Distributed Detection of New Virus Threats in Large Scale Networks

Joshua M. Hailpern
Carnegie Mellon University
School of Computer Science
www.hailpern.com/joshua
Joshua@hailpern.com

Advisor: Professor Benoit Morel
Carnegie Mellon University
Engineering and Public Policy
http://www.epp.cmu.edu/httpdocs/people/bios/morel.html
Bm1v@andrew.cmu.edu

ABSTRACT
The goal of this research is to explore the possibility of extending ideas proposed by von Neumann in the paper *Probabilistic Logics and the Synthesis of Reliable Organisms from Unreliable Components,* to build a very large scale intrusion detector system able to detect new cyber-attacks with higher reliability than the "sum" of its components. The proposal is to build an information processor made of many components networked in such a way that the probability of false positive and false negative is smaller than its individual components.

The first phase of the work consisted of familiarization with the details of the original paper. Von Neumann describes a system of "organs" in a nervous system. These "organs" (not like a heart or lungs, but rather a term to describe a small unit) have some chance of misfiring, *epsilon*. As a result, his paper discusses a method of merging this data together so as to reduce the effect of "faulty" data. In addition, his paper addresses the issue of different messages being detected by different sensors. Von Neumann provides a method for combining this information by using properties of large numbers to find the "true state" of the system.

We are investigating whether or not Von Neumann's nervous system design can be applied to anti-virus detection. Unlike the nervous network proposed by Von Neumann to transmit a single, binary signal, the proposed virus detection network must make affordances for other critical pieces of data; multiple viruses/different signatures, time discrepancy, and virus spread. In our paper, we investigate possible solutions to these aspects of the application of Von Neumann's work to that of a virus detection network.

The most recent phase of the work consisted of an in-depth study of the world of anomalies (the main mechanism for that kind of detection), and in particular system calls. We wished to understand how detectors using system calls can exchange information in such a way that their aggregated probability of false positive and false negative is much smaller than their individual probabilities of false positive and false negative.

Author Keywords
Virus, Anti-Virus, Worms, Anomaly, Security

ACM Classification Keywords
D4.6. Invasive Software

INTRODUCTION
As computers become pervasive in world society, the potential financial damage from a large viral or worm attack could be in the billions of dollars. To combat such attacks, many systems of anti-viral software have been developed. However, most of these systems are reactionary. They require a human to be "in the loop" to help identify new threats. Though this approach has proved effective, this reactionary model is slow, and usually cannot prevent threats in zero time. This paper proposes the conceptual background for a new virus detection system, whose basis lies in the work by Jon von Neumann.

In his paper *Probabilistic Logics and the Synthesis of Reliable Organisms from Unreliable Components*, von Neumann proposes a model of the human nervous system based on "organs" (a term used to describe a small unit, rather than a heart or lungs) which have an *epsilon* (small) chance of misfiring. Relying on the law of large numbers, his network of organs reduce the chance of overall "false positive." We propose a model of a computer network, based on von Neumann's that follows many of the same principles to dynamically detect new virus and worm threats in approximately zero time based on anomalous activity.

We will begin by discussing more details of von Neumann's paper, including our own exploration into his mathematical analysis. Following the review of his literature, we will point out the aspects that do not directly map to the computer model, and our proposed solutions to said issues. These solutions include the creation of new types of "organs", as well as innovations in the overall structure of our computer network. We will then show, through our own mathematical analysis, the strength of our new network to reduce the possibility of false positive and

the increased chance of true positive detection. Finally, we propose some future conceptual modifications to further improve our network.

VON NEUMANN

The basis of the research originates with Jon Von Neumann's paper *Probabilistic Logics and the Synthesis of Reliable Organisms from Unreliable Components.* In this ground-breaking article, Von Neumann proposes a logic gate based model of the human nervous system, which through the properties of large numbers, filters out unsubstantiated signals (false positives). His model uses a series of "organs" to send binary signals throughout the network, and combines these signals together to create a general analysis of the system as a whole. Each node in the network (a total of N nodes), has a chance of malfunction (ε). The goal is to reduce the probability of overall malfunction (some fraction above a fiduciary level Δ).

Before we could start to apply his findings to our model of zero-time virus detection, we analyzed Von Neumann's equations and system. Von Neumann sets forth some base numbers for key values in his system;

- $\varepsilon \leq 0.0107$
- $\Delta = .7$

As a result, we wanted to explore other values of *epsilon* and *delta* to confirm that large numbers do, in fact, reduce the fraction above the fiduciary level.

Appendix 1-6 illustrates Von Neumann's system, with different values for *epsilon*, *delta*, and N. From these results, we concluded that epsilon is the key value to work around in that it is the threshold above which the fiduciary level may not be reached. In other words, the system will fail.

PROBLEMS WITH MAPPING

In intrusion-detection, the overall challenge is to reduce the false positive rate while maintaining the lowest level possible of false negatives. Though Von Neumann's system elegantly deals with erroneous data, providing a connection between that system and the real world is not so easy. During the next step of the research, we brainstormed a list of potential mapping problems between Von Neumann's network and our goal of a virus detection system.

One hurdle to realizing a virus detection system is that the Von Neumann network assumes all information (all signals) occur at the exact same moment in time. Unfortunately, viruses and worms require some amount of time to spread and to cause "abnormal" occurrences on a user's system. Therefore some longitudinal or temporal accommodations must be made. Beyond the difference between a single instant of time and a span of time, there is another difference: a static system versus a dynamic system. A single snapshot in time (or a pause for a system to reach equilibrium) is different from a system that continues to change (as, for example, as a worm expands through a network).

Another problem is that there will potentially be more than one type of virus or worm in a network. Thus a binary detection system (something is wrong, something is not wrong) will find it difficult to identify if a single threat exists (in the presence of multiple infections). Further, a single threat may cause different symptoms on different systems, or the symptoms may evolve over time (for example, a stealth-like growth until some fraction of the network is infected, followed by a vigorous denial of service attack on a predefined target). Thus any method used to compare a system's reactions must take these complications into consideration.

Automatic anomaly detection, in the current state of the practice, tends to have a high rate of false positive. An unusual "event" can occur when a new program is installed or the user does something "un expected." Further, depending on how automatic anomaly detection is set up, simply more "dangerous" commands or sequences could be flagged, like a delete request, or a write request. We can try to reduce that rate by requiring the same detector to notice activity multiple times to avoid transient errors and detect only persistent problems. That approach, however, would produce false negatives when viruses or worms only need one "attack" to do all of their intended damage.

Because this paper is the culmination of only two semesters worth of work, we will be addressing the former of the two mapping problems. As a result, the proposed system will be used to identify a general trend of the network. However, in the future work section, there is a brief discussion of theories towards dealing with different anomalies and different threats.

PROPOSED ARCHITECTURE

Our new system is composed of three levels: Level 1 (L1), Level 2 (L2), and the Von Neumann Network (VNN). The goal of this layout is to reduce the rate of false positives, as well as to manage the asynchronous nature of the network. The first level, L1, is the detector machine. This machine uses an anomaly detection system to determine the possibility of an attack. The second level of the system, L2, handles network asynchrony, crashed systems (possibly due to attacks), and validation of a threat on multiple computers. The third level, the VNN, is the system proposed by Von Neumann in his original paper. See *Diagram 3* for an illustrated layout of the proposed architecture.

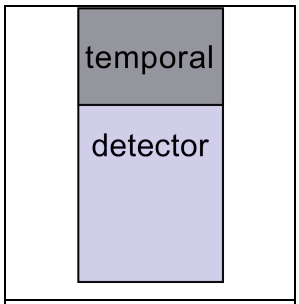


Diagram 1 – L1

L1 can be represented by <N, D, T> where N is the network (graph) of system components above L1 (comprising of L2 and VNN), and the directional links between them, D is the set of individual systems which are used to detect anomalous activity, and T which is the set of temporal gates, each of which provide a link from one element in D to many elements in N. More specifically, a detector gate is connected to a temporal gate, which connects to many gates in L2. Each set of detector and temporal gates can be considered a L1 node. A node at L1 can be illustrated in Diagram 1.

L1 begins with a detector: that is, a component system detects anomalous activity (currently a black box), and alerts the L1 level when the amount of activity reaches or exceeds a predetermined threshold. These alerts are forwarded to a temporal gate. The purpose of this gate is to monitor the alerts that the detector finds. Within an established window of time, if there are n alerts (whether or not similarity is required has yet to be determined) the temporal gates alerts L2 of its findings. This temporal windowing should reduce the chance of transient false positives, because it requires some level of sustained malicious action to occur.

Note that this boundary between L1 and L2 is, in some sense arbitrary. In an alternative configuration, L2 could hold the temporal filter for false positives. That alternative would, however, require an expanded bandwidth between L1 and L2, because L2 would have to monitor all alerts from the L1 layer. In general, a sound software engineering principle is to minimize the information flow between architectural components and levels, so as to provide better information hiding and increased flexibility at implementation time.

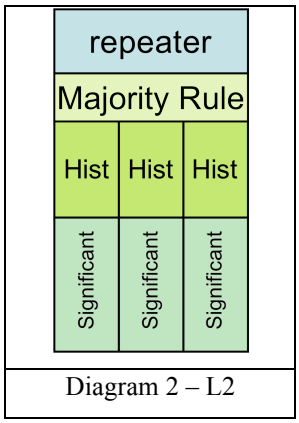


Diagram 2 – L2

L2 can be represented by <G, V, S, H, M, R> where G is the set of inputs coming to L2 from L1. V represents the VNN, S is the set of "significant gates", one gate for every input to each L2 node. H represents the set of the history gates, one for every element in S, M is the set of all majority gates, and R for the set of repeater gates. A L2 node consists of 1 element for S and H for every input to the node. Each S element matches to one H element. The L2 node the has one element from M, which all elements from H in this node connect to. Then the M element connects to an R element, which connects to V. Diagram 2 represents a L2 node with 3 inputs.

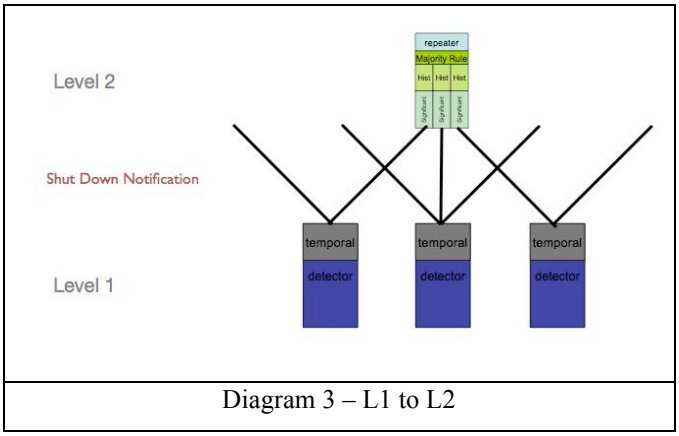


Diagram 3 – L1 to L2

The connection between L1 and L2 is like a web of signal wires, much like the human nervous system. This is illustrated in Diagram 3, with a simple 3 node situation. Every detector in L1 connects to at least two, if not more L2 nodes. This design decision was made because virus and worm do not necessarily spread out linearly through the network. Furthermore, they do not always spread through physical neighbors of infected nodes. Thus, by increasing the number of possible connections, we increase the chance that infected systems will be detected. L2 and L1 systems maintain active "heartbeat" connections, constantly

checking to see if each is still "alive". If an L1 system is shut down, it will send a "shut down" signal to the L2 systems connected to it. However if an L1 crashes, such a signal will not be sent, and as a result, action can be taken to reflect a crashed system, especially if an alert was sent immediately before.

L2 is more complex then L1. It consists of four new types of gates; *repeater* gate, *significant* gate, *history* gate, and *majority rule* gate. The overall goal of L2 is to decide when there is enough evidence to report (to the next level) that an event has occurred. When a signal enters L2, it first enters through a significant gate, which monitors for a steady level of positives, which would suggest something of significance may have occurred. Once an "event" is determined to exist by the significant gate, a signal is passed to the associated history gate. The history gate processes events related to the asynchronous nature of the network by repeating the "event" status to the next gate in the L2 node. Following the significant gate and history gate series, each input into L2 converges into a majority rule gate. The majority rule gate takes in N inputs. If $\lceil N/2 \rceil$ inputs have an alert in common, then the majority rule gates passes on an alert. Even if some systems go down, the majority rule gate can still function as long as it still has viable sources of input. Once the majority rule gate determines that there is a common alert across some portion of the network, it forwards the "event" onto a repeater gate. The repeater is used to deal with asynchrony in the level above, the VNN. Because all signals in the network will not occur at the same time, nor will virus spread occur at a simultaneous rate, the repeater gate will continue to broadcast an alert to the VNN for a predetermined length of time, hoping to increase the chance of an alert collision (in time) in the VNN

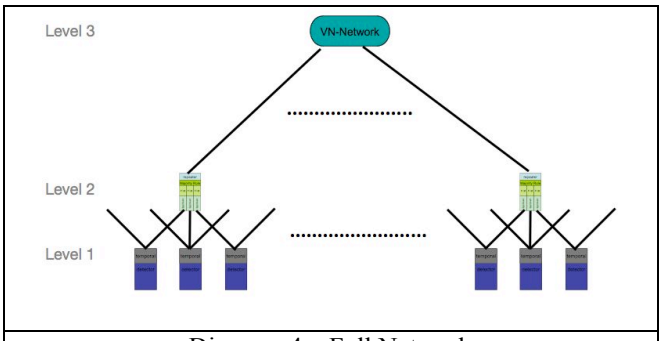


Diagram 4 – Full Network

Finally, the signal reaches the third level, the VNN. Diagram 4 illustrates this. By using only binary signals, the input/notification of event to the VNN will work exactly like the network proposed in Von Neumann's paper. As a result, if enough systems in the network detect an "event", then the due to the properties of large numbers, the VNN will notify the central "brain" that there appears to be malicious code spreading throughout the network as a whole.

This architecture, as it stands, assumes that there exists one standard uniform system architecture on each of the nodes in the network. Thus there are no "OS specific" attacks. In a more realistic model, there could be multiple parallel detector "levels" each running concurrently, linking systems of similar design together, so as to create a more accurate picture of the state of the network.

In addition, it has been hypothesized that a modification to the VN could be made to simply the overall network. Currently in the VNN, there exists an "organ" called the Sheffer Stroke. The goal of this organ is to collect all the (multiplexed) incoming data (inputs) and react similarly to a NAND gate. The inputs are put in pairs. When two input links are activated simultaneously there is no output, the rest of the time the output link is activated. In VN networks, many Sheffer strokes work in parallel and what matters is the number of output links non-activated as compared to the number of input links activated.

It is theorized, that the work achieved by L1 and L2 as proposed not only performs the same function as the Shafer Stroke, but does the task in a more practical way in relation to this model. Yet, having another step to reduce false positives could be useful to reduce the effectiveness of the new network as a whole.

MATHMATICAL JUSTIFICATION

Ostensibly the paper of John von Neumann, addresses the question of how to reduce the error due to unreliable components to an arbitrary small level using multiplexing and large numbers. In practice, the ideas developed in that paper have the potential to be applied to a large variety of problems involving unreliable components and we think among others the problem of early detection of new malware. Here we described succinctly the major observation of von Neumann.

Von Neumann discussion deals with two kinds of gates: the majority rule and the Sheffer stroke (NAND gate). A majority rule gate receives information from three sources. The probability that the gate yields a false information is the probability that at least two of the three sources were providing a false information. If χ_i is the probability that line "i" gives a false negative, the probability that at least two of the three incoming lines give a wrong information and that the gate is sending a false negative signal is show in Equation 1.

$$\pi_g = \chi_1\chi_2 + \chi_1\chi_3 + \chi_2\chi_3 - 2\chi_1\chi_2\chi_3$$

Equation 1

The result is shown in Figure 1. If one can assume that in average $\chi_i \approx 10\%$, then the probability of false positive of the whole system based on a majority rule will be $\pi_g \approx 3\%$. The probability of false negative is about 3% and the probability that the gate will keep the evidence (under these assumptions) is 97%. Grouping computers in three and make them feed a majority rule gate would produce an aggregate with a somewhat improved probability of false positive and false negative.

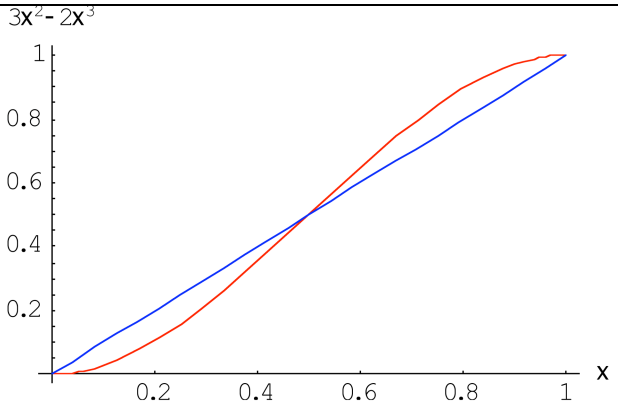


Fig 1: Output of a majority rule gate, assuming that: $\chi_1 = \chi_2 = \chi_3 = \xi$, i.e. that: $\pi_g = 3\xi^2 - 2\xi^3$.

This means that if in average, 90% of the incoming lines are activates, 97.2% of the outgoing lines will be activated. On the other hand, if in average 10% of the incoming lines are activated, only 2.8% of the outgoing lines are activated. This can be interpreted as meaning that if the detectors were identical and had in average a 10% false positive or false negative rate, a majority rule gate would have a 2.8% false positive or false negative rate. Another way to use this graph is to choose a value for the maximum allowable probability of false positive for the gate (say 1%). Then the maximum tolerable probability of false positive for the computer sending a message to each of the three lines (in a sample where all the lines are activated) is 5.8%.

In a stochastic situation there are fluctuations. The majority rule gate has the potential to amplify their effect (by filtering our the "right information") and distort the message. In order to avoid a distortion of the original message due to statistical fluctuations, von Neumann makes a compelling case that multiplexing the information sent by the computers helps.

Von Neumann studies in details the effect of multiplexing in the case of the Sheffer stroke (NAND gate). The N times multiplexed Sheffer stroke is made of 2N lines arrived in pair in N NAND gates and N lines come out. It is only when both lines of a pair are simultaneously activated that the line leaving the gate is not. If one assumes that among the incoming lines arriving in pair, p of N and q of N respectively are activated, the probability that r of the N outgoing lines are not, is given by the ratio of the number of ways that a combination of p and q lines generate r lines with the right properties, over the total number of ways of combining p and q lines in pair:

$$\rho(r,N) = \frac{N!}{\frac{r!(p-r)!(q-r)!(N-p-q+r)!}{\frac{N!}{p!(N-p)!}\frac{N!}{q!(N-q)!}}}$$

Equation 2

Defining $p = \xi N$, $q = \eta N$ and $r = (1-\zeta)N$, one can show that in the large N limit:

$$\rho \approx \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2\sigma^2}(\zeta - (1-\xi\eta))^2\right)$$

Equation 3

With:

$$\sigma = \sqrt{\frac{\xi(1-\xi)\eta(1-\eta)}{N}}$$

Equation 4

I.e. one effect of large N is that the outcome is normally distributed with mean $\zeta = 1-\xi\eta$ and with a variance decreasing with N.

MATHMATICS WITH ERRORS

Equation 3 suggest that in the large N limit, when ξ and η represent the excitation level of the input bundle, $\zeta = 1-\xi\eta$ is the relative excitation level of the output bundle, assuming the gate do not introduce any additional error. If one assumes that the gates have a non-zero probability "ψ" to make an error, the result is modified into:

$$\rho \approx \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2\sigma^2}\left(\zeta - (1-\xi\eta) - 2\psi\left(\xi\eta - \frac{1}{2}\right)\right)^2\right)$$

Equation 5

with:

$$\sigma = \sqrt{\frac{(1-2\psi)^2 \xi(1-\xi)\eta(1-\eta)+\psi(1-\psi)}{N}}$$

Equation 6

von Neumann shows that in order for the network to be functional, the error probability ψ should be less than 1%. If $\psi < 0.01$, it is possible to improve the accuracy of the network by increasing the number N. If one assumes $\psi = 0.005$, von Neumann found that the probability of malfunction was:

$$p(N) = \frac{1}{\sqrt{2\pi}} \int_{\kappa}^{\infty} e^{-\frac{k^2}{2}} dk \text{, with } \kappa = 0.062\sqrt{N}$$

Equation 7

For N = 1000, the probability of malfunction of the whole system is 0.249%. For N = 5000, it becomes 5.82×10^{-6}.

In the context of intrusion detection, the probability that the system is functional corresponds to the probability that the detection system makes the right identification. In Figure 2, N corresponds to the number of "Gates".

In our architecture, the source of the activation of the system will be computers detecting anomalies. We have in mind a set-up where the message of each individual computer is multiplexed and sent to many "gates". The gates would not be Sheffer strokes, but gates receiving signals from many computers (Whether in actuality the "gates should be the same computers is not excluded, but of no relevance in this discussion). The "gates" have to make a determination whether they should pass the signal on. Following von Neumann computation, the probability of error by the gates should be kept very small (significantly less than one percent if it were Sheffer strokes), otherwise the intrusion detection system will not provide an information with adequate precision: it will be "dysfunctional".

Our "gates" will defer significantly from von Neumann's gates. How they will operate is not yet completely clear. In this proposal we show that it is (at some computational cost) possible to generalize the results of von Neumann to other gates.

MATHMATIC GENERALIZATION TO OTHER GATES

Pushing one step further the analogy with the brain, we imagine a system with a very high cross connectivity. I.e. we imagine a system with N (large) sensors (level 1) connected to N sensors (level 2), And each level 2 sensor is connected with q level 1 sensors.

Assume that out of the N level 1 sensors, p are activated and send a signal. Let k be the number of signals received by individual level 2 sensors. The distribution of k is the hyper geometric distribution:

$$H(N,p,q;k) = \frac{\binom{p}{k}\binom{N-p}{q-k}}{\binom{N}{q}}$$

Equation 8

Its mean is: $\frac{q\,p}{N}$ and $H(100,50,10;k)$ has the general shape show in Figure 2.

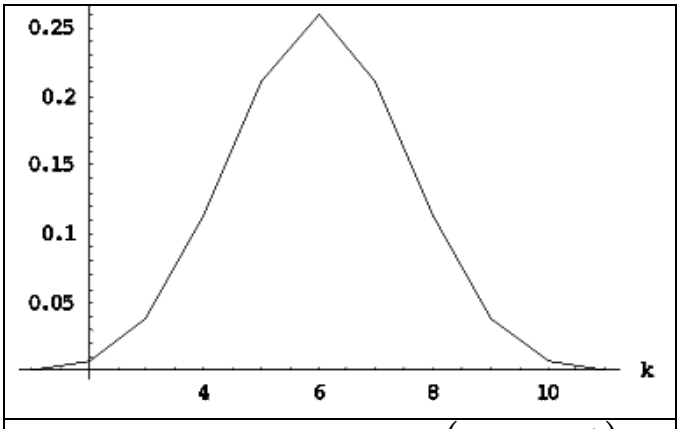


Fig 2: Hyper geometric function $H(100,50,10;k)$.

MATHMATIC PROBABILITY OF FALSE POSITIVES

If each level 1 sensor has the same probability π of false positive, the probability that k signals are all false positive is: $P^{FP}(k) = \pi^k$. One can choose a threshold κ on the value k ($k \geq \kappa$) for the level 2 detector to be triggered, such that $P^{FP}(\kappa) = \pi^{\kappa} \leq \varepsilon$.

FUTURE IMPROVMENTS

Clearly, the proposed system falls short of providing complete detection of arbitrary virus spread throughout a network with zero false positives. Yet notification of suspicious events throughout a network is helpful for a system administrator, or a company like Symantec. The authors, however, have considered several ways to improve

the current model. The following are some possible additions to the architecture to facilitate more complete and/or accurate conclusions.

Currently, sensors send a simple "error" indication. A simple modification of the architecture would be to elaborate event signals with a brief description of the anomaly that occurred. Such metadata could improve detection and diagnosis in several ways. On a pragmatic level, a supervisor or administrator could examine the profiles of the events at the brain level of the VNN, and draw more informed conclusions. On a more autonomic level, the brain could examine the profiles, and match similarities between events, and system configurations to find possible commonalities or patterns. However this metadata is used, it would provide for a more accurate level of reseponse, beyond mere notification. Of course, the cost is the requirement for higher network and processing bandwidths, to send and manipulate the metadata, and larger storage requirements at any node responsible for temporal aggregation or time shifting.

Another modification would be to have specialized parallel networks layered on top of the same base network. Each layer would be used to deal with a unique profile or common pattern representing a given threat. This would allow the network to detect the spread of each category of threat throughout the network. As a result, the network would be alerted each time a specific threat has propagated through the network. However, this poses a new set of questions: How would a designer determine what threats are the same, and what are different? What is the optimal number of parallel networks? Would the reliance on specific networks to detect threats reduce the sensitivity to new, previously unobserved infections?

Once the architecture separates different kinds of threat detection, the problem of recombining sets of events from different networks becomes significant. We have explored several possible ways of identify and group such threats. One way is by detecting anomalies by a series of unusual system calls. A snapshot of these system calls would be used as a rough signature for the identification of a specific threat. As the alert travels through the network, a signature or pattern match would be run against each alert. This pattern matching could be done using a dynamic programming algorithm, hash tables, or the like. However, for dealing with more complicated threats, more intelligent pattern detection, or anomies detection might be necessary. When a match reaches a certain trigger level, it would be considered significant. This sounds very promising, but unfortunately, every detection network would need to have a deep understanding as to what series of system calls represent a class of anomalies. This could be difficult in a heterogeneous configuration of the different types of systems (hardware and operating systems) or in a homogeneous configuration with different tasks being performed on each node. Reliance on a fix set of detection patterns, could be vulnerable to malicious code that chose random system calls (generally considered normal) to mask its critical calls.

Another possible solution to this problem of correlating events would be to trace back anomalous activity to the binary events that began the detection pattern. Therefore, no matter the series of system calls, or activities, however a virus or worm is detected, the same initial binary event would be able to be compared. However, this may prove to be easier said then done. In a very superficial analysis, we can only find a function called "strace." Though this function does appear to be a good start, it does not appear to be a quick solution. One problem with this binary solution is that the malicious code could simply generate random bits, thus similar binary events on different systems would appear to be different.

Lastly, we have discussed using a neural network to self organize based on known infections, and the network history. Clearly, this is the most complex of the proposed solutions, and it is also the one we have pursued the least. The main reason for the lack of Neural Network exploration is because it only deals with half of the problem. Though it would help mobilize the detection based on historical patterns, and known spreads, it does not help reduce false positives for new threats, which spread in an unconventional pattern. Consider the situation where a system is constantly running compromised software, and always downloads and spreads the newest and greatest worm or virus. The NN solution would clearly help in fast detection. However, for the threats which target general users (ie ones that spread over email or target systems with specific configurations, like SLAMMER), the NN might not help, or could even hurt if the network is designed to look within a given pathway.

Though none of these additional enhancements solve a problem completely, they do make a step in the right direction. We strongly believe in the direction of this work, and further meditation on these solutions could provide the basis for a stronger, and more reliable network that would not only inform about events, but also provide information to combat the threats.

FUTURE WORK

The next step in this research would be to conduct real world testing of our architecture. The goal of this would be to develop the code-level architecture and protocol system to determine the necessarily level of complexity that would be necessary to support virus detection. In addition to being a proof of concept, this implementation would also allow us to observe how the actual system reacts to a virus/worm threat, and what kind of analysis we can perform on the results. Though we know what information we would like systems to pass on, this would allow us to

examine all the information we can collect about a virus. This approach would also provide a useful baseline of information for the study of more intelligent reactionary systems. This small initial implementation would also serve as the basis for improvements in robustness, practicality and scalability.

In addition to testing the feasibility of such a network, implementing it on top of an existing system could provide real world results. Symantec has a large scale virus detection network called Deepsight, which would provide a great opportunity to test our model. It is feasible to consider working with Symantec towards testing this research in the future.

CONCLUSION

This project has laid the groundwork for a broad agenda of virus detection research. Von Neumann's original paper provides us with a method to systematically reduce false positives based on the law of large numbers. We have shown, both conceptually and mathematically, that it can be applied to virus detection.

More generally, our work shows that we can use the same property of large numbers that the brain uses to weed out faulty information. In effect, the total is better than the sum of the parts. Even more intriguing is that sometimes the most reliable components do not always produce the most reliable output.

One observation is that many detectors will be needed to put into practice the theory discussed here. And the larger the number of detectors, the better the automatic detection of new viruses through anomalies will be. Given the current state of virus detection in the real world, if it were anomaly based, it would be effective if the malicious traffic was about 1% or more of all the traffic. At this point, if detection occurred, it would be too late. We hope our proposed model can improve the chance of virus detection.

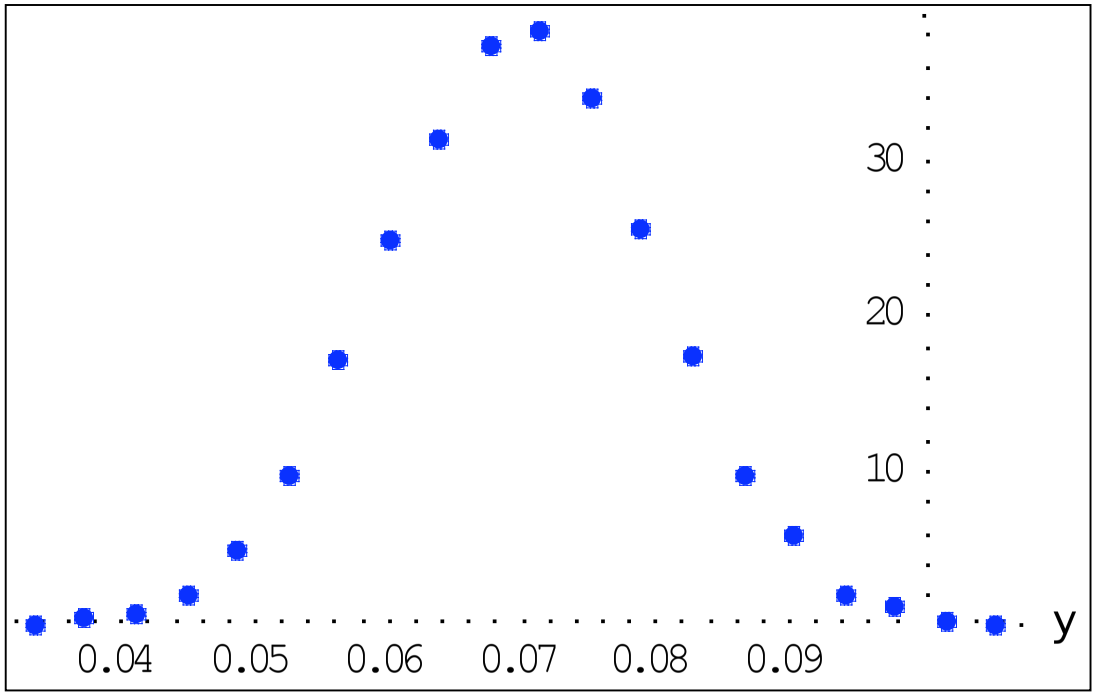
ACKNOWLEDGMENTS

I would like to thank my advisor Professor Morel for his years of support and guidance in my academic pursuits, and for his friendship. In addition, I would like to thank Mark Stehlik for seeing me through my years at Carnegie. Lastly, I thank my parents and friends for their tireless support and love.

REFERENCES

1. von Neumann, J. "Probabilistic Logics and the Synthesis of Reliable Organisms from Unreliable Components." C.E. Shannon and J. McCarthy, editors*, Annals of Math Studies,* number 34. Princeton Univ Press. (1956): 329-378.

Von Neumann Upper ε



$\varepsilon = 0.0107$

$N = 1000$

$\Delta = 0.07$

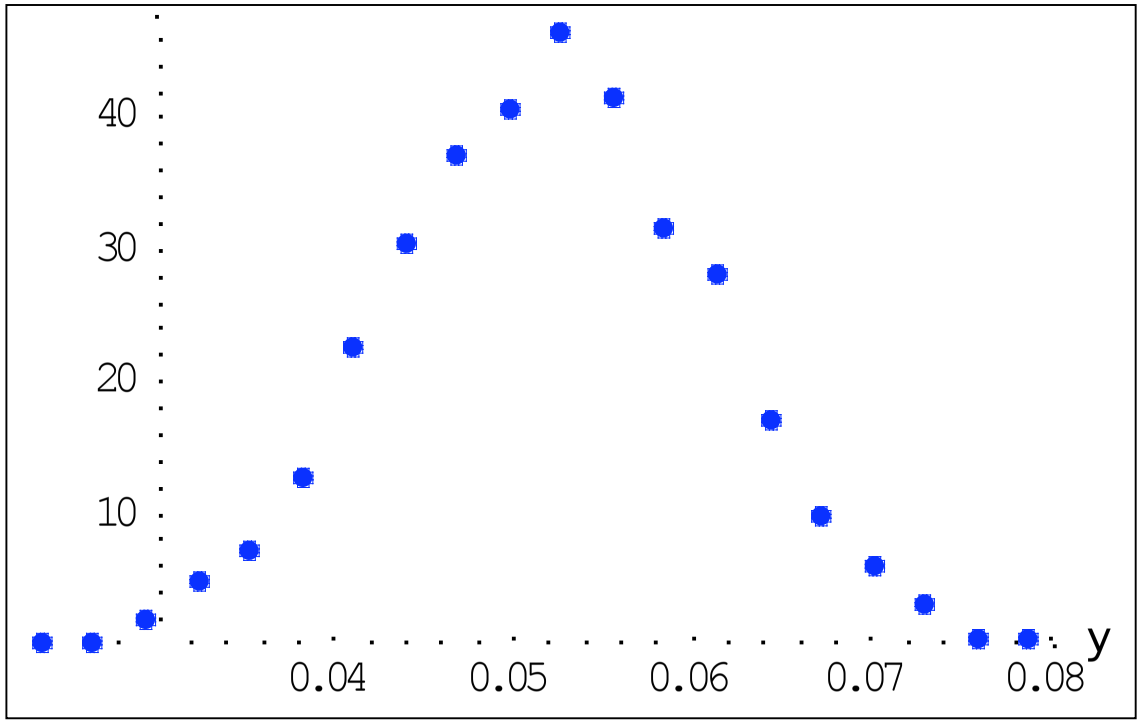
$> \Delta = 0.489$

"Most favorable fiduciary level … =.07. That is to say.. At least 93 % of the lines report A positive message…" - Von Neumann (365)

"There exists also an upper bound for the allowable values of e… = 0.0107"
- Von Neumann (365)

Appendix 1

Von Neumann Tested ε



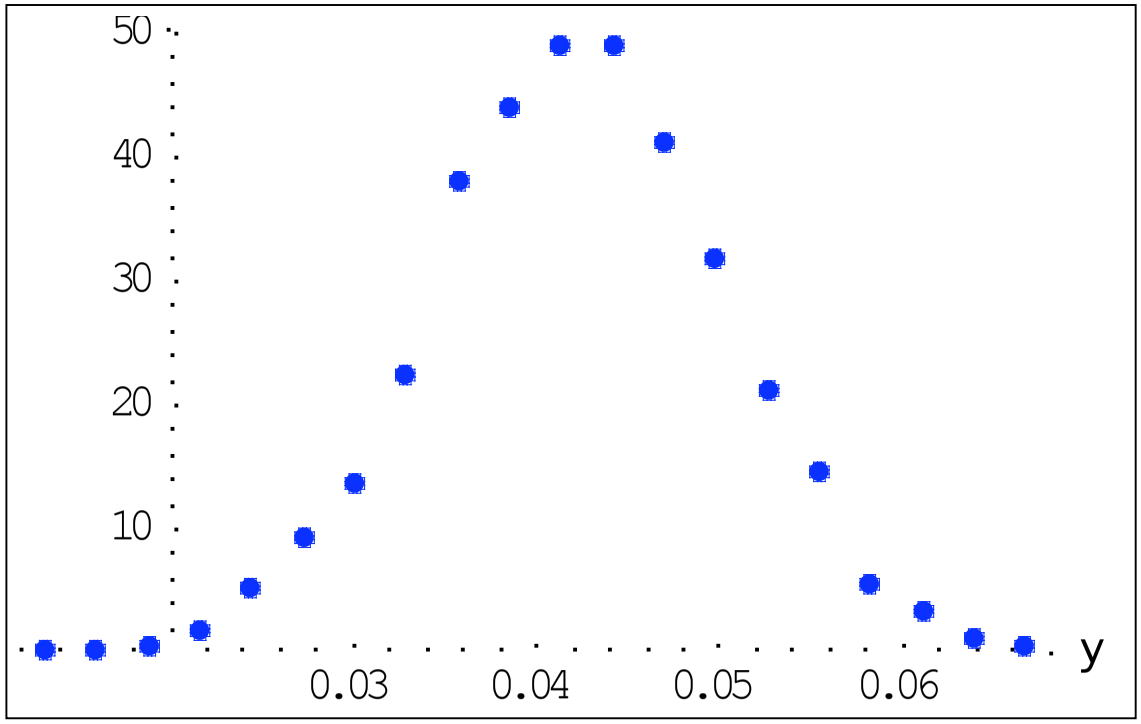
ε	=	0.05
N	=	1000
Δ	=	0.07
> Δ	=	0. 0203333

Stimulation would be of at least 93 % of the lines have a positive message

Risk of malfunction is 0.05

Appendix 2

Very Small ε



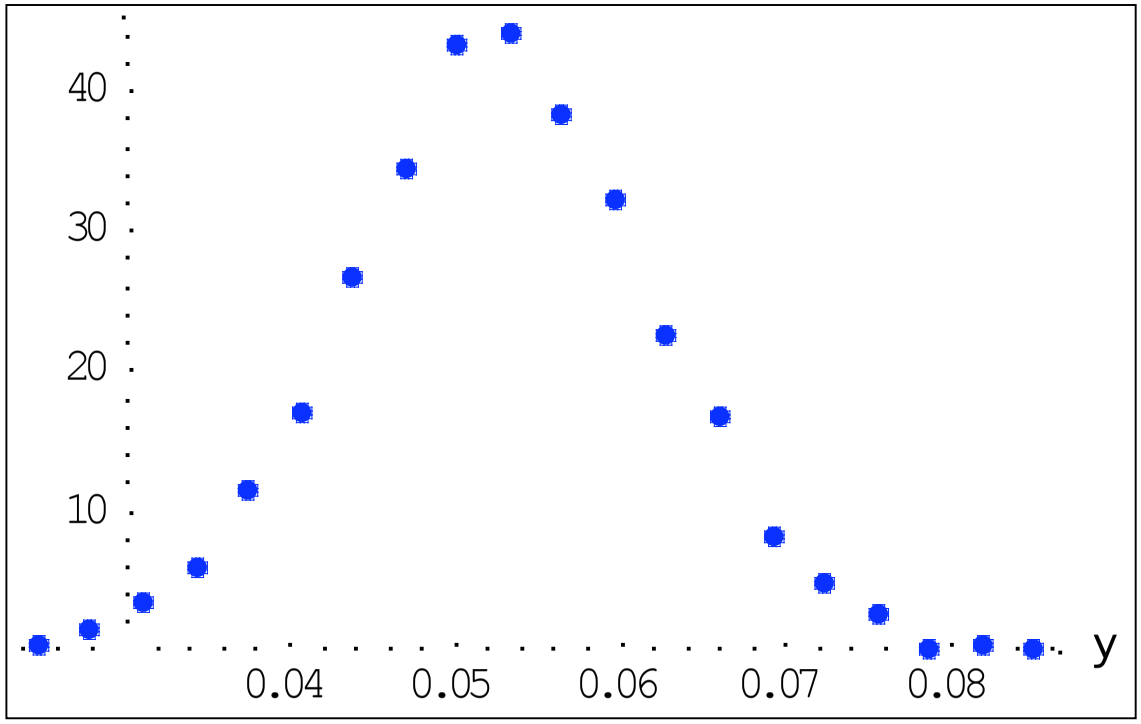
ε	=	0.002
N	=	1000
Δ	=	0.07
> Δ	=	0. 0

Stimulation would be of at least 93 % of the lines have a positive message

Risk of malfunction is 0.002

Appendix 3

Change Delta (smaller)



$\varepsilon = 0.0107$

$N = 1000$

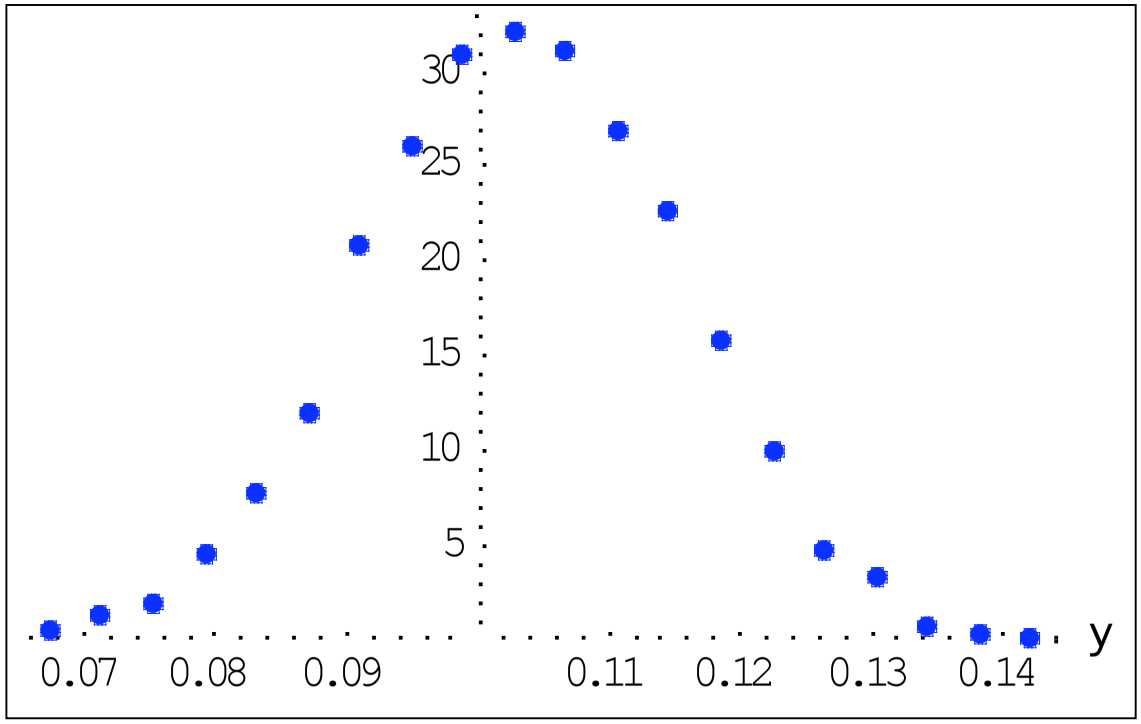
$\Delta = 0.05$

$> \Delta = 0.617667$

Stimulation would be of at least 95 % of the lines have a positive message

Appendix 4

Change Delta (larger)



$\varepsilon = 0.0107$

$N = 1000$

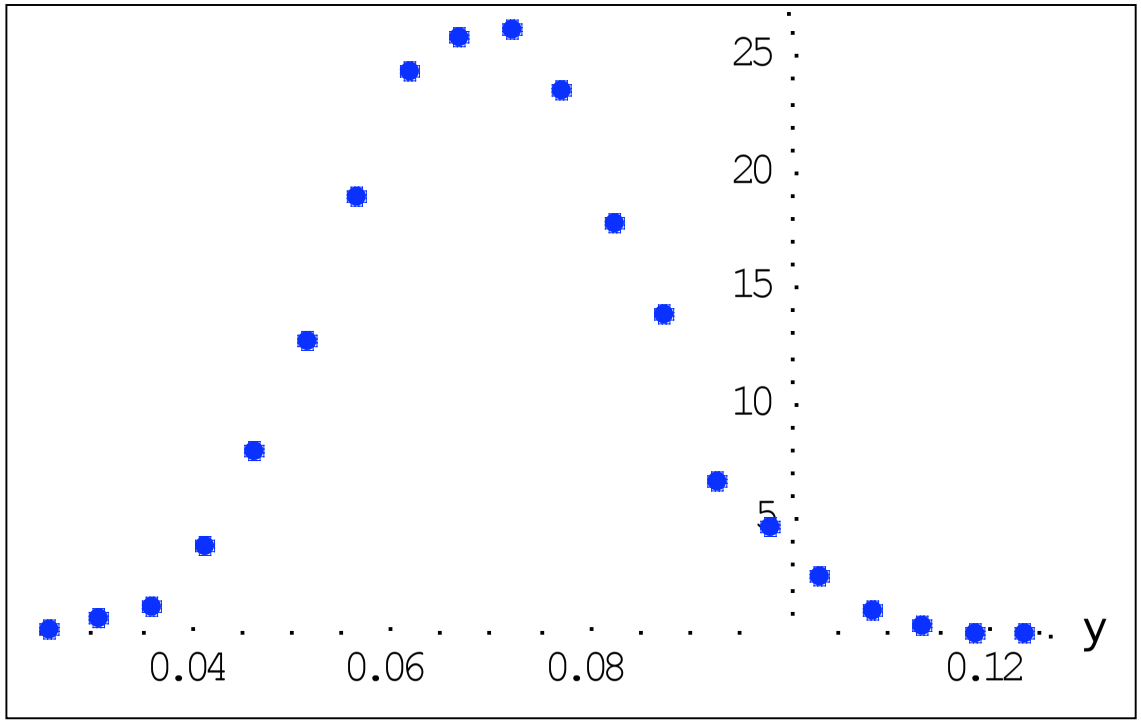
$\Delta = 0.1$

$> \Delta = 0.2667$

Stimulation would be of at least 90 % of the lines have a positive message

Appendix 5

Smaller # of Organs



$\varepsilon = 0.0107$

$N = 500$

$\Delta = 0.07$

$> \Delta = 0.490333$

About half of the output is above the fiduciary level, 0.07

Appendix 6