropriate in most scientific and engineering applications, with this worm-propagation modeling problem being seen as no different.

The good news is that regardless of the assumption's validity, the predicted values that result from the μ_i and σ_i fitting process will in fact match exactly the fitted data points at the 16 data points, and since such points are balanced in coverage across the five factors, the prediction model should yield excellent prediction values at the remaining 16 unsampled data points from the 2^5 hypercube. In the event that some higher-order interactions are in truth large, then again the 16 sampled points will still have a perfect-fit, but the predicted values at the 16 unsampled points will have higher error (compared to a 32-run 2^5 full-factorial sampling plan).

In summary, this ($k = 5$, $n = 16$) design and modeling procedure has the property that the least squares fit of the 16 coefficients of the empirical models are such that the subsequent predicted values match perfectly (zero-error) the 16 input values μ_i and σ_i. As before with the 32-point case, the resulting n = 16-point global model for the $Y(t)$ = number of infected hosts response is identical in form, namely,

$$
\begin{aligned}
Y(t) &= f(t, X1, X2, X3, X4, X5, X6) \\
&= X4 + (X1X2\text{–}X4)L(t, \hat{\mu}(X1, X2, X3, X5, X6), \hat{\sigma}(X1, X2, X3, X5, X6))
\end{aligned}
\tag{31}
$$

This $k = 6$ factor general worm propagation model may thus be fit in a highly efficient fashion: "costing" only $n = 32$ runs. If affordable, the ($k = 6$, $n = 32$) design is recommended, but if not affordable, the ($k = 6$, $n = 16$) design is an excellent and highly-affordable alternate recommendation for the worm-propagation modeling case.

Refining the Global Model

As with the $k = 3$ case, the global model $Y(t)$ be further improved by

1. Collecting triplicated data at the coded center point (0,0,0,0,0,0) for all 6 factors.
2. Transforming the response (i.e., $T(Y) = 1 / \sqrt{Y}$) and reanalyzing–with the net affect that an interaction term is eliminated, linearity is imposed, and prediction accuracy is improved for the center point in particular and for other interpolation points in general.

Single Versus Multiple Replications

Note that for all three of the two-level experiment-design cases discussed above: the 2^3, the 2^5, and the 2^{5-1}, only a single data trace was generated at each of the (8, 16 and 32) nodal points, respectively. This single sampling was done to simplify the exposition of the proposed two-stage (local, then global) modeling methodology. In practice (and if affordable), then at each training set nodal point, it is good statistical practice to generate multiple (= replicated) infected-hosts traces (at differing seed values) so as to more closely mimic random fluctuations existent in reality. In such case, an additional component of error will be introduced having to do with induced error due to replication. In our worm-modeling case, such induced error is relatively small. In short, some replication would assist in gaining insight into the magnitude of run-to-run variation intrinsic in our process. On the other hand, regardless of the size of replication error, it in no way detracts for the utility of the recommended modeling technique described above. If replication is available, then all of the individual replicated data values would be used in the individual logistic fits, but then an average number of iterations-to-saturation would be computed and used for the exact-fitting process of the estimated μ_i and σ_i.

7. Conclusion

This paper demonstrated an extensible, simulator-based/data-based methodology for developing a high-precision global model for local-scanning computer worms propagating in networks. The methodology involved both experiment design and analysis components. Two responses were considered:

$Y(t)$ = the number of infected hosts and
Y' = the number of interactions to saturation.

The model initially included $k = 3$ factors, which are known to influence worm propagation:

1. $X1$: size of the address space (Population Size),
2. $X2$: size of the susceptible host sub-population (Susceptible Proportion), and
3. $X3$: worm scanning rate (Scanning Rate),

The initial ($k = 3$-factor) model construction was done as follows:

1. Construct a generic algorithm capable of generating simulated worm-infection data (Number of Infected Hosts vs Iteration Id) for a given (Population Size, Susceptible Proportion, Scanning Rate) combination.
2. Simulate experimental (training set) infection data for a sampled, small number (in our case, 8) of representative (Population Size, Susceptible Proportion, Scanning Rate) combinations.
3. Carry out 8 high-precision fits (based on the logistic model)—one fit for each of the 8 factor combinations.
4. Carry out local goodness of fit tests. The 8 fitted local logistic models were all high-precision, with all residual standard deviations $< 4\%$.
5. Note the estimated logistic model location $\hat{\mu}_i$ and the scale parameters $\hat{\sigma}_i$ across the 8 factor combinations.
6. Fit a perfect-fit empirical model to these parameters across the three-factor space.
7. Synthesize the eight local models and the two parametric models into a single high-precision global model.
8. Carry out global model training-set goodness-of-fit tests. Since fitting the parameter values contributed no additional error, then the global model was seen to fit the $Y(t)$ training set with error from 1.5% to 4% over all observed iteration data, and was seen to fit the Y' = iterations-to-saturation perfectly over the eight data-domain nodal points.
9. Carry out global model test-set interpolatory goodness-of-fit tests. For the raw response data, the model prediction was slightly high: 7.5 iterations-to-saturation compared to true value of 6 iterations. An appropriate transformation solved the problem: for the transformed ($1/Y'$) data, the model predicted 6.2 iterations.
10. Extend the model to $k = 6$ factors, via a complete ($n = 32$) 2^5 full factorial design or a more efficient ($n = 16$) 2^{5-1} orthogonal fractional factorial design.

The paper also demonstrated sensitivity analysis as part of the proposed methodology discussion to gain insight into the relative importance of the various factors (and interactions). For the simple $k = 3$ factor case, it was seen that the rank of factor importance was

1. $X3$ (Scanning Rate), followed by
2. $X2$ (Susceptible Proportion) and the
3. $X3X2$ (Scanning Rate)(Susceptible Proportion) interaction.

with factor $X1$ (Population Size) seen to be of lesser importance.

References

[1] Kienzle DM, Elder MC (2003) Recent worms: A survey and trends. *Proceedings of the 2003 ACM Workshop on Rapid Malcode* WORM '03 (ACM, New York, NY, USA), pp 1–10. https://doi.org/10.1145/948187.948189. URL http://doi.acm.org/10.1145/948187.948189

[2] https://www.nist.gov/sites/default/files/documents/itl/bits-malware-report-jun2011.pdf, .

[3] Zargar ST, Joshi J, Tipper D (2013) A survey of defense mechanisms against distributed denial of service (ddos) flooding attacks. *IEEE Communications Surveys Tutorials* 15(4):2046–2069. https://doi.org/10.1109/SURV.2013.031413.00127

[4] Khonji M, Iraqi Y, Jones A (2013) Phishing detection: A literature survey. *IEEE Communications Surveys Tutorials* 15(4):2091–2121. https://doi.org/10.1109/SURV.2013.032213.00009

[5] Moore D, Shannon C, K Claffy (2002) Code-red: a case study on the spread and victims of an internet worm, . pp 273–284.

[6] Moore D, et al. (2003) Inside the slammer worm. *IEEE Security Privacy* 1(4):33–39. https://doi.org/10.1109/MSECP.2003.1219056

[7] Chen Z, Gao L, Kwiat K (2003) Modeling the spread of active worms. *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No.03CH37428)*, Vol. 3 Vol. 3, pp 1890–1900 vol.3. https://doi.org/10.1109/INFCOM.2003.1209211

[8] https://en.wikipedia.org/wiki/IP_address.

[9] http://www.itl.nist.gov/div898/software/dataplot/refman2/auxillar/samprand.htm.

[10] https://www.itl.nist.gov/div898/handbook/eda/section3/eda35i.htm